

Ambiente de Simulação Gráfica 3D para Ensino da Arquitetura de Processadores

Eduardo de Lucena Falcão¹, Erick Vagner Cabral de Lima Borges¹, Igor Lucena Peixoto Andrezza¹, Glauco Sousa e Silva¹, Kívio Grangeiro Wanderley¹, Berg Élisson Sampaio Cavalcante¹, Hamilton Soares da Silva¹

¹Departamento de Informática – Universidade Federal da Paraíba (UFPB)
Cidade Universitária 58.059-900 – João Pessoa – PB – Brasil

{eduardolfalcao, erickvagnerr, igorlpa90, glauco.sousa6, wkivio, bergelisson01}@gmail.com, hamilton@di.ufpb.br

Abstract. *In Computer Architecture, one of the challenges is getting the students to understand the workings of a processor correctly. To assist in this task, applications are used to simulate these architectures. From these barriers of education, is proposed the development of an application that visually simulates the instructions of an implementable and didactic architecture. Such instructions can be visualized, both with and without pipeline, which facilitates the comprehension of this concept, and of the operation of a processor while executing an instruction.*

Resumo. *Em arquitetura de computadores, um dos desafios é conseguir que os alunos compreendam o funcionamento de um processador corretamente. Para auxiliar esta tarefa são utilizadas aplicações que simulam essas arquiteturas. A partir dessas barreiras do ensino, se propôs desenvolver uma aplicação que simula visualmente as instruções de uma arquitetura didática e implementável. Tais instruções podem ser visualizadas, tanto com quanto sem pipeline, o que facilita a compreensão desse conceito, e do funcionamento de um processador enquanto realiza uma instrução.*

1. Introdução

A informática tem tido uma relevância cada vez maior nos dias de hoje no cenário educacional. Aplicações para o ensino facilitam a transmissão do conhecimento de professores para alunos através de métodos dinâmicos, massificando uma aula para múltiplos ramos sem haver grande esforço ou repetições desnecessárias.

O uso da computação gráfica tem se mostrado útil nos métodos de aprendizagem em diferentes áreas. O uso de CAD, sistemas computacionais que auxiliam na construção de desenhos técnicos, na geologia, engenharia e arquitetura, por exemplo, mostra o quanto uma interface gráfica com 3D facilita no desenvolvimento de um projeto, no seu entendimento e posterior ensino.

Facilitar a aprendizagem do correto funcionamento de um processador para os alunos e proporcionar uma visão comparativa sobre algumas possibilidades arquiteturais são os principais problemas encontrados no ensino de arquitetura de computadores. Um desafio ainda maior é tornar o conteúdo atrativo visualmente. Várias caixas e fios com letras pequenas tornam os circuitos um assunto de difícil entendimento a partir de uma simples leitura.

Os simuladores atuais que estão disponíveis para o ensino apresentam uma interface de usuário pouco elaborada e com poucos recursos operacionais, e mesmo considerando o uso de arquiteturas mais simples, é usual por parte dos alunos certa dificuldade de trabalhar com eles [Borges e Silva 2006]. A grande maioria destes simuladores exige dos alunos codificarem o sistema, ou seja, é necessário conhecimento em programação de linguagem de máquina ou rodam no DOS apenas exibindo os valores presentes nos registradores, o que torna a aprendizagem do funcionamento da arquitetura do processador uma tarefa longa e exigente.

Para auxílio a esta tarefa, está sendo desenvolvida uma aplicação com objetivo de prover maior motivação e interesse por parte dos alunos em participar das aulas de arquitetura de computadores. Recursos como animações gráficas tridimensionais estão sendo utilizados para facilitar o entendimento do funcionamento dos processadores, já que na prática não é possível ver o caminho percorrido pela informação processada. Este simulador não exige do aluno nenhum grau de programação para que a arquitetura seja demonstrada, Dessa forma é possível efetuar um estudo detalhado dos elementos presentes na arquitetura.

2. Trabalhos Relacionados

Existem muitas ferramentas que simulam processadores, como exemplos temos o Abacus, WinMIPS64 [Scott 2006], SATSim [Wolf e Wills 2000] e o GNUSim8085 [Sridhar 2002].

O Abacus (desenvolvido em Windows) e o GNUSim8085 (desenvolvido para plataformas Unix) simulam o microprocessador 8085 da Intel, o qual tem sido base do ensino de arquitetura de computadores, devido a sua simplicidade, seguindo o modelo de Von Neumann e sem possuir *pipeline*. Ambos apresentam os valores dos registradores no decorrer da execução das instruções, além de exibir o conteúdo presente na memória do processador.

O simulador WinMIPS64 simula a arquitetura do processador MIPS64, que utiliza o *pipeline*. Essa ferramenta possibilita a visualização das instruções passando pelos *pipelines* e os valores presentes nos registradores.

Para o ensino de arquiteturas *superescalares* há o SATSim que não se baseia em nenhum processador real, dessa maneira ele possibilita a configuração da quantidade de unidades funcionais por tipo de instrução, o número de elementos por *buffer* de reordenação, o número de instruções buscadas por ciclo e etc.

3. Fundamentação Teórica

3.1. Arquitetura de Processadores

É necessário conhecer a arquitetura interna de um processador para facilitar a compreensão de como ele executa programas. Em um processador existem vários circuitos chamados de registradores que funcionam como posições de memória onde o seu acesso é extremamente rápido.

Todos os processadores usam uma linguagem baseada em códigos numéricos na memória, onde cada código significa uma instrução no qual chamamos de linguagem de máquina. Os códigos numéricos são representados por siglas que são os nomes das

instruções, e os operandos dessas instruções são os registradores, valores existentes na memória e valores constantes.

Alguns processadores são baseados no modelo arquitetural de Von Neumann que considera uma unidade central de processamento (CPU) com apenas uma unidade funcional, podendo buscar apenas uma instrução por ciclo de execução como demonstrado na Figura 1.

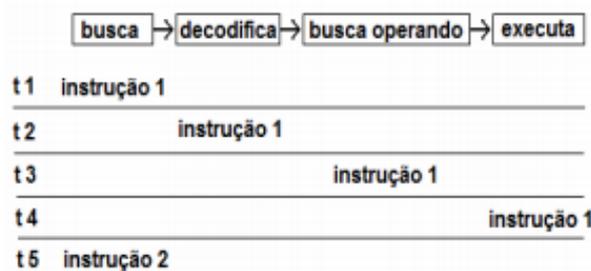


Figura 1 – Ilustração do modelo de Von Neumann

Para o ensino de arquitetura de computadores, é interessante demonstrar o conceito de *pipeline*, técnica de hardware que permite que a unidade central de processamento realize a busca de uma ou mais instruções além da próxima a ser executada. Estas instruções são colocadas em uma fila de memória dentro do processador (CPU) onde aguardam o momento de serem executadas, esta técnica permite que o tempo de processamento de uma instrução seja menor do que o tempo que seria gasto caso o *pipeline* não fosse utilizado. A estratégia do *pipeline* é demonstrada na Figura 2.

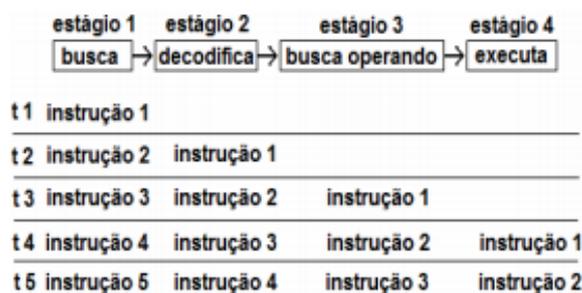


Figura 2 – Ilustração da Estratégia do pipeline

3.2. X3D

O X3D (*Extensible 3D*) é um padrão aberto adotado internacionalmente para disponibilização de conteúdos tridimensionais na Web [Brutzman e Daly 2007]. Ele permite descrever em um arquivo, formas e comportamentos de ambientes virtuais tridimensionais complexos e interativos.

A visualização destas cenas descritas em X3D é realizada através de *browsers* específicos que consistem em aplicações capazes de interpretar e processar os arquivos X3D, apresentando os modelos tridimensionais, animados ou não, e permitindo interações do usuário com os objetos. Os *browsers* X3D (também chamados de navegadores ou *players*) podem se apresentar como *plugins* ou *applets* em navegadores

Web como o *Mozilla Firefox* ou *Google Chrome*, ou como aplicações independentes [Web3D Consortium 2011].

Para a formação gráfica da cena são utilizados os nós X3D, que por sua vez compõem o grafo de cena. Cada nó é formado por uma seqüência específica de campos. Os nós descrevem as funcionalidades disponibilizadas pelo X3D, e os campos servem para especificar os atributos e características dos nós X3D utilizadas para descrição tridimensional do ambiente. A interatividade de uma cena X3D pode ser caracterizada por mudanças de posição, orientação, tamanho, cores ou outras características pertinentes aos campos do nó X3D, resultando em uma animação. Eventos e rotas são as funcionalidades do X3D que permitem adicionar tais comportamentos a objetos do ambiente virtual de forma simples. Para interações mais complexas são utilizados o nó *Script* e a API SAI (*Scene Access Interface*) que permite o controle do grafo da cena através de linguagens de programação [Web3D – *Part 2: Scene Access Interface*].

Para este trabalho, foi utilizado o *browser Xj3D*, a API SAI (*Scene Access Interface*) que permite manipulação do grafo de cena X3D, e a linguagem de programação Java, para programar toda interatividade do aplicativo.

4. Simulador desenvolvido

O simulador produzido demonstra o funcionamento do processador didático apresentado em [Tanenbaum 2006], que apesar de não ser comercial é implementável.

O nível de microarquitetura que será descrito tem a função de simular o nível ISA (*Instruction Set Architecture* – Arquitetura do Conjunto de Instruções). O Caminho de Dados utilizado para a simulação é a parte da CPU que contém a ULA, suas entradas e saídas. Ele contém vários registradores aos quais foram atribuídos nomes simbólicos como PC, SP, MDR e MBR, onde MBR foi dividido em MBR1 com 8 bits e MBR2 com 16 bits. É importante ressaltar que esses registradores são acessíveis apenas no nível de microarquitetura (pelo microprograma).

Com o objetivo de que o comprimento do caminho de execução seja menor, foram utilizados dois barramentos completos de entrada para a ULA (barramentos A e B) e um barramento de saída. Todos (ou ao menos a maioria) dos registradores devem ter acesso a ambos os barramentos de entrada. A vantagem de ter dois barramentos de entrada é que não é possível adicionar um registrador a outro registrador em um só ciclo.

O simulador é composto por uma interface gráfica que contém uma lista das instruções disponíveis, uma breve descrição do objetivo da instrução escolhida e um espaço reservado para o visualizador da simulação. Na lista fornecida, o usuário escolhe qual instrução do processador deseja visualizar como ilustrado na Figura 3.

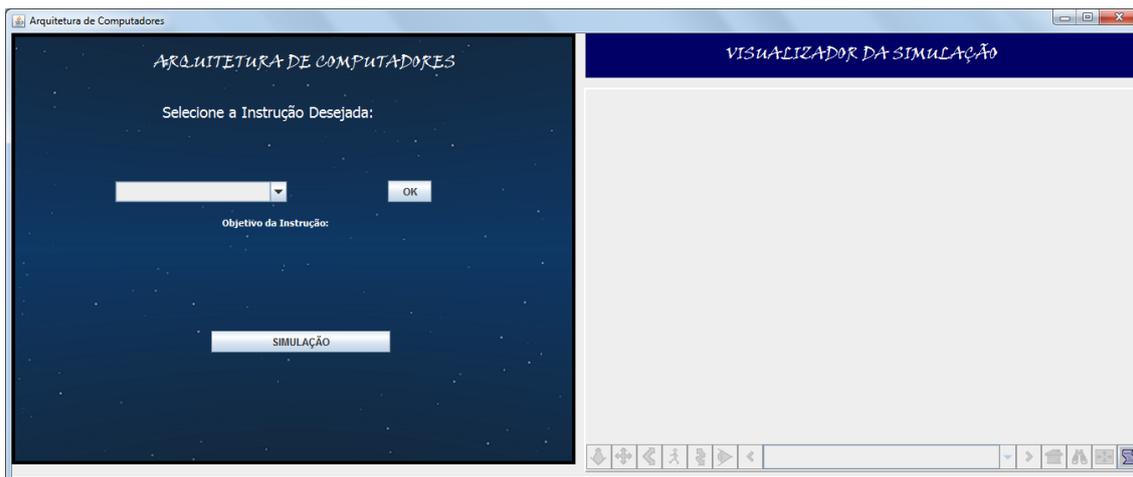


Figura 3 – Tela Inicial

Após a escolha da instrução o botão “OK” deve ser pressionado. Então uma breve descrição com o objetivo da instrução é mostrada logo após, como se observa na Figura 4.

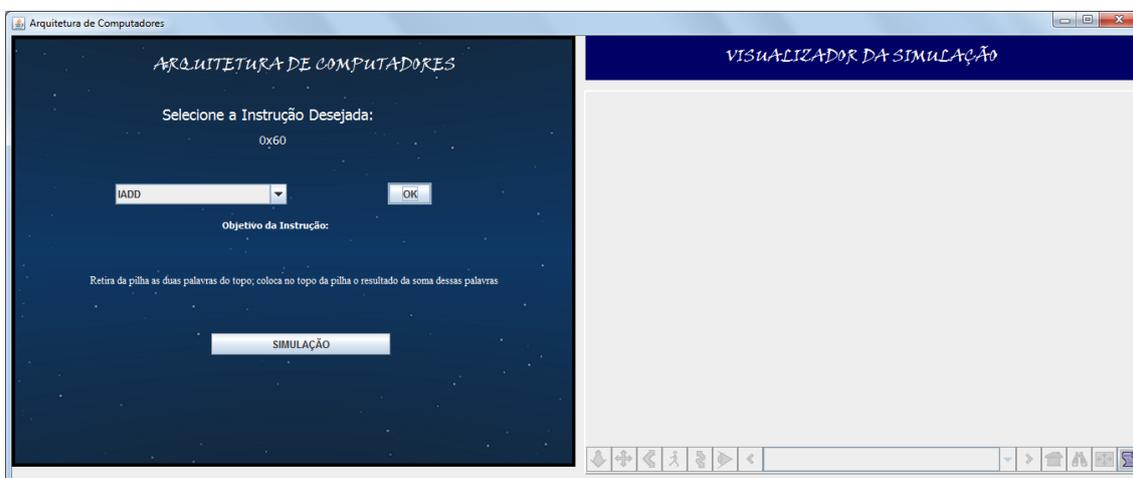


Figura 4 – Descrição do Objetivo da Instrução

Ao pressionar o botão “SIMULAÇÃO” ela é iniciada de acordo com a instrução escolhida. A posição das telas lado a lado facilita a compreensão, já que o usuário pode observar a descrição do objetivo e o mesmo em ação.

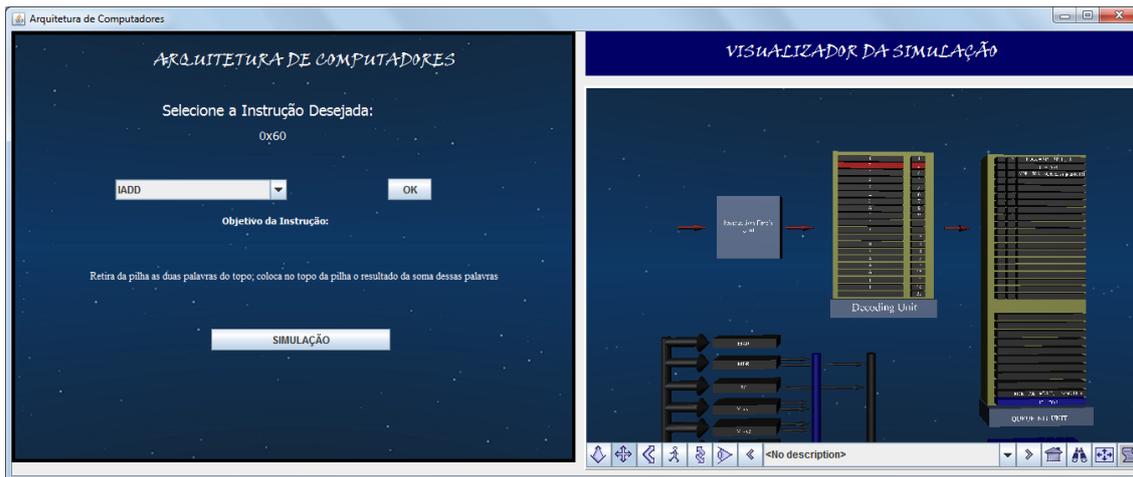


Figura 5 – Simulação da Instrução

A memória e todos os outros elementos do processador são representados em um contexto tridimensional. A simulação demonstra a seqüência de passos da execução de uma instrução na Figura 5, desde sua saída da memória até seu retorno, caso ocorra. A exibição de cada passo é feita através da interpolação das cores ou posição dos elementos que estiverem ativos. Cada instrução exibida contém um conjunto de micro-instruções que são demonstradas pela exibição passo-a-passo a cada *clock*, de acordo com o conceito de *pipeline* visto anteriormente, e são marcadas com uma cor diferente com o objetivo de demonstrar o *clock* e o *pipeline* como demonstrado na Figura 6. Para melhor visualização, a câmera acompanha o fluxo de passos da execução da instrução, focando nos blocos que estão sendo utilizados em determinado momento da simulação.



Figura 6 – Ilustração da Simulação do Pipeline

5. Conclusões e Trabalhos Futuros

O entendimento de como funciona uma determinada arquitetura é uma tarefa difícil de ser compreendida quando se estuda apenas a teoria. Isto evidencia a importância do uso de simuladores neste contexto.

A partir dessa constatação, o simulador apresentado neste trabalho visa facilitar o entendimento da arquitetura baseada no modelo de Von Neumann. A facilitação da compreensão ocorre devido à apresentação intuitiva das informações ao usuário pelo simulador. Os trabalhos correlatos encontrados na literatura apresentam apenas informações textuais dos valores de registradores, *flags* e memória, e algumas estatísticas.

O presente trabalho propõe que o fluxo de informações do processador seja ilustrado através de um ambiente 3D, que torna simples a visualização de informações sendo transferidas de um objeto como o barramento, registrador, ou ULA, para outros por intermédio da mudança de cores dos mesmos. O principal objetivo desse simulador é facilitar o entendimento dos alunos de cursos da Ciência da Computação e similares no estudo da arquitetura dos computadores. Outra característica positiva, que o diferencia em relação aos demais simuladores citados nesse trabalho, é a possibilidade de visualizar uma mesma instrução sendo executado em uma mesma arquitetura de processador, com e sem o uso do *pipeline*, facilitando de forma substancial a compreensão dessa importante evolução ocorrida na área.

Para trabalhos futuros propõe-se a realização de uma avaliação do uso do simulador por estudantes para coletar a impressão destes e verificar a facilidade de uso, e grau de aprendizado adquirido, para posterior aperfeiçoamento da aplicação. Propõe-se também a disponibilização via Web do aplicativo, promovendo o acesso dos alunos à ferramenta e possibilitando uma facilidade para a prática da Educação à Distância.

Referências

- Borges, José Antonio S. Silva, Gabriel P. (2006) “NeanderWin - Um Simulador Didático para uma Arquitetura do Tipo Acumulador”.
- Brutzman, D., Daly, L., (2007) “X3D: 3D Graphics for Web Authors”. Morgan Kaufmann Publishers.
- Moreira, Angélica A., Martins, Carlos Augusto Paiva da Silva. (2009) “R2DSim: Simulador Didático do RISC Reconfigurável”
- Scott, M. (2006) “WinMips64, version 1.5, School of Computing, Dublin City University, Ireland”, <http://www.computing.dcu.ie/~mike/winmips64.html>, 07 2009, 20.
- Sridhar, Z, (2002) “GNUSim8085, versão 1.3”, http://gnusim8085.sourceforge.net/index.php/Main_Page, 06 2009, 23.
- Tanenbaum, Andrew S. (2006) “Organização Estruturada de Computadores”, Editora Prentice-Hall.
- Verona, André B., Martini, João A., Gonçalves, Tiago Lopes (2009) “SIMAEAC: Um Simulador Acadêmico para Ensino de Arquitetura de Computadores”.

- Web3D Consortium, “X3D – FAQ (Frequently Asked Questions)”. Online: <http://www.lsi.usp.br/~lsoares/x3d/faq.html>. Acesso em: março/2011.
- Web3D – Part 2: Scene Access Interface, “Extensible 3D (X3D). Part 2: Scene access interface (SAI). ISO/IEC 197752.2: 2009”. Online: <http://www.web3d.org/x3d/specifications/ISOIECFDIS197752.2X3DSceneAccessInterface/index.html>. Acesso em março/2010.
- Wolff, M. e Wills, L. (2000) “SATSim: A Superscalar Architecture Trace Simulator Using Interactive Animation”, In: WCAE: Workshop On Computer Architecture Education, Vancouver, Canada.
- Zelenovsky, Ricardo. (2002) “PC: um guia Prático de Hardware e Interfaceamento”, Editora MZ Produções Culturais.