

# SEAC 2.0 - Uma Aplicação Web de Simulação Gráfica para Ensino de Arquitetura de Processadores

Teobaldo Leite Bulhões Júnior  
Universidade Federal da Paraíba  
Cidade Universitária, João Pessoa,  
Paraíba, Brasil  
teobaldoleite@ci.ufpb.br

Hugo Neves de Oliveira  
Universidade Federal da Paraíba  
Cidade Universitária, João Pessoa,  
Paraíba, Brasil  
hugoneves@di.ufpb.br

Erick Vagner C. de L. Borges  
Universidade Federal da Paraíba  
Cidade Universitária, João Pessoa,  
Paraíba, Brasil  
erickvagnerr@gmail.com

Igor Lucena Peixoto Andrezza  
Universidade Federal da Paraíba  
Cidade Universitária, João Pessoa,  
Paraíba, Brasil  
igorlpa90@gmail.com

Yuri Gil Dantas  
Universidade Federal da Paraíba  
Cidade Universitária, João Pessoa,  
Paraíba, Brasil  
ygd@lavid.ufpb.br

Raul Felipe de M. Fernandes  
Universidade Federal da Paraíba  
Cidade Universitária, João Pessoa,  
Paraíba, Brasil  
raulfelipe@gmail.com

Hamilton Soares da Silva  
Universidade Federal da Paraíba  
Cidade Universitária, João Pessoa,  
Paraíba, Brasil  
hamilton@di.ufpb.br

## ABSTRACT

The Computer Science and its various fields deal with several levels of abstraction. The higher the level of abstraction, less about hardware the scientist needs to know. Computer architecture is a field that deals with a rather low level of abstraction, which hinders their teaching for students in the computer area. To aid this learning, various graphics applications that simulate the hardware architectures have been developed for many different platforms, including Desktop and Web. In this context, this paper presents a web application that visually simulates the execution of instructions on a didactic and implementable architecture. The application, which is under construction, will provide the user, whether student or teacher, less abstract visualization of the execution of instructions on architectures with or without the pipeline and may be used to assist the teaching of computer architecture.

## RESUMO

A ciência da computação, em suas diversas áreas de atuação, lida com vários níveis de abstração. Quanto maior o nível de abstração, menos do hardware o cientista precisa conhecer. A arquitetura de computadores é uma área que lida com um nível de abstração bastante baixo, o que dificulta o seu ensino para alunos dos cursos da área de informática. Visando auxiliar esse aprendizado, várias aplicações gráficas que simulam as arquiteturas de hardware vêm sendo desenvolvidas para as mais

diversas plataformas, incluindo Desktop e Web. Nesse contexto, este trabalho apresenta uma aplicação Web que simula visualmente a execução das instruções de uma arquitetura didática e implementável. A aplicação, que está em fase de construção, fornecerá ao usuário, seja ele aluno ou professor, uma visualização menos abstrata da execução de instruções em arquiteturas com ou sem pipeline e poderá ser utilizada para auxiliar o ensino de arquitetura de computadores.

## Descritor de Categorias e Assuntos

Miscellaneous.

## Termos Gerais

Miscellaneous.

## Palavras Chaves

Arquitetura de Computadores, Simulador, Ensino.

## 1. INTRODUÇÃO

Métodos mais dinâmicos de ensino vêm ganhando uma importância cada vez maior no cenário educacional, pois facilitam o processo de aprendizagem. Na maioria das vezes, esses métodos envolvem o uso de simuladores computacionais. Exemplos de tais aplicações são os softwares CAD, presentes em áreas como engenharia, arquitetura e geologia e são utilizados na concepção de projetos e para fins educacionais. Algumas áreas, porém, sofrem com a carência de ferramentas que apoiem o processo de ensino-aprendizagem. Como exemplo, pode-se citar arquitetura de computadores, já que essa é uma disciplina que exige dos alunos um alto grau de abstração, tendo em vista que, na prática, é impossível visualizar fisicamente o funcionamento completo das partes internas do computador. Por isso, é importante a utilização de modelos que representem a arquitetura e a comunicação interna entre seus módulos. Esses modelos podem ser desde figuras, representando um modelo mais estático das arquiteturas, até

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

XXXXXXXXXXXXX – As informações serão preenchidas no processo de edição dos Anais.

simuladores gráficos, que possibilitam uma visualização mais completa e interativa dos sistemas.

Os simuladores atuais disponíveis para o ensino apresentam uma interface com usuário pouco elaborada, possuindo poucos recursos operacionais, e ainda que considerem o uso das arquiteturas mais simples, é usual por parte dos alunos terem certa dificuldade de trabalhar com eles [7]. Muitos desses simuladores exigem que os alunos codifiquem o sistema, sendo necessário conhecimento de programação em linguagem de máquina; ou são executados em terminal, apenas exibindo os valores presentes nos registradores, o que não é didático.

Neste trabalho é apresentada a segunda versão do Simulador para Ensino de Arquitetura de Computadores (SEAC 2.0), um software de simulação da execução das instruções da arquitetura didática e implementável definida em [3]. O simulador, que ainda está em fase de construção, é baseado no software apresentado em [5] e [6], contando com uma mudança para uma tecnologia mais moderna, versátil e adequada às necessidades da aplicação, o Unity. A mudança foi necessária porque a tecnologia utilizada em [5] e [6], o X3D, mostrou-se muito complicada na construção da interação com o usuário e na expansão das funcionalidades da ferramenta. Com o Unity, será possível, por exemplo, criar um ambiente muito mais interativo e distribuir o simulador em várias plataformas, como Android, iOS, Windows e Mac OS.

A ferramenta está sendo desenvolvida para prover um aumento no interesse e no grau de entendimento dos alunos em aulas de arquitetura de computadores. Recursos como animações gráficas são utilizados para melhorar o entendimento dessas aulas, já que, na prática, não é possível ver o que acontece com os dados enquanto eles são processados. O simulador apresentado não exigirá do aluno nenhum conhecimento em programação para que a arquitetura seja demonstrada e apresentará o fluxo que uma instrução percorre ao ser executada. Dessa forma, será possível efetuar um estudo detalhado dos elementos presentes na arquitetura do computador.

## 2. TRABALHOS RELACIONADOS

Abacus, WinMIPS64 [8], SATSim [9], GNUSim8085 [14] e a própria primeira versão do SEAC [SEAC] são exemplos de simuladores de arquitetura de computadores.

O Abacus, desenvolvido para o sistema operacional Windows, e o GNUSim8085, desenvolvido para plataformas Unix, simulam o microprocessador 8085 da Intel, um processador muito utilizado para o ensino de arquitetura de computadores. Ambos os simuladores são bem simples, uma vez que apenas apresentam os valores dos registradores e conteúdo da memória durante a execução das instruções.

O WinMIPS64 simula a arquitetura do processador MIPS64, sendo um pouco mais complexo que os simuladores citados acima, pois, além de mostrar os valores presentes nos registradores, ele representa o *pipeline*.

O SATSim é um simulador que não se baseia em nenhum processador real e que representa arquiteturas superescalares. O SATSim permite ao usuário configurar vários parâmetros da simulação, como o número de instruções buscadas por ciclo.

A primeira versão do SEAC, descrita em [5] e [6], é um simulador avançado em relação aos citados anteriormente. Ele representa graficamente os principais componentes da arquitetura descrita em [3] e simula a execução das instruções através de animações, o que facilita o processo de aprendizagem dos alunos. No entanto, o

simulador possui uma interatividade limitada com o usuário e não é um software multiplataforma, o que limita a sua utilização.

## 3. FUNDAMENTAÇÃO TEÓRICA

Os computadores atuais seguem a arquitetura de Von Neumann e são, portanto, constituídos de uma Unidade Central de Processamento (UCP), uma memória principal, que armazena dados e instruções, e equipamentos periféricos. O elemento-chave dessa arquitetura é a UCP. A UCP possui registradores, que são memórias locais de rápido acesso, Unidade de Lógica e Aritmética (ULA), responsável por realizar operações lógicas e aritméticas, e Unidade de Controle (UC), que, dentre outras coisas, gera sinais que controlam o funcionamento de toda UCP e decodifica instruções. De forma periódica, a UCP busca uma instrução na memória para ser executada, operação conhecida como *fetch*. A instrução buscada, entretanto, só é de fato executada após a sua decodificação, feita pela unidade de controle da UCP, e após a busca de seus operandos (que podem estar na memória). Esse processo é ilustrado na Figura 1.



Figura 1. Ilustração do Modelo de Von Neumann. Adaptado de [3].

As instruções executadas pelos processadores são identificadas por códigos numéricos, os quais são representados por símbolos mnemônicos, para facilitar a memorização. Essas instruções possuem operandos, que podem ser registradores, valores residentes na memória ou valores constantes. O conjunto desses códigos é chamado de linguagem de máquina.

Um conceito que deve ser abordado no ensino de arquitetura de computadores é o de *pipeline*. O *pipeline* é uma técnica que permite que a UCP realize a busca de uma ou mais instruções além da próxima a ser executada, mantendo internamente uma fila de instruções. A execução dessas instruções é dividida em estágios independentes, o que permite que duas ou mais instruções sejam executadas em paralelo, em estágios diferentes. Assim, o *pipeline* acelera bastante o processamento das instruções, como pode ser visto na Figura 2.



Figura 2. Ilustração de um *pipeline*. Adaptado de [3].

#### 4. MATERIAIS E MÉTODOS

A construção do SEAC foi dividida em duas etapas: a modelagem da cena e a lógica de programação. A modelagem foi feita usando a ferramenta de criação de modelos 3D Blender [15], devido à sua capacidade de criar cenas complexas usando apenas primitivas gráficas como pontos e retas. Na etapa de criação lógica da ferramenta usou-se a *engine* gráfica Unity, que une facilmente os modelos estáticos criados com um motor de renderização poderoso e interação com o usuário.

##### 4.1. Unity

O Unity é uma *engine* 3D proprietária criada pela UnityTechnologies e que conta com *workflows* intuitivos que possibilitam a publicação do conteúdo produzido em múltiplas plataformas, como iOS, Android, Windows, Mac OS, Linux, consoles e navegadores Web [4].

#### 5. SEAC 2.0

O simulador em desenvolvimento, denominado SEAC 2.0, demonstrará o funcionamento do processador didático apresentado em [3], que, apesar de não ser comercial, é implementável. A ferramenta inicial foi desenvolvida utilizando a tecnologia X3D, porém o uso dessa tecnologia gerou dificuldades no acesso dos usuários à ferramenta por meio de dispositivos móveis, assim como na geração de uma versão WEB estável, já que, por vezes, os usuários não conseguiram utilizá-la. A ferramenta também não continha originalmente a opção de enfileirar instruções em *pipeline*, contando apenas com o *pipeline* de microinstruções. O SEAC 2.0 simulará o *pipeline* de instruções. A Figura 3 demonstra os registradores e os barramentos contidos na arquitetura simulada pelo SEAC 2.0. A Figura 4 exibe o ambiente completo de simulação do SEAC.

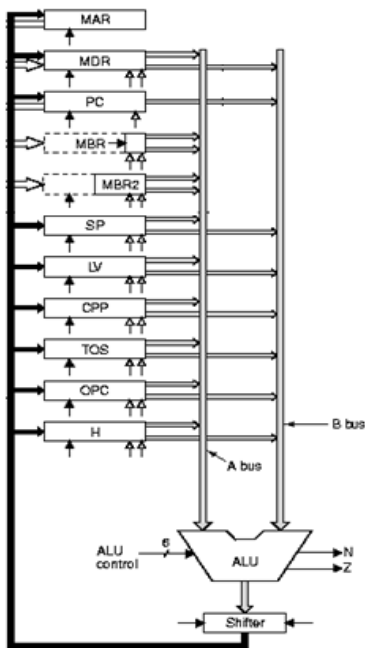


Figura 3. Caminho de Dados simulado pelo SEAC (Fonte: adaptada de [3]).

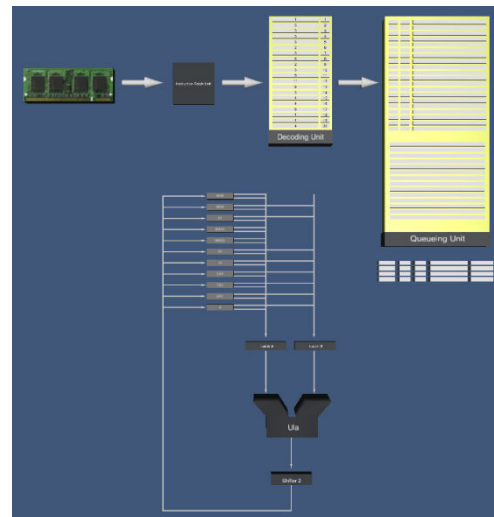


Figura 4. Ambiente de simulação do SEAC 2.0.

Com o início da simulação, a ferramenta carregará as instruções escolhidas e executará lentamente suas microinstruções. O usuário também poderá pausar a simulação a qualquer momento, tendo mais tempo para analisar cada passo da execução das instruções. Também será adicionada uma interface à parte que auxiliará na criação de novas instruções ao concatenar microinstruções. Para uma melhor visualização, o usuário pode movimentar a câmera e ajustar o zoom do cenário, acompanhando de forma mais específica o fluxo de passos da execução da instrução. A Figura 5 exibe a simulação de uma instrução no SEAC 2.0.

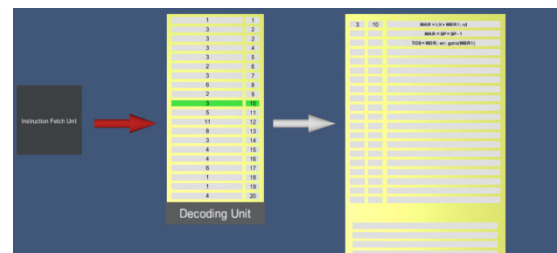


Figura 5. Simulação da instrução IAND.

##### 5.1. Representação do Pipeline

O simulador SEAC 2.0 pretende apresentar aos alunos conceitos importantes relacionados à arquitetura de computadores. Um desses conceitos é o *pipeline*, que possibilita a execução de instruções paralelamente apenas alterando seus modos de execução. O SEAC permitirá que o usuário visualize a execução de instruções com e sem *pipeline*.

O SEAC 2.0 demonstra como o processador executa uma instrução, ilustrando a passagem das microinstruções pelos elementos na ordem: Registradores, ULA, Registradores, Memória. Quando uma instrução é executada sem *pipeline*, uma microinstrução deve percorrer o caminho inteiro para que a próxima microinstrução possa começar a percorrer o mesmo caminho. Quando a instrução é executada com *pipeline* (Figura 5), as microinstruções podem percorrer esse caminho simultaneamente. Para melhor identificar essas diferentes microinstruções, cada microinstrução é identificada por uma cor. Esse esquema de cores já foi utilizado na primeira versão do

SEAC e é ilustrado na Figura 6. O *pipeline* do SEAC 2.0 ainda está em desenvolvimento.

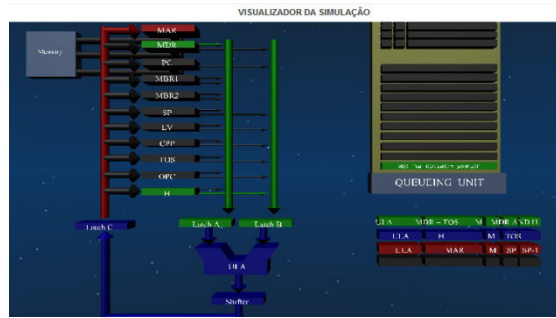


Figura 6. Simulação da instrução IAND com *pipeline* na primeira versão do SEAC.

### 5.2. Utilização do SEAC 2.0

Inicialmente, o simulador havia sido desenvolvido para ser uma ferramenta desktop[5][6]. Entretanto, para facilitar o acesso por outras instituições de ensino, por professores e por alunos, e para proporcionar um incentivo à prática da educação à distância, optou-se por disponibilizar a primeira versão do simulador através de uma aplicação Web e de um aplicativo para dispositivos móveis. Tanto no aplicativo como no site, o simulador será discutido e a sua utilização será demonstrada através de tutoriais em vídeo, incluindo simulações com e sem *pipeline*, para permitir o contato com o simulador mesmo para usuários submetidos a alguma limitação técnica. O *site* e o aplicativo contemplarão também uma apresentação do processador Mic-4, explicando o que é cada componente da arquitetura e sua funcionalidade [3]. A Figura 7 demonstra essa página do *site*.

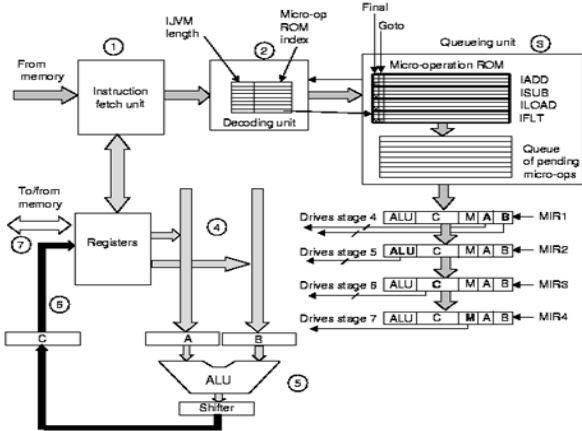


Figura 7. Tutorial com ilustração do processador Mic-4 [3].

## 6. CONCLUSÃO E TRABALHOS FUTUROS

O simulador proposto está sendo desenvolvido para facilitar o entendimento da arquitetura baseada no modelo de Von Neumann, contendo uma apresentação intuitiva das informações ao usuário.

O SEAC 2.0 propõe que o fluxo de informações do processador seja ilustrado num ambiente gráfico, que torna simples a visualização de informações através de mudança de cores. O principal objetivo desse simulador é facilitar a aprendizagem de

arquitetura de computadores pelos alunos que estudem a Ciência da Computação e áreas similares. Outra característica que o diferenciará positivamente em relação aos demais simuladores citados nesse trabalho, é a possibilidade de visualizar uma mesma instrução sendo executada com e sem o uso do *pipeline*. O simulador será disponibilizado via Web para facilitar o acesso dos alunos e professores interessados. Também são previstas versões do simulador para as plataformas *Desktop* (Mac OS, Windows) e *mobile* (iOS e Android). Em adição, a criação de uma suite de aplicativos para o ensino de arquitetura de computadores está sendo estudada.

## 7. REFERÊNCIAS

- [1] A. A. Moreira, C. A. P. S. Martins, “R2DSim: Simulador Didático do RISC Reconfigurável”. 2009.
- [2] A. B. Verona, J. A. Martini, T. L. Gonçalves, “SIMAEAC: Um Simulador Acadêmico para Ensino de Arquitetura de Computadores”. 2009.
- [3] A. S. Tanenbaum, “Organização Estruturada de Computadores”, Editora Prentice-Hall. 2006.
- [4] Unity Technologies, "Unity 3D", <http://unity3d.com>. Acessado em 09/09/2013.
- [5] E. L. Falcão, E. V. C. L. Borges, I. L. P. Andrezza, G. S. Silva, B. É. S. Cavalcante, H. S. Silva, "Ambiente de Simulação Gráfica 3D para Ensino da Arquitetura de Processadores", XIX Workshop sobre Educação em Computação, 2011.
- [6] E. L. Falcão, E. V. C. L. Borges, I. L. P. Andrezza, G. S. Silva, B. É. S. Cavalcante, H. S. Silva, "Simulador Gráfico 3D para Ensino da Arquitetura de Processadores", Escola Paraibana de Informática, 2011.
- [7] J. A. S. Borges, G. P. Silva “NeanderWin - Um Simulador Didático para uma Arquitetura do Tipo Acumulador”. WEAC, 2006.
- [8] M. Scott, “WinMips64, version 1.5, School of Computing, Dublin City University, Ireland”, <http://www.computing.dcu.ie/~mike/winmips64.html>. 2006. 20 de Julho de 2009.
- [9] M. Wolff e L. Wills, “SATSim: A Superscalar Architecture Trace Simulator Using Interactive Animation”, WCAE: Workshop On Computer Architecture Education, Vancouver, Canada. 2010
- [10] Oracle Java Applet, "Java Applets", <http://java.sun.com/applets>, 30 de Agosto de 2012.
- [11] R. Zelenovsky, “PC: um guia Prático de Hardware e Interfaceamento”, Editora MZ Produções Culturais. 2010.
- [12] Web3D – Part 2: Scene Access Interface, “Extensible 3D (X3D). Part 2: Scene access interface (SAI). ISO/IEC 197752.2: 2009”, <http://www.web3d.org/x3d/specifications/ISOIECFDIS197752.2X3DSceneAccessInterface/index.html>. Março de 2010.
- [13] Web3D Consortium, “X3D – FAQ (Frequent Asked Questions)”, <http://www.lsi.usp.br/~lsoares/x3d/faq.html>. Março de 2011.
- [14] Z. Sridhar, (2002) “GNUSim8085, versão 1.3”, [http://gnusim8085.sourceforge.net/index.php/Main\\_Page](http://gnusim8085.sourceforge.net/index.php/Main_Page), 23 de Junho de 2009.
- [15] Blender 3D. <http://www.blender.org/>. Acessado em 06/09/2013.