

Simuladores e ferramentas educacionais no ensino de arquitetura de computadores

Luis Antonio Soares
UNESP - IBILCE

1. INTRODUÇÃO

Ao analisar o conteúdo da disciplina de arquitetura de computadores, é possível verificar a extensão de seu conteúdo e a sua complexidade, pois tratamos de assuntos abstratos com objetivo de demonstrar o funcionamento dos processadores de uma maneira geral.

Conhecer as características dos computadores, seu funcionamento, o caminho dos dados, como é a comunicação entre memória e processador, entre outros assuntos, se torna extremamente complexo de se explicar e de se compreender utilizando o método tradicional de ensino, ou seja lousa e giz.

Segundo Elias *et al.* [3] Softwares educacionais são desenvolvidos para auxiliar alunos em várias áreas da computação e outras ciências. Estes softwares são cada vez mais sofisticados e intuitivos e tem por objetivo tentar levar para os alunos mais facilidades nos estudos dos conteúdos de disciplinas, ao proporcionarem testes e práticas de operações que não são facilmente realizadas sem o acesso físico a tais objetos.

Assim a simulação é uma técnica muito utilizada na área de arquitetura de computadores para reproduzir o comportamento de um sistema real. Tal técnica é utilizada para testes e validação de arquiteturas experimentais e para dar suporte ao ensino e aprendizagem de fundamentos de arquitetura de computadores [2].

Como uma ferramenta educacional, é importante que um simulador valorize e enriqueça as experiências de aprendizagem dos alunos e, ao mesmo tempo, facilite a compreensão profunda dos conceitos tecnológicos fundamentais. A ferramenta deve incentivar ativamente experimentação, exploração e resolução de problemas de investigação com os alunos trabalhando individualmente ou em grupos [3].

Assim, tendo como principal objetivo investigar softwares simuladores para o apoio ao ensino de arquitetura de computadores, buscamos uma ferramenta gráfica capaz de mostrar o caminho que as informações percorrem durante todo percurso entre a entrada e a saída do processador.

2. FUNDAMENTAÇÃO TEÓRICA

Para discutir e analisar o tema em questão de maneira educacional, foi considerada a pesquisa bibliográfica como metodologia, a qual limita-se ao

campo teórico já existente sobre o tema, e apresentaremos um novo simulador, que poderá ser adotado por estudantes e professores no ensino da arquitetura de computadores. De acordo com Severino (2009, p. 223)[13]:

É válido aceitar esses tipos de trabalho justamente por permitirem a formação de um material básico de documentação de onde partirão outros estudos interpretativos.

Assim, foi pesquisado vários autores e materiais acadêmicos que tratam do assunto a fim de aprofundarmos o máximo possível o tema desta pesquisa.

3. JUSTIFICATIVA E RELEVÂNCIA DO TEMA DE PESQUISA

A partir de uma rápida pesquisa, em dissertações e teses sobre arquitetura de computadores, notou-se uma grande preocupação com o ensino da mesma, dado seu nível de abstração e a dificuldade dos alunos em colocar em prática toda teoria apresentada pelo professor, assim foi apresentada várias abordagens educacionais para uso na disciplina de arquitetura de computadores. O que justifica o desenvolvimento e o estudo de novas ferramentas educacionais.

Nesse sentido, o uso de simuladores como ferramentas pedagógicas no ensino de arquitetura de computadores é de extrema relevância no meio acadêmico.

3.1. Ferramentas Educacionais no Ensino

Segundo o MEC – Ministério de Educação e cultura em suas diretrizes base para os cursos de computação nos diz que: A metodologia de ensino deve ser centrada no aluno como sujeito da aprendizagem e apoiada no professor como facilitador do processo de ensino-aprendizagem. O professor deve fortalecer o trabalho extraclasse como forma de o aluno aprender a resolver problemas, aprender a aprender, tornar-se independente e criativo[7].

Partindo desse princípio, utilizar simuladores como ferramentas de ensino, traz uma abordagem diferente, pois seu uso ou de outras ferramentas educacionais, traz uma visão do tema mais abrangente, e torna o ensino um pouco menos cansativo.

O educador, para desenvolver a autonomia de seus alunos, precisa realizar atividades que estimulem a tomada de decisão e a responsabilidade para que eles aprendam a decidir com liberdade, assumindo todas as consequências desse ato (5).

O educador precisa antes de tudo mostrar para aluno a necessidade de uma consciência crítica, e não a simples transmissão do saber. Para isso é necessário que existam atividades que estimulem e instigue essa curiosidade

no estudante, que faça com que ele tenha vontade de ir buscar o conhecimento, a problematização do conteúdo, a vontade de aprender.

3.2. Arquitetura de computadores

O MEC – Ministério de Educação e Cultura e a SBC – Sociedade Brasileira de Computação, definem a estruturação e as características curriculares dos cursos de Ciência da Computação, Engenharia da computação entre outros cursos.

As diretrizes curriculares contêm uma estrutura curricular abstrata, organizada de tal forma que as Instituições de Ensino Superior possam, a partir desta, conceber currículos plenos diversificados que atendam as necessidades regionais onde se situam. Esta estrutura serve como base, para que através de detalhamentos sucessivos, chegue-se a um grupo de disciplinas distribuídas ao longo do curso, formando assim um currículo pleno que deve ser executado pelo corpo docente [6].

Podemos dizer então que a arquitetura de computadores visa o conhecimento funcional de um computador, que em si possui muito níveis a ser considerado, o mais alto deles que é onde o usuário executa programas, ao mais baixo que consiste basicamente aos componentes eletrônicos em si.

3.3. Arquitetura RISC

O simulador que será apresentado apresenta fortes características de processadores com arquitetura RISC, por esse motivo descreveremos um pouco sobre esse tipo de equipamento.

A arquitetura RISC é constituída por um pequeno conjunto de instruções simples que são executadas diretamente pelo hardware, sem a intervenção de um interpretador (microcódigo), ou seja, as instruções são executadas em apenas uma microinstrução [8].

Os conjuntos de características básicas de uma arquitetura RISC são[8]:

- o coração de todo computador é o datapath (ULA, registradores e os barramentos que fazem sua conexão); uma das maiores características das máquinas RISC é utilizar apenas uma instrução por ciclo do datapath (uma instrução é similar a uma microinstrução);
- projeto carrega/armazena, ou seja, as referências à memória são feitas por instruções especiais de load/store;
- inexistência de microcódigo; sendo assim, a complexidade está no compilador;

- instruções de formato fixo, permitindo uso consistente do formato e facilitando a decodificação de instruções por controle fixo, o que torna mais rápido os dutos de controle;
- conjunto reduzido de instruções, facilitando a organização da UC de modo que esta tenha uma interpretação simples e rápida;
- utilização de pipeline, uma técnica de dividir a execução de uma instrução em fases ou estágios, abrindo espaço para execução simultânea de múltiplas instruções; à medida que um estágio é concluído, a instrução vai para a frente, no canal, e a instrução seguinte vai para aquele estágio; como as instruções RISC são do mesmo tamanho, elas levam um ciclo para completar cada estágio de pipeline. Por exemplo, se uma instrução pode ser dividida em 4 fases, 4 instruções podem ser executadas simultaneamente;
- utilização de múltiplos conjuntos de registradores.

Sendo assim, o principal objetivo de uma máquina de arquitetura RISC é executar uma instrução por ciclo; como o acesso a memória utiliza mais ciclos, a solução foi criar vários registradores. E é esse número elevado de registradores que impacta profundamente no desempenho de máquinas RISC.

4. Trabalhos Relacionados

Na literatura encontram-se diversas ferramentas desenvolvidas para o ensino de arquitetura de computadores, que nos mostram diferentes tipos de processadores e seu funcionamento interno de forma interativa.

Podemos dizer então que os simuladores computacionais são recursos tecnológicos que podem auxiliar em diversas áreas. No contexto abordado neste trabalho, os simuladores auxiliam os professores a provocar, favorecer e orientar situações controladas de ensino e aprendizagem, pedagogicamente interessantes para o nível de ensino que deseja [5].

Dentre os diversos tipos de simuladores existentes hoje no ensino da arquitetura de computadores podemos citar:

Simulador	Características
NeanderWin	É um simulador da máquina Neander, definida no livro do Raul F. Weber (UFRGS), Fundamentos de Arquitetura de Computadores, Ed. Sagra Luzzatto[10].
ProcSIM	Permite visualizar a simulação de circuitos internos, e executar código de montagem, é usado principalmente para

MARS aprender como o processador funciona[11].
Permite a interação e desenvolvimento de códigos em linguagem de máquina(programação Assembly) é utilizado por Patterson and Hennessy's *Computer Organization and Design*[12].

Tabela 1 - Lista de alguns simuladores para disciplina de Arquitetura de Computadores

Após analisar diversos simuladores, mostraremos alguns recursos do CPU-OS simulator, que foi projetado por Besim Mustafá [3] na Edge Hill University.

O autor relata a importância de se ter um simulador quando nos diz que em um ambiente de simulação é essencial ser capaz de suspender, parar ou reiniciar as simulações de algum evento em um estado predeterminado. Também pode ser necessário fazer manualmente as alterações nos componentes do sistema (memórias, registros, instruções, etc.) [3]

O CPU-OS simulator, possui apresentações visuais animadas e em tempo real, com cores e displays visuais, que fornecem feedback imediato que mostram as transições de estado o que facilita em muito as comparações, o sistema tem suporte também para aprendizagem baseada em problemas, que é um método que estimula o aprendizado do aluno. Possui suporte para funcionalidades avançadas: vários processadores, níveis de instruções em pipeline, otimizações do compilador e virtualização [3].

5. CPU-OS simulator

A interface gráfica do CPU-OS simulator, nos permite visualizar os recursos disponíveis para estudo de processadores, baseado em arquiteturas RISC, ele possui um pipeline de 5 estágios, simuladores de cache, ISA, que são utilizadas no ensino introdutório de arquitetura de computadores.

A tela inicial é apresentada abaixo e nos mostra os principais recursos do simulador.

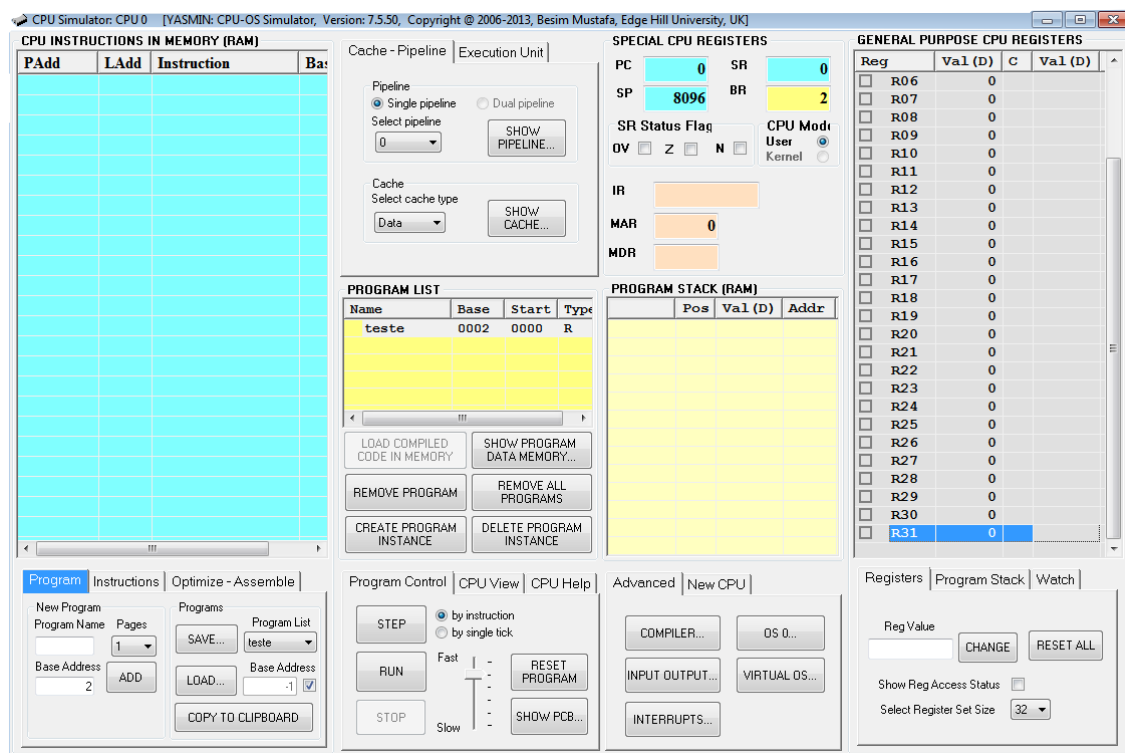


Imagem 1 – Tela principal do simulador (CPU-OS simulator)

Mostraremos as diferentes funcionalidades desse simulador e quais suas principais características através das imagens a seguir.

Na **imagem 2**, vemos as instruções de memória. Para tornar o código legível é apresentado mnemônico de instruções de baixo nível (assembly). Em cada instrução é associado um endereço físico (PADD) e o endereço lógico (Ladd). O endereço de base e a sequência de instruções pertencentes a cada programa.

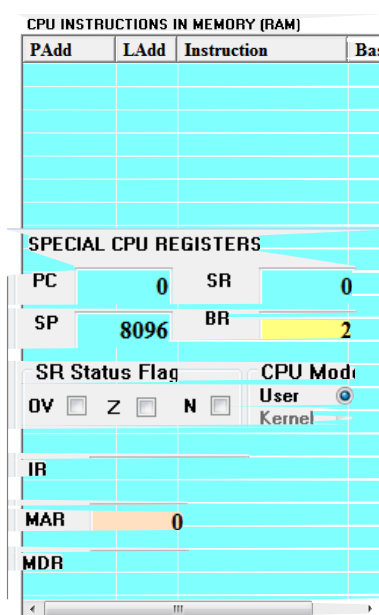
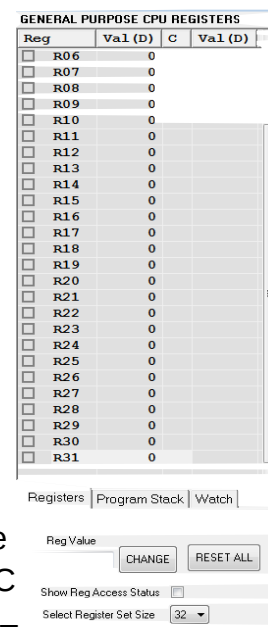


Imagem 2. – CPU memory
Imagem 3 – Special CPU Registers
Imagem 4 - Registradores



Na **imagem 3**, observamos as instruções e o conjunto de registradores da CPU como PC – contador de programas, IR – Registro de instrução, SR – registro de status, SP –

ponteiro de pilha, BR – Base de registro, MAR – Endereço de registro de memória.

Na imagem 4, visualizamos o conjunto de registradores.

PROGRAM STACK (RAM)

	Pos	Val (D)	Addr
--	-----	---------	------

Imagem 5 – Lista de Pilhas

A imagem 5, apresenta as pilhas do programas, que é frequentemente utilizado no tratamento das interrupções.

Na imagem 6, é listado os programas que estão em execução, podemos remover um programa ou todos os programas selecionados.

Na imagem 7, observamos a aba para criação de programas. Para criar um novo programa basta inserir um nome para seu programa e um endereço e clicar no botão adicionar. O nome do novo programa será exibido no modo de exibição da lista de programas.

PROGRAM LIST

Imagem 6 – Lista dos programas

LOAD COMPILED CODE IN
MEMORY

SHOW PROGRAM DATA
MEMORY...

REMOVE PROGRAM

REMOVE ALL PROGRAMS

CREATE PROGRAM INSTANCE

DELETE PROGRAM INSTANCE

Program

Instructions

Optimize - Assemble

New Program

Program Name

Pages

1

Base Address

2

ADD

Programs

Program List

SAVE...

ALL

LOAD...

Base Address

-1

☒

COPY TO CLIPBOARD

Imagem 7 – Criando programas.

Na imagem 8, podemos criar, editar e mover instruções ao nosso programa.

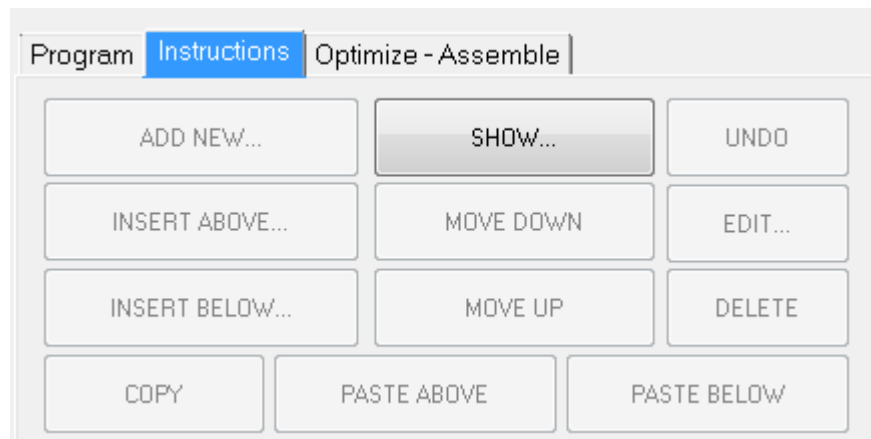


Imagem 8 – Janela de Instruções.

Disponibilizar ferramentas para que o estudante torne o abstrato em concreto é o que torna interessante o estudo e aplicação de simuladores nas disciplinas de arquitetura de computadores, com o CPU-OS simulator, os estudantes terão uma importante ferramenta de apoio pedagógico. Ele de forma alguma irá substituir a figura do professor, pois esse será o responsável por conduzir todo o processo de ensino, mas facilitará na aprendizagem.

Conclusões

A proposta deste estudo é a apresentação de uma nova ferramenta que permite dar opções pedagógicas de ensino que visam:

- Ser um meio de interação entre os conteúdos apresentado aos alunos;
- Ser um mecanismo prático e interativo para exposição dos tópicos da disciplina por parte do professor;
- Observar detalhes das operações da CPU com múltiplos níveis de abstração;
- Explorar o ensino do conteúdo: memórias e registradores, mais naturalmente.

Como é possível notar, o simulador apresentado possui várias ferramentas visuais de suporte pedagógico ao aprendizado de arquitetura de computadores, com uma interface gráfica relativamente simples. Permite ao aluno visualizar o que ocorre com o processador em cada uma de suas fases.

Como possíveis trabalhos futuros, pretende-se avaliar a usabilidade da ferramenta com alunos e professores das disciplinas de arquitetura de computadores, visando assim obter um estudo quantitativo feito por meio de questionários.

Referencias Bibliográficas

- [1] RODRIGUES, Rafael Porto; MARTINS, Carlos Augusto Paiva da Silva. Ensino e aprendizado de pipeline de modo motivante e eficiente utilizando simuladores didáticos, WORKSHOP SOBRE EDUCAÇÃO EM ARQUITETURA DE COMPUTADORES - WEAC, p. 2008.
- [2] JUNIOR, Nelson A. Gonçalves; et. al. R10k: Um simulador de de Arquitetura Superescalar, Workshop sobre Educação em Arquitetura de Computadores (WEAC), p. 23, 2007.
- [3] Mustafa, Besim. YASS: A System Simulator for Operating System and Computer Architecture Teaching and Learning, Proceedings of the Frontiers in Science Education Research Conference, Famagusta, North Cyprus., 2009.
- [4] Freire, Paulo. Pedagogia da autonomia: saberes necessários à prática educativa. 35. ed. São Paulo: Paz e Terra; 1996.
- [5] TORRES, André L.L.; BRITO, Alisson V. Extensão do Ptolemy para o ensino de Organização e Arquitetura de Computadores, workshop sobre educação em arquitetura de computadores - WEAC, p.13. 2011.
- [6] LOURENÇO, Andre E.; MIDORIKAWA, Edson T. Ensino de arquitetura de computadores utilizando simuladores completos. Congresso Brasileiro de Engenharia - COBENGE 2004, Brasília, 2004.
- [7] MINISTÉRIO DA EDUCAÇÃO E CULTURA. Diretrizes Curriculares Nacionais para os cursos de graduação em Computação.. Disponível em: http://portal.mec.gov.br/index.php?option=com_docman&task=doc_download&gid=11205&Itemid= Acesso em: 15 de maio 2015.
- [8] SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. *Currículo de Referência da SBC para cursos de Graduação em Computação*. Versão 1999. versão de consulta pública 2011. http://www.sbc.org.br/index.php?option=com_jdownloads&Itemid=195&task=finish&cid=186&catid=36. Acesso em: 15 maio 2015.
- [9] MATTOS, N.P., RASKIN, S.F. *Aspectos da Arquitetura de processadores RISC*. Companhia de Informática do Paraná – CELEPAR. Disponível em: <http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1259>>. Acesso em: 17 maio 2015.
- [10] BORGES, Jose Antonio. SILVA, Gabriel P. Neanderwin. Disponível em: <http://www.dcc.ufrj.br/~gabriel/neander.php>. Acesso em 18 maio 2015.
- [11] ProcSIM. A Visual MIPS R2000 Processor Simulator – Freeware. Disponível em: <http://jamesgart.com/procsim>. Acesso em: 19 de maio 2015.
- [12] MARS. MARS – Mips Assembly and Runtime Simulator. Disponível em: <http://courses.missouristate.edu/KenVollmar/MARS>. Acesso em: 19 de maio de 2015.
- [13] SEVERINO, Antonio Joaquim. Metodologia do trabalho científico. São Paulo: Ed. Cortez; 23ª ed; 2009.