

FACULDADES INTEGRADAS DE CARATINGA

FACULDADE DE CIÊNCIA DA COMPUTAÇÃO

MIPS – PMS: IMPLEMENTAÇÃO DE UM SIMULADOR  
MULTICICLO DO PROCESSADOR MIPS 32 BITS PARA  
APOIO AO ESTUDO EM ARQUITETURA DE  
COMPUTADORES

WARLEY JACINTO ELIAS

CARATINGA

2010

**Warley Jacinto Elias**

**MIPS – PMS: IMPLEMENTAÇÃO DE UM SIMULADOR MULTICICLO DO  
PROCESSADOR MIPS 32 BITS PARA APOIO AO ESTUDO EM ARQUITETURA  
DE COMPUTADORES**

Monografia apresentada ao Curso de  
Ciência da Computação da Faculdades  
Integradas de Caratinga como requisito  
parcial para obtenção do título de  
Bacharel em Ciência da Computação  
orientada pelo Prof. Jacson Rodrigues  
Correia da Silva.

**CARATINGA**

**2010**

**Warley Jacinto Elias**

MIPS – PMS: IMPLEMENTAÇÃO DE UM SIMULADOR MULTICICLO DO  
PROCESSADOR MIPS 32 BITS PARA APOIO AO ESTUDO EM ARQUITETURA  
DE COMPUTADORES

Monografia submetida à Comissão  
examinadora designada pelo Curso de  
Graduação em Ciência da Computação  
como requisito para obtenção do grau de  
Bacharel.

---

Prof. Jacson Rodrigues Correia Silva

Faculdades Integradas de Caratinga

---

Prof. Fabrícia Pires Tiola

Faculdades Integradas de Caratinga

---

Prof. Zamith França Neto

Faculdades Integradas de Caratinga

Caratinga, 09/12/2010

## **AGRADECIMENTOS**

Ó Deus, o que seria de mim sem a fé que tenho nele. Aos meus pais, irmãos, e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Ao professor e orientador Jacson R. Correia da Silva por seu apoio e inspiração no amadurecimento dos meus conhecimentos e conceitos que me levaram a execução e conclusão desta monografia. À professora e coordenadora do Curso de Ciência da Computação, Fabrícia Pires Tiola, pelo convívio, apoio, compreensão e amizade.

Aos amigos e colegas de classe, em especial, Simone Dornelas, pelo incentivo e pelo apoio constantes.

A todos meus amigos e colegas que estiveram presentes nessa jornada, me apoiando em cada degrau dessa subida.

A todos que me ajudaram na conclusão desta monografia gostaria de deixar aqui meu agradecimento.

## RESUMO

O ensino da arquitetura MIPS (*Microprocessor Without Interlocked Pipeline Stages*) na disciplina de Arquitetura de Computadores tem como objetivo introduzir o conceito do funcionamento dos processadores em geral. É utilizado por ser uma arquitetura simples, o que facilita na compreensão do caminho de dados e do controle de um processador, os métodos tradicionais de ensino, na maioria das vezes, podem ser desestimulantes para o aluno. Softwares educacionais são desenvolvidos para auxiliar alunos em várias áreas da computação e outros cursos. Estes softwares são cada vez mais sofisticados e mais intuitivos e tem por objetivo tentar levar para os alunos mais absorção do conteúdo da disciplina. Algumas áreas da computação ainda estão defasadas em material didático e softwares para auxiliar no aprendizado das mesmas. Isso traz para os alunos algumas dificuldades para assimilar o conteúdo, o que às vezes faz com que esses se desinteressem pela disciplina.

Este trabalho apresenta a implementação de uma ferramenta de simulação da arquitetura MIPS Multiciclo, que permite aos estudantes da disciplina de Arquitetura de Computadores sua utilização para complementar o material de didático disponibilizado pelo professor. O Simulador foi desenvolvido em JAVA, pensando na execução do software educacional em multiplataformas operacionais. Além disso, o software foi devidamente comentado para possíveis modificações e melhorias.

O trabalho desenvolvido também apresenta uma avaliação realizada com diversos alunos do Curso de Ciência da Computação utilizado para aferir as funcionalidades e design do software desenvolvido. Nesta avaliação o simulador MIPS – PMS se mostrou com um ótimo desempenho e uma aceitação de utilização de 100% nos quais os alunos afirmaram que utilizariam o simulador como complemento de estudos de Arquitetura de Computadores.

Palavras Chaves: MIPS, Softwares Educacionais e MIPS Multiciclo.

## ABSTRACT

MIPS (Microprocessor Without Interlocked Pipeline Stages) architecture teaching in Computers Architecture subject, has as its aim to introduce processor's performance concept by and large. It is used owing to be a simple architecture which makes easier comprehension on data's track and processor's control. Traditional teaching methods, most of the time, can be unexciting for the students. Educational software are developed to help students in several computing areas and other courses. Such software are even more sophisticated and more intuitive; and their aim is try to take to students more assimilation on subject's content. Some of the computing areas are still lagged in didactical material and software to aid their own learning. This may convey to students many difficulties to understand content, what makes them lose the interest on the discipline.

This paper presents MIPS Multicycle architecture simulation tool implementing, which allows its use to Computing Architecture discipline students in order to complete didactical material available by the teacher. The Simulator was developed in JAVA because it was thought to be executed joint with .educational software in operational multiplatform. Besides this, the software was appropriately commented to possible changes and improvements.

The developed paper also presents an assessment made with several Computing Science Course students using it to check the own software features and design. On this assessment MPS-PMS Simulator had an excellent performance assessment with a 100% acceptance for use when students claimed they have used it as a completion for the Computing Architecture studies.

Key-words: MIPS, Educational Software, MIPS Multicycle.

## LISTA DE ILUSTRAÇÕES

Figura 2.2.2.1. Caminho de dados completo para a implementação multíciclo.....	19
Figura 2.3.2.1. Máquina de estados finitos.....	23
Figura 3.3.1. Botões que compõe o layout do simulador MIPS – PMS .....	29
Figura 3.3.2. Painéis que contém os valores dos registradores temporários, banco de registradores e memória presentes no layout do simulador MIPS – PMS. ....	30
Figura 3.4.1. Layout do simulador MIPS – PMS.....	33
Figura 4.1. Gráfico dos dados referentes às questões um, dois e três .....	37
Figura 4.2. Gráfico dos dados referente às questões quatro, cinco e seis.....	39
Figura 4.3. Gráfico referente às questões oito e nove.....	40
Figura 4.4. Gráfico referente às questões dez, onze e doze.....	41
Figura 4.5. Gráfico referente a questão sete.....	43

## LISTA DE TABELAS

Tabela 4.1. Dados dos testes feitos com o simulador MIPS – PMS.....	36
Tabela 4.2. Tabela contendo os dados referentes à questão sete .....	42



## LISTA DE SIGLAS

ALU	Arithmetic Logic Unity (Unidade Lógica e Aritmética)
CPU	Central de Processamento Único
FIC	Faculdades Integradas de Caratinga
MDR	Registrador de Dados da Memória
MIPS	Microprocessor Without Interlocked Pipeline Stages (Microprocessador sem Estágios Interligados de Pipeline)
PC	Program Counter (Contador de Programa)
RISC	Reduce Instruction Set Computer (Computador com um Conjunto Reduzido de Instruções)
SBC	Sociedade Brasileira de Computação
WEAC	Workshop sobre Educação em Arquitetura de Computadores
WEI	Workshop sobre Educação em Computação

## SUMÁRIO

1.	INTRODUÇÃO .....	11
2.	REFERENCIAL TEÓRICO .....	14
2.1.	ARQUITETURA DE COMPUTADORES .....	14
2.2.	MIPS .....	15
2.2.1.	A Arquitetura Monociclo .....	16
2.2.2.	A Arquitetura Multiciclo .....	17
2.3.	CONTROLE DO CAMINHO DE DADOS DO MIPS .....	20
2.3.1.	Controle Utilizando Máquina de Estados Finitos .....	21
2.3.2.	Controle Utilizando Microprogramação .....	24
2.4.	SOFTWARES EDUCACIONAIS .....	24
3.	METODOLOGIA .....	27
3.1.	PARA O DESENVOLVIMENTO .....	27
3.2.	INSTALAÇÃO .....	28
3.3.	FUNCIONALIDADES E INTERFACE .....	29
3.4.	UTILIZANDO O SIMULADOR .....	32
3.5.	PARA A AVALIAÇÃO .....	34
4.	RESULTADOS .....	36
5.	CONCLUSÃO .....	45
6.	TRABALHOS FUTUROS .....	47
	REFERÊNCIAS .....	48
	ANEXO 1 – QUESTIONÁRIO .....	51

## 1. INTRODUÇÃO

O estudo em arquitetura de computadores pode ser fácil e promissor para o aluno através da utilização de Softwares Educacionais como os simuladores, que permitem a abstração de detalhes. Dentre os simuladores existentes temos o DIMIPSS proposto por Felix, Pousa e Carvalho em 2006, o Ultimate Pipeline Simulator proposto por Barcellos e Simões em 2009, o MPDSim proposto por Avelar et al em 2008, o PS – CAS MIPS proposto por Maia, Vieira e Pessoa em 2009 e o SIPPIS MIPS proposto por Consendey, Brasil e Martins em 2009.

Softwares Educacionais segundo Lucena (1992) é todo aquele programa que possa ser usado para algum objetivo educacional, pedagogicamente defensável, por professores e alunos, qualquer que seja a natureza ou finalidade para o qual tenha sido criado.

Normalmente, os professores utilizam métodos pouco atrativos como desenhos no quadro, apresentações de *slides*, ou mesmo recorrerem ao auxílio de livros didáticos, como por exemplo, o livro Organização e Projeto de Computadores, escrito por Patterson e Hennessy em 2005, fazendo das aulas de arquitetura de computadores rotineiras e até mesmo cansativas.

Com a utilização de simuladores o aluno aprenderá desde conceitos básicos até tópicos mais avançados, conhecendo todos os componentes do processador da arquitetura MIPS e interagindo com cada um deles com mais naturalidade. Neste contexto, foram desenvolvidos e propostos vários simuladores cujo objetivo é permitir ao estudante, através de um software com uma interface gráfica visualizar e aprender os conceitos estudados.

O *MIPS – PMS* – Simulator Multicycle Processor – foi desenvolvido para simular a execução (caminho de dados e de controle) das instruções do MIPS

Multiciclo, tentando assim preencher a lacuna deixada pelos outros simuladores que deixam de tratar o MIPS Multiciclo.

Na implementação deste Software Educacional foi utilizado o NetBeans, um ambiente de desenvolvimento integrado e multiplataforma que consta em NETBEANSORG (2010), de código aberto e que contem uma plataforma de aplicativos que permite aos desenvolvedores criar rapidamente aplicativos diversos utilizando diversas linguagens. Foi utilizada também a plataforma *Java* que é uma linguagem de programação orientada a objeto, multiplataforma e de simples utilização.

O MIPS é um microprocessador, desenvolvido pela *MIPS Computer System Inc.* e disponível em MIPS (2010), que utiliza um conjunto de instruções de 32 bits e possui arquitetura RISC (*Reduce Instruction Set Computer*). De acordo com Felix, Pousa e Carvalho (2006) “por ser simples, o MIPS é adotado em livros didáticos”, como o de Patterson e Hennessy em 2005 e em várias instituições de ensino.

O objetivo deste trabalho foi desenvolver um software educacional capaz de simular as instruções *assembly* do MIPS Multiciclo, representando graficamente os componentes do caminho de dados e de controle, com intuito de auxiliar no ensino da disciplina de Arquitetura de Computadores. É um software que irá auxiliar tanto os alunos quanto os professores nas aulas dessa disciplina.

Para alcançar este objetivo foi realizado um estudo sobre linguagens de programação verificando qual a melhor linguagem para tal desenvolvimento. Analisou-se também simuladores já existentes que trouxeram conceitos básicos que ajudaram no desenvolvimento do simulador. Além disso, definiu-se o conceito de software educacional onde se localizou pontos mais importantes que um software educacional necessita para trazer benefícios para quem o utilizar, sendo um deles a interface agradável. Para realizar os testes, alguns alunos foram selecionados de um grupo, para utilizar o software simulador e responder questionários específicos. Logo após foi realizada uma análise definindo as conclusões.

O questionário foi respondido por 15 alunos, onde os mesmo avaliaram, no geral, o simulador como bom para ser utilizado como apoio ao estudo de Arquitetura

de Computadores. Na avaliação dos gráficos gerados a partir do questionário nota-se que 100% dos alunos afirmaram que utilizariam o simulador em seus estudos.

## 2. REFERENCIAL TEÓRICO

### 2.1. ARQUITETURA DE COMPUTADORES

No mundo globalizado tem-se observado uma atuação cada vez maior dos computadores nas diversas atividades do nosso dia a dia. As operações bancárias, as telecomunicações e o manuseio de muitos aparelhos eletrodomésticos são exemplos claros das facilidades trazidas pela utilização dos computadores. Isto sem falar em aplicações mais clássicas, como os sistemas de reservas de passagens aéreas e a previsão meteorológica. Toda essa atuação dos computadores causa um avanço acelerado na área da computação.

Como afirma Patterson e Hennessy (2005) “os computadores levaram a humanidade a enfrentar uma revolução, a revolução da informação, que assumiu seu lugar junto das revoluções industriais e agrícola.”

Com essa grande ascensão tecnológica a arquitetura de computadores foi estudada e melhorada. A evolução da informática foi caracterizada pelo desenvolvimento de computadores com características diversas, traduzidas pelos diferentes parâmetros, cada vez mais conhecidos da maioria de usuários de computador: a *CPU* adotada, a capacidade de memória, a capacidade do disco rígido, a existência de memória *cache* e outros menos conhecidos. A definição destes parâmetros e a forma como os diversos componentes de um computador são organizados, define aquilo que é conhecido por *arquitetura de computador* e vai determinar aspectos relacionados à qualidade, ao desempenho e à aplicação para a qual o computador vai ser orientado.

Neste trabalho é abordado sobre a *CPU*, exclusivamente sobre a arquitetura da CPU MIPS Multiciclo, que por possuir todos os componentes básicos dos processadores e ser simples é utilizado em livros didáticos para o ensino da disciplina de Arquitetura de Computadores.

## 2.2. MIPS

O MIPS é uma arquitetura de processadores RISC desenvolvida pela MIPS Computer Systems. Os processadores MIPS são usados em aplicações tais como computadores da Silicon Graphics, sistemas embarcados, Roteadores da Cisco, videogames como Nintendo 64 e PlayStation. Por ser simples, o MIPS é adotado em livros didáticos como o de Patterson e Hennessy em 2005.

O MIPS é implementado sobre uma determinada arquitetura chamada *load-store*, onde apenas as instruções de manipulação de dados têm acesso à memória, todas as outras instruções operam apenas sobre valores armazenados em registradores.

Segundo Patterson e Hennessy (2000) o conjunto de instruções do MIPS divide-se em dez categorias, sendo estas instruções lógicas e aritméticas, de manipulação de constantes, desvio condicional, desvio incondicionais, de carga, de armazenamento, de movimento de dados, de ponto flutuante e instruções para o tratamento de exceções e interrupções.

Além dessa divisão em categorias as instruções do MIPS dividem-se em instruções do tipo R, tipo I e tipo J, as instruções são divididas de acordo com os campos que cada uma possui. As instruções do tipo R possuem os campos de operação básica da instrução ou *opcode (op)*, registrador do primeiro operando de origem (*rs*), registrador do segundo operando de origem (*rt*), registrador de destino (*rd*), quantidade de deslocamento (*shamt*) e o campo código da função (*funct*). Já as instruções do tipo I possuem os campos *op*, *rs*, *rt*, e o campo de endereço. Por último as instruções do tipo J possuem apenas os campos (*op*) e endereço.

O software que foi desenvolvido para simulação da arquitetura MIPS Multiciclo, o MIPS – PMS, utilizará as principais instruções das categorias citadas, sendo essas instruções *add*, *sub*, *and*, *or*, *slt*, *addi*, *subi*, *lw* e *sw*.

As instruções *add*, *sub*, *and*, *or* e *slt*, realizam operações com dois registradores e salvam o resultado em um terceiro registrador. A instrução de *add*

realiza uma soma, *sub* realiza uma subtração, *and* realiza uma comparação *AND* (e) lógico, *or* realiza uma operação de *OR* (*ou*) lógico e *slt* realiza uma comparação onde se o valor do primeiro registrador for menor que o do segundo o registrador de destino ficará igual a um, caso contrário o valor será zero.

Já as instruções *addi* e *subi* utilizam um registrador e uma constante para realizar suas operações e salvar o resultado obtido em um segundo registrador. A instrução *addi* soma o conteúdo de um registrador com o valor da constante e salva o resultado em um terceiro registrador. Já a operação *subi* realiza uma subtração com o conteúdo de um registrador com o valor da constante e salva no segundo registrador.

Por último as instruções *lw* e *sw* utilizam um registrador e um endereço alvo. A instrução *lw* salva no registrador os trinta e dois bits referentes ao conteúdo armazenado no endereço.

### **2.2.1. A Arquitetura Monociclo**

A arquitetura MIPS Monociclo consiste basicamente em um processador que executa uma instrução em um único ciclo de clock. O que implica que o clock dessa arquitetura seja determinado pelo tempo da instrução que leva mais tempo para ser executada. Isso implica que mesmo que existam instruções que poderiam levar um tempo menor para executar, as mesmas terão que utilizar o tempo que for predefinido de acordo com a instrução mais demorada, gerando perdas de processamento.

Além do tempo existe outro fator que torna o MIPS monociclo ineficiente que é a utilização de vários componentes extras, como por exemplo, alguns dos



somadores utilizados além da *ALU* o que tornam esta arquitetura um tanto quanto mais cara que a arquitetura Multiciclo. Isso acontece pois cada unidade funcional só pode ser usada um vez por ciclo de clock para não haver perdas de dados.

### 2.2.2. A Arquitetura Multiciclo

Na arquitetura multiciclo as instruções são divididas claramente em etapas onde cada etapa executa em um ciclo de relógio e este período é determinado pela etapa mais lenta. Cada instrução executa em um número apropriado de ciclos de relógio (proporcional ao número de etapas). Com isso as instruções que possuem menor quantidade de etapas em sua execução ganham em desempenho da arquitetura monociclo.

As principais vantagens da implementação multiciclo são a possibilidade de fazer com que instruções sejam executadas em quantidades diferentes de períodos do clock, e a capacidade de compartilhar unidades funcionais dentro do espaço de tempo necessário à execução de uma única instrução (Patterson; Hennessy, 2000).

Para o desenvolvimento de uma arquitetura MIPS multiciclo deve-se pensar em quais passos uma instrução necessita para sua execução, quais estruturas necessita para realizar suas operações e quanto tempo leva para completar a execução. Com base nessas informações pode-se “quebrar” a execução das instruções em ciclos de *clock*, com o objetivo de balancear a quantidade de trabalho realizado em cada ciclo, de modo a minimizar o tamanho do ciclo de *clock* e agilizar

a execução das instruções que necessitem uma menor quantidade de ciclos para concluir.

Para quebrar a execução das instruções precisa-se pensar que qualquer dado que for necessário a um ciclo subsequente deve ser armazenado em um elemento de estado, pois sem esse armazenamento perdem-se tudo o que o processador já fez referente à instrução que está sendo executada.

Cada instrução na arquitetura MIPS multiciclo precisa de, no mínimo, três e, no máximo, cinco passos para ser executada. Estes passos são:

- **Passo da busca da instrução:** o processador MIPS busca a instrução na memória e a salva em um registrador de estado, ao mesmo tempo em que a instrução é salva o processador calcula o endereço da instrução seguinte.
- **Passo de decodificação da instrução e busca do registrador:** nesta etapa ainda não se sabe com qual instrução o processador está lidando, mas pode-se realizar algumas funções em paralelo, como ler os dois registradores indicados pelo campo *rs* e *rt* presentes na instrução e armazená-los nos registradores de estado A e B. Podemos também calcular o endereço de desvio com a *ALU* e salvá-lo em *SaídaALU*.
- **Execução, cálculo do endereço de memória ou conclusão do desvio condicional:** se a operação for de desvio incondicional o valor de *PC* é substituído pelo valor do desvio incondicional. Se a operação for desvio condicional, a *ALU* é utilizada para realizar a verificação do conteúdo dos registradores lidos no passo anterior. Esta verificação constata se os registradores são ou não iguais, verificando o sinal Zero da *ALU*. Logo após essa verificação termina-se o desvio condicional, gravando o valor do deslocamento, que foi calculado na etapa anterior e está salvo no registrador temporário, no registrador *PC*. Isto tudo só acontece se a condição do desvio condicional for satisfeita. Se a operação for de referência a memória, utiliza-se a *ALU* para calcular o endereço de mesma e a salva no registrador *SaídaALU*. Se operação for de instrução lógica ou aritmética, utiliza-se a *ALU*

para efetuar a operação aritmética ou lógica. Em seguida salva o resultado no registrador temporário *SaídaALU*.

- **Final da execução das instruções de acesso à memória e do tipo R:** se a instrução é do tipo R, o valor do registrador temporário *SaídaALU* será no registrador especificado na instrução no banco de registradores e termina a execução de uma instrução do tipo R. Se a instrução for de referência à memória, faze-se acesso à mesma com o valor presente em *SaídaALU*. Para a leitura, o endereço acessado é escrito no registrador de dados da memória (MDR). Para escrita, o valor é escrito na memória e fim da execução da instrução *store word*.
- **Final do passo de leitura da memória:** as instruções de *load word* são finalizadas realizando a escrita do valor lido da memória e que está no registrador temporário no banco de registradores.

Na Figura 2.2.2.1 pode-se observar o caminho de dados completo para a implementação multiciclo contendo o PC, a memória, o banco de registrador, a ALU e os registradores de transição. Além destes componentes têm-se as linhas de controle necessárias ao seu funcionamento.

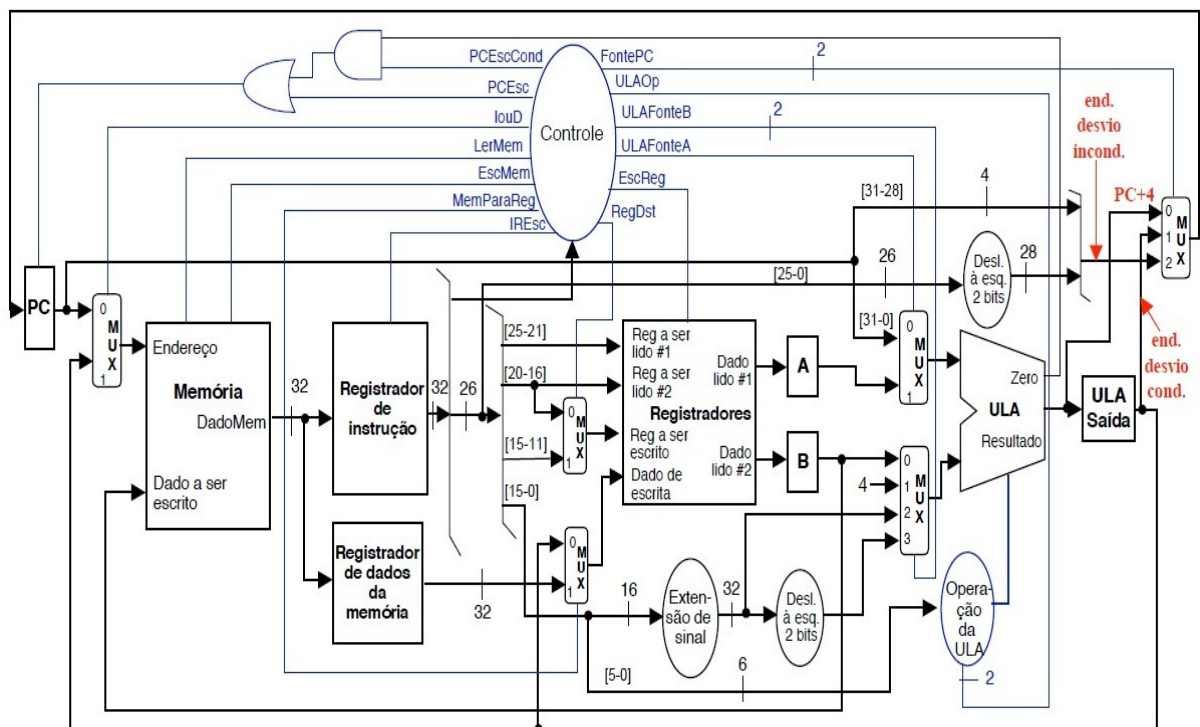


Figura 2.2.2.1. Caminho de dados completo para a implementação multiciclo. Fonte: (Güntzel, 2010).

A Figura 2.2.2.1 detalha o caminho de dados completo para uma implementação da arquitetura MIPS Multiciclo onde as linhas azuis indicam sinais de controle utilizados para controlar escrita em diversos componentes. Quais entradas e quais saídas utilizar e em qual momento utilizar. As linhas em preto são por onde os dados a serem utilizados como valores, endereços de instruções, endereços de dados e outros dados passam para serem utilizados pelos componentes da arquitetura. Ambas as linhas e componentes trabalham juntamente e sincronizados para obter os resultados esperados.

### **2.3. CONTROLE DO CAMINHO DE DADOS DO MIPS**

A unidade de controle do caminho de dados realiza, a partir dos sinais de entrada, a geração dos sinais de escrita de todos os elementos de estados, sinais de controle dos multiplexadores e os sinais de controle da ALU. A ALU possui três entradas que criam oito possíveis combinações das mesmas, porém apenas cinco destas oito combinações são utilizadas. Dependendo da classe de instruções a ALU executa uma destas cinco funções.

A implementação do controle do caminho de dados é definida utilizando uma máquina de estados finitos, que consiste em um conjunto de estados e diretrizes indicando quais ações devem ser realizadas no estado presente e qual o próximo estado a ser executado, ou através de microprogramação que utiliza uma representação de programação para o controle. A microprogramação é um programa composto de estruturas mais simples denominadas microinstruções, onde idéia básica é representar simbolicamente os valores ativos das linhas de controle, de modo que o microprograma seja uma representação das microinstruções.

Segundo Patterson e Hennessy (2005) “a microprogramação é usada hoje para implementar algumas partes de um conjunto de instruções complexo, como um Pentium, bem como em processadores de uso especial”.

O controle de máquina de estado é viável apenas se o subconjunto de instruções a serem utilizadas for simples, englobando apenas as principais instruções do MIPS. O conjunto de instruções completo do MIPS possui cerca de 100 instruções, o que torna inviável a utilização da implementação do controle baseada em máquina de estados, já que especificar o controle contendo todas essas instruções em forma gráfica será extremamente incômodo, uma vez que a máquina de estados pode vir a ter centenas ou milhares de estados.

Ambos os controles selecionam quais as saídas dos multiplexadores serão usadas, quais sinais de controle estarão ativos, quais registradores poderão ser escritos, qual a próxima instrução a ser gravada no registrador PC. Enfim, os controles definem todos os sinais para realizar o controle do caminho de dados, afim de executar perfeitamente todas as instruções para o qual o mesmo foi programado.

### **2.3.1. Controle Utilizando Máquina de Estados Finitos**

Este método consiste na definição de uma série de estados e na definição de regras para a transição entre os estados. As regras são definidas pela função próxima estado, que realiza o mapeamento do estado atual e as entradas do novo estado.

De acordo com Patterson e Hennessy (2005) “ao usar uma máquina de estados finitos para especificar o controle, cada estado também define um conjunto de saídas que estarão ativas quando a máquina estiver em tal estado”. As saídas

que não estiverem explicitamente ativas, deverão ser consideradas inativas por padrão.

Cada um dos estados da máquina de estados finitos corresponde a um dos passos da execução de uma instrução, portanto cada estado utilizará de um ciclo de *clock* para completar sua execução.

Considerando que os dois primeiros passos são iguais para todas as instruções do MIPS, os dois primeiros estados da máquina também serão iguais para qualquer que seja a instrução. Já os passos de três a cinco se diferem e dependem do *opcode*, onde para cada instrução existe um estado específico, contendo os sinais de controle para a execução do estado. Após o último estado da máquina, qualquer instrução retorna ao estado inicial. Na Figura 2.3.1.1 encontra-se o modelo da máquina de estados finitos para as instruções do tipo R, tipo J e as instruções *lw*, *sw* e *beq*, definindo quais estados uma instrução necessita para completar sua execução.

A Figura 2.3.1.1 demonstra também, detalhadamente quais passos uma instrução do tipo *lw*, *sw*, *beq*, tipo *R* e tipo *J* necessitam para sua execução ser realizada com sucesso. Como se pode observar, todas as instruções possuem os dois primeiros passos em comum, pois a busca da instrução na memória de instrução é realizada da mesma maneira para todas as instruções, e o passo de busca de registradores e decodificação da instrução também. Já os outros passos para cada tipo de instrução necessitam de uma configuração diferente nos sinais de controle para que toda execução grave e leia os dados dos componentes certos.

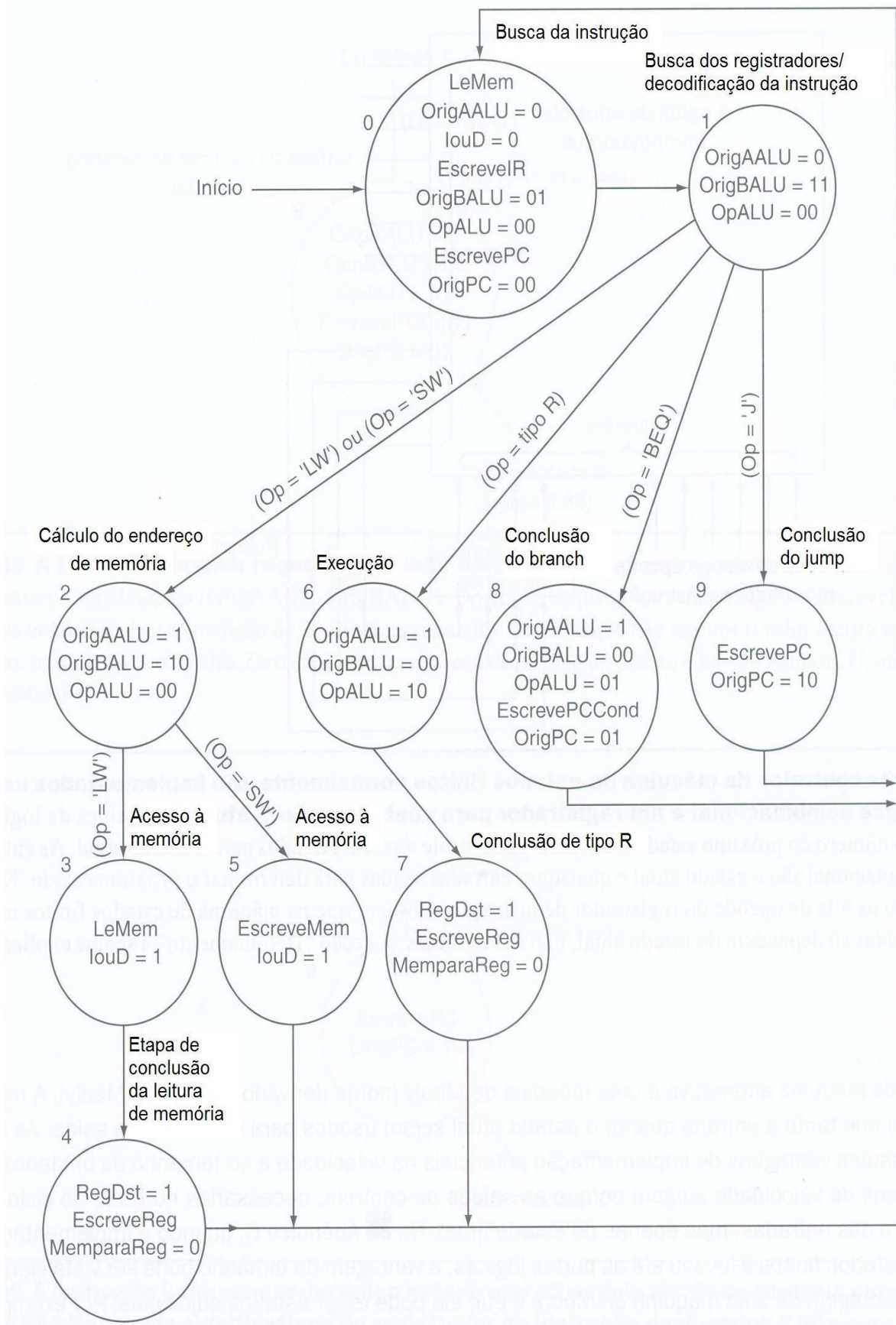


Figura 2.3.1.1. Máquina de estados finitos. Fonte: (Patterson; Hennessy, 2005).

### **2.3.2. Controle Utilizando Microprogramação**

Para representar graficamente um subconjunto das instruções do MIPS como mostrado na Figura 2.3.1.1, retirada do livro Organização e Projeto de Computadores por Patterson e Hennessy em 2005, é perfeitamente simples. O MIPS, porém possui cerca de 100 instruções e por isso torna-se inviável a utilização de representação gráfica. Pode-se pensar em programação para implementar um controle para todas essas instruções, diminuindo o trabalho de representá-las graficamente, utilizando micro-programação .

A micro-programação é uma técnica para projetar unidades de controle complexas. Ela usa um mecanismo de hardware simples que pode, então ser programado para implementar um conjunto de instruções mais complexo (Patterson; Hennessy, 2005).

A idéia de micro-programação é representar simbolicamente os valores do controle de modo que o microprograma seja uma representação das micro-instruções, da mesma forma que a linguagem de montagem é uma representação das instruções de máquina.

## **2.4. SOFTWARES EDUCACIONAIS**

Software educativo é um software cujo principal propósito é o ensino ou o auto-aprendizado. O objetivo principal é que o aluno que faça uso do Software e o mesmo tenha prazer em lidar com ele podendo praticar de maneira clara e objetiva o conteúdo a qual o software se refere.



De acordo com Avelar *et al* (2008) “para os alunos que estudam arquitetura e organização de computadores, dado o elevado nível de integração nos computadores modernos, e bastante difícil obter uma boa compreensão do que esta acontecendo por dentro do hardware”.

O uso do computador como ferramenta educacional tem se mostrado útil e proveitoso no processo de ensino-aprendizagem. Contudo, é importante frisar que o software educativo não deve ser tomado como algo que independe da orientação de professores e/ou tutores. Dentro de um contexto educacional propício e inovador, deve ser utilizado como complemento da didática já utilizada pelo professor.

A utilização do computador em Educação só faz sentido na medida em que os professores o conceberem como uma ferramenta de auxílio as suas atividades didático-pedagógicas, como instrumento de planejamento e realização de projetos interdisciplinares, como elemento que motiva e ao mesmo tempo desafia o surgimento de novas práticas pedagógicas, tornando o processo ensino-aprendizagem uma atividade inovadora, dinâmica, participativa e interativa (Teixeira; Brandão, 2003).

No intuito de discernir novas tecnologias e trabalhos aplicados na área de softwares educacionais existem comunidades e eventos com esse propósito sendo alguns deles presentes no congresso da Sociedade Brasileira de Computação (SBC 2010).

Dentre os eventos existentes tem-se o WEI – Workshop sobre Educação em Computação (2010), que tem por objetivo debater diferentes temas relacionados ao ensino de computação e informática visando discutir o currículo, metodologias e políticas, entre outros, de cursos de nível superior na área de Computação e Informática.

O WEAC – Workshop sobre Educação em Arquitetura de Computadores (2010), que também é um evento que aborda essas novas ferramentas para educação, e tem como objetivo principal a arquitetura de computadores. Seu foco

são os aspectos relacionados à educação neste objeto. A principal meta deste workshop é prover um ambiente (fórum) para professores, pesquisadores e estudantes trocarem, compartilharem e discutirem suas experiências, novas ideias e resultados de pesquisas em todos os aspectos e tópicos relacionados com a educação (aprendizado e ensino) em Arquitetura de Computadores.

Todos esses eventos e comunidades priorizam a educação em computação e suas vertentes áreas, como a arquitetura de computadores, e vem mostrando o quão importante é a utilização de softwares educacionais inclusos na didática aplicada pelos professores, trazendo benefícios para os alunos.

Um importante software educacional é o SPIM (LARUS, 1990), projeto começado em 1990 e encontra-se em desenvolvimento até os dias atuais. O SPIM é um simulador de uma arquitetura baseada nos processadores MIPS R2000/R3000 desenvolvido pelo Departamento de Ciências de Computação da Universidade de Wisconsin.

Além desse simulador, outro simulador muito importante é o WebSimple – MIPS onde SOUSA, Bruno F. De; et al. (2008) afirmam que “o uso do simulador causa um impacto positivo no aprendizado da disciplina Arquitetura de Computadores, auxiliando o aluno nas correções de suas práticas de aprendizado”, com isso observa-se que a utilização de complementos no estudo de Arquitetura de Computadores ajuda os alunos no aprendizado do conteúdo, assim o objetivo do simulador MIPS – PMS foi baseado neste conceito, onde a aceitação dos usuários na utilização do mesmo para alcançar um ganho no aprendizado foi alcançado.

### 3. METODOLOGIA

O projeto tem como finalidade desenvolver um software simulador para apoio no estudo da disciplina de Arquitetura de Computadores. Esse software educacional pode ser empregado pelo professor, nas aulas de Arquitetura de Computadores, como complemento do material didático já utilizado, a fim de tentar trazer mais absorção do conteúdo e facilidades no ensino dessa disciplina.

O software pode ser utilizado juntamente com a explicação teórica da arquitetura MIPS multiciclo, onde o professor pode demonstrar como ocorre a execução desta arquitetura no simulador, mostrando as mudanças nos valores dos registradores, tanto temporários quanto do banco de registradores, além mudanças nos valores das posições de memória e as imagens do circuito, mostrando quais elementos estão sendo usados.

#### 3.1. PARA O DESENVOLVIMENTO

Para o desenvolvimento do projeto utilizou-se de algumas ferramentas, como por exemplo, o *NetBeans*, que como consta em NETBEANSORG (2010) é um ambiente de desenvolvimento integrado e multiplataforma, de código aberto e que contém uma plataforma de aplicativos que permite aos desenvolvedores criar rapidamente aplicativos diversos utilizando diversas linguagens.

Com a utilização do *NetBeans* foi possível criar a interface de maneira fácil, onde pode-se preocupar ainda mais com a execução do simulador, já que a interface foi toda criada com as ferramentas que o *NetBeans* disponibiliza para ser utilizada.

Foi utilizada também a plataforma *Java*, que é uma linguagem de programação orientada a objeto. A linguagem *Java* “possui versatilidade, eficiência, portabilidade de plataforma e segurança, o que fazem dela a tecnologia ideal para a computação em rede” JAVA (2010).

A utilização do *Java* tem como objetivo atingir ao máximo de usuários possíveis utilizando a portabilidade que a mesma possui, ou seja, como essa linguagem possui portabilidade para ser utilizada em diversos sistemas operacionais, assim tanto alunos que utilizam Microsoft Windows, Mac OS e derivados do Linux, poderão utilizar o simulador para complementar seus estudos.

### **3.2. INSTALAÇÃO**

Para utilizar o simulador o usuário deverá instalar ou ter instalado a Máquina Virtual Java, que é a plataforma que o software foi desenvolvido. Logo após instalar a Máquina Virtual, deve-se verificar se o equipamento possui os requisitos abaixo.

Os requisitos do sistema são:

- Pentium 166 MHz ou superior.
- 64 MB de RAM ou superior.
- 98 MB de espaço disponível no disco rígido ou mais.

Para finalizar a instalação basta baixar o arquivo executável do simulador MIPS – PMS executá-lo com um duplo click, fazendo com que assim sua interface seja iniciada e o usuário possa utilizá-lo.

### 3.3. FUNCIONALIDADES E INTERFACE

A interface do sistema foi criada para ser intuitiva, limpa e dinâmica. Côtém apenas os campos e botões necessários para executar os códigos *assembler* de maneira eficaz e simples. Essa simplicidade contida na interface foi feita pensando em não dispersar a atenção dos alunos na hora da utilização do simulador, sendo assim a simplicidade deixa com que os alunos fixem sua atenção na simulação focando nas mudanças causadas pela execução das instruções no simulador. Tudo isso ajuda ainda mais aos alunos em entender o funcionamento da arquitetura.

O layout do sistema conta com 3 (três) botões de controle execução, que podem ser utilizados para carregar, dar início, ou parar a execução do código. Esses botões foram criados pensando em deixar o controle da execução da simulação da execução das instruções nas mãos dos usuários. Ao se carregar o código, o simulador faz uma análise sintática do mesmo, essa funcionalidade foi desenvolvida com intuito de mostrar aos alunos como é a sintática das instruções, para que os mesmos entendam também o que cada parte da instrução representa para a execução das mesmas. O software ainda possui um campo no qual se pode regular a velocidade (em segundos) dos ciclos de *clock* para que os alunos ajustem a velocidade da execução da simulação, tudo isso para que os mesmos possam adequar à simulação de formar a acompanhar a execução da melhor forma possível



Figura 3.3.1. Botões que compõe o layout do simulador MIPS - PMS

O layout representado na Figura 3.3.1 mostra os botões de controle de execução do simulador MIPS – PMS, demonstrando o botão carregar, *stop*, *play* e o controle de velocidade do ciclo de *clock*. Ambos os botões são utilizados para

controlar o fluxo de execução, possibilitando ao usuário um controle total da execução do simulador para que os alunos obtenham um melhor entendimento do que está acontecendo durante a execução.

O simulador possui também um painel onde ficam as informações sobre todos os 32 (trinta e dois) registradores que compõe a arquitetura. Este painel é atualizado a cada ciclo de *clock*, mostrando ao usuário como em cada passo de determinadas instruções mudam os valores destes registradores. O intuito da criação deste painel foi deixar claro para o usuário do simulador o conteúdo presente no banco de registradores, bem como as modificações realizadas pelas instruções nos valores presentes nesse banco de registradores.

Existe outro painel que mostra os registradores de estado, onde ficam as informações que serão utilizadas de um ciclo para outro subsequente, este painel foi criado com a intenção de mostrar para o usuário quais valores necessitam ser salvos nesses registradores temporários para que no próximo ciclo de *clock* não seja perdido no meio da simulação.

Outro painel importante é o que mostra os valores contidos na memória. Essa memória é dividida para guardar tanto instruções quanto dados, este painel atualiza a cada ciclo de *clock* seus valores. A criação deste outro painel tem como objetivo mostrar para o usuário do simulador como as instruções ficam salvas na memória e como essa mesma se divide para serem salvo tanto instruções quanto dados. Além disso, o painel foi criado com intuito de mostrar a dinâmica de acesso à memória, para que o usuário consiga ver e compreender os locais onde os dados podem ser salvos e onde fica a parte das instruções que fica bloqueada para o usuário.

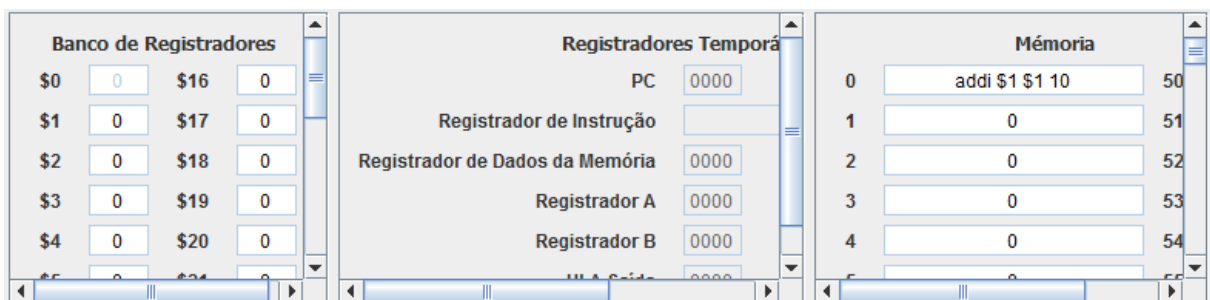


Figura 3.3.2. Painéis que contém os valores dos registradores temporários, banco de registradores e memória presentes no layout do simulador MIPS – PMS.

Para que o usuário consiga visualizar os valores presentes nos registradores do banco de registradores, na memória e nos registradores temporários foi criado no simulador MIPS – PMS os três painéis demonstrados na Figura 3.3.2 Estes painéis têm como maior utilidade mostrar ao usuário como os valores são atualizados a cada passo da execução das instruções, tentando passar para o usuário a lógica da simulação e como as instruções realizam as modificações nos valores destes painéis, isso para facilitar a compreensão da utilização de cada componente da arquitetura MIPS Multiciclo.

Um painel contendo apenas a imagem representativa do circuito da arquitetura MIPS Multiciclo também foi adicionado ao simulador como demonstrado na Figura 3.4.1 Este painel atualiza essa imagem de acordo com a instrução que está sendo executada e de acordo com cada passo de sua execução, identificando quais componentes do circuito esta sendo executado e os valores de controle que estão sendo utilizados pela instrução em execução. Este painel foi desenvolvido pensando na interatividade das imagens com a execução da instrução, onde o usuário possa identificar e entender quais componentes do circuito lógico está sendo utilizado em cada passa da execução da instrução. Este painel em conjunto com os demais presentes na Figura 3.3.2 dão uma abstração melhor para o usuário, afinal o usuário além de identificar quais componentes a instrução está utilizando, o mesmo ainda pode visualizar como as alterações neste componentes ocorrem.

O simulador ainda conta com um campo texto demonstrado também na Figura 3.4.1, onde é possível digitar um código *assembler*, respeitando as instruções que o programa reconhece e suas sintáticas, para que o software possa executá-las. Duas opções para tratar desse código também estão presentes no simulador, a opção de salvar o código digitado, onde o usuário pode salvar o código em um arquivo para eventuais futuras execuções, e a opção de carregar um código já existente, que carrega o arquivo salvo para o campo texto do simulador e permite ainda ao usuário alterar o código carregado.

O campo texto e suas ferramentas de abrir e salvar um código *assembler* tem como objetivo trazer interatividade do software com o usuário, além de facilitar a simulação dos códigos *assembler* dos mesmos, já que o usuário não necessita digitar o mesmo código toda a vez que precisar simulá-lo.

### 3.4. UTILIZANDO O SIMULADOR

Para simular a execução do processador MIPS Multiciclo no simulador, pode-se utilizar a funcionalidade de abrir um arquivo contendo um código *assembler*, digitar um código no campo de texto ou mesmo abrir um arquivo e editá-lo para se adequar a execução que se pretende e utilizá-lo na execução.

O simulador ainda conta com um campo onde o usuário poderá alterar a velocidade do ciclo de *clock*, colocando o intervalo, em segundo, de um ciclo para outro. Tudo isso para facilitar na visualização da execução, colocando o usuário no controle da execução da simulação.

Além disso o simulador ainda conta com botões de controle para facilitar ainda mais a integração do usuário com o sistema. O software possui um botão de *stop* (parar) que pode ser acionado em qualquer parte da execução de um código que irá para a execução e zerar todos os valores nos registradores e memória do simulador. O simulador possui também o botão *play* utilizado para dar início a execução do código. Por fim o botão carregar, que quando pressionado faz a análise sintática do código e o libera para ser executado. Todos estes botões foram criados pensando na interatividade do software com o usuário, colocando o controle da simulação nas mãos do usuário.

A Figura 3.4.1 mostra o layout do simulador MIPS – PMS que contém 3 painéis, sendo um de registradores que mostra todos os valores presente nos 32 registradores, um painel de registradores temporários, e um que contém todas as posições de memórias, mostrando os valores na parte de dados da memória e mostrando também a parte onde ficam as instruções.



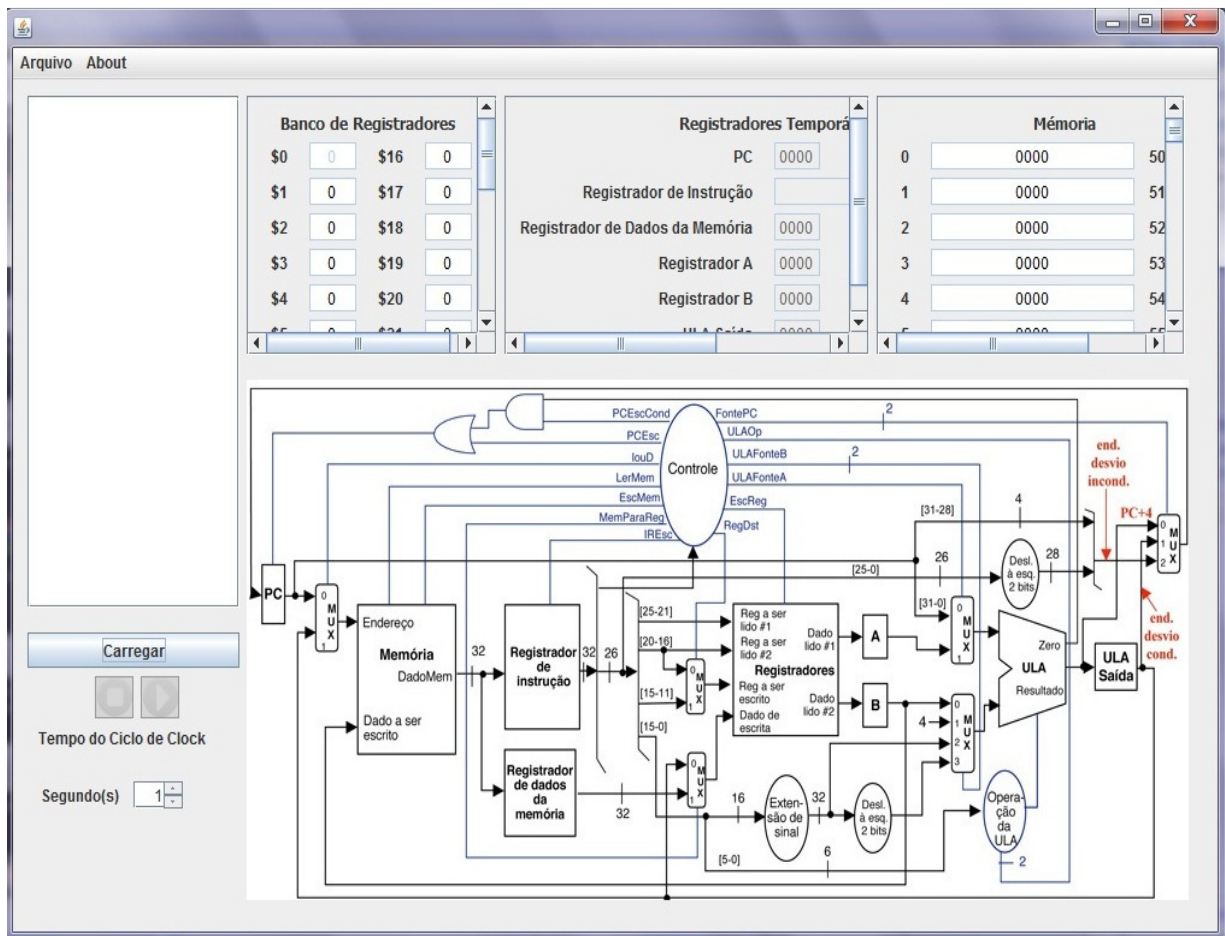


Figura 3.4.1. Layout do simulador MIPS – PMS.

O layout mostrado na Figura 3.4.1 também contém uma imagem que se atualiza a cada passo da execução das instruções mostrando o que está sendo usado em cada passo para exemplificar o que acontece com o caminho de dados. Além disso o layout conta com um campo de texto para inserir o código, onde o mesmo pode ser carregado ou salvo em um arquivo. Os botões de play, stop, carregar e o contador de segundos que determina a velocidade de cada clock também estão presentes no layout.

Quanto a sintática do código *assembler* a utilização se restringe apenas em utilizar espaços entre cada item da instrução, a sintática utilizada nas instruções foi baseada com a do livro de Patterson e Hennessy em 2005, isso para que o aluno possa utilizar os conceitos teóricos da sintática das instruções também no simulador, facilitando assim a utilização do mesmo.

Na seqüência do texto está um exemplo da especificação da sintaxe correta de cada instrução que o simulador reconhece.

- `addi $1 $1 10`
- `add $3 $1 $2`
- `subi $4 $3 3`
- `sub $5 $3 $1`
- `and $6 $2 $5`
- `or $7 $1 $2`
- `xor $8 $5 $4`
- `sw $3 $0 ( 204 )`
- `lw $10 $0 ( 204 )`

O simulador ainda conta com uma opção para salvar o código inserido no campo texto para novas execuções e um analisador sintático que é utilizado para evitar o erro na digitação das instruções afim de que os usuários aprendam não só como funciona um processador MIPS Multiciclo, mas também como é a sintática das instruções e como elas se dividem.

### **3.5. PARA A AVALIAÇÃO**

Foi aplicado um questionário para alunos do curso de Ciência da Computação. O intuito deste questionário é conhecer opiniões sobre aspectos gerais do sistema, tais como, facilidade de uso, funcionalidades, interface, compreensão do conteúdo e satisfação em usá-lo. O questionário foi aplicado em aula da disciplina de Arquitetura de Computadores, logo após os alunos terem utilizado o simulador..

O questionário, que pode ser encontrado em (ANEXO 1 – Questionário), foi criado com a ferramenta de criação de formulário do GoogleDocs (2010), que é um pacote de aplicativos do Google baseado em *AJAX* e totalmente online. Utilizando

as respostas dos usuários foram criados gráficos onde se pode visualizar de forma mais fácil e intuitiva as respostas dos alunos nos questionários.

## 4. RESULTADOS

Para analisar o ganho que este trabalho representa para a sociedade acadêmica foi realizado testes com alunos do Curso de Ciência onde os resultados mostram o que há de bom no simulador e o que precisa ser adicionado para que o simulador se torne ainda mais intuitivo e didático.

Tabela 4.1. Dados dos testes feitos com o simulador MIPS – PMS.

	1.Qual o grau de facilidade em utilizar o simulador MIPS – PMS?	2. Qual o grau de facilidade em acompanhar a execução das instruções no simulador?	3. O simulador trouxe facilidade na compreensão do conteúdo?	4.A interface é intuitiva?	5.A interface é agradável?	6. A interface é simples e contém o conteúdo principal para visualizar a execução das instruções?
1 (Péssimo)	0	0	0	0	0	0
2 (Ruim)	0	0	0	1	0	0
3 (Regular)	1	2	2	2	2	0
4 (Bom)	7	7	8	4	8	6
5 (Ótimo)	7	6	5	8	5	9
	8. A funcionalidade de abrir um arquivo contendo o código assembler facilita a utilização do simulador?	9. A funcionalidade de salvar um arquivo contendo o código assembler facilita a utilização do simulador?	10. Qual o grau de satisfação em utilizar o simulador?	11. Qual seu grau de interesse em Arquitetura de Computadores?	12. Qual seu grau de interesse em estudar a Arquitetura dos processadores MIPS?	
1 (Péssimo)	0	0	0	1	2	
2 (Ruim)	0	0	0	0	0	
3 (Regular)	3	2	1	4	2	
4 (Bom)	6	7	8	6	10	
5 (Ótimo)	6	6	6	4	1	

Os números mostrados na Tabela 4.1 expõem a avaliação dos alunos sobre o simulador, onde os mesmos avaliaram as funcionalidades, interface, facilidade na compreensão da execução, na utilização e interesse na matéria de Arquitetura de Computadores.

Analisando os resultados pode-se observar que no geral os alunos aprovaram o simulador e suas funcionalidades, mas faz-se uma ressalva que o mesmo ainda precisa de algumas modificações para ser melhor avaliado.

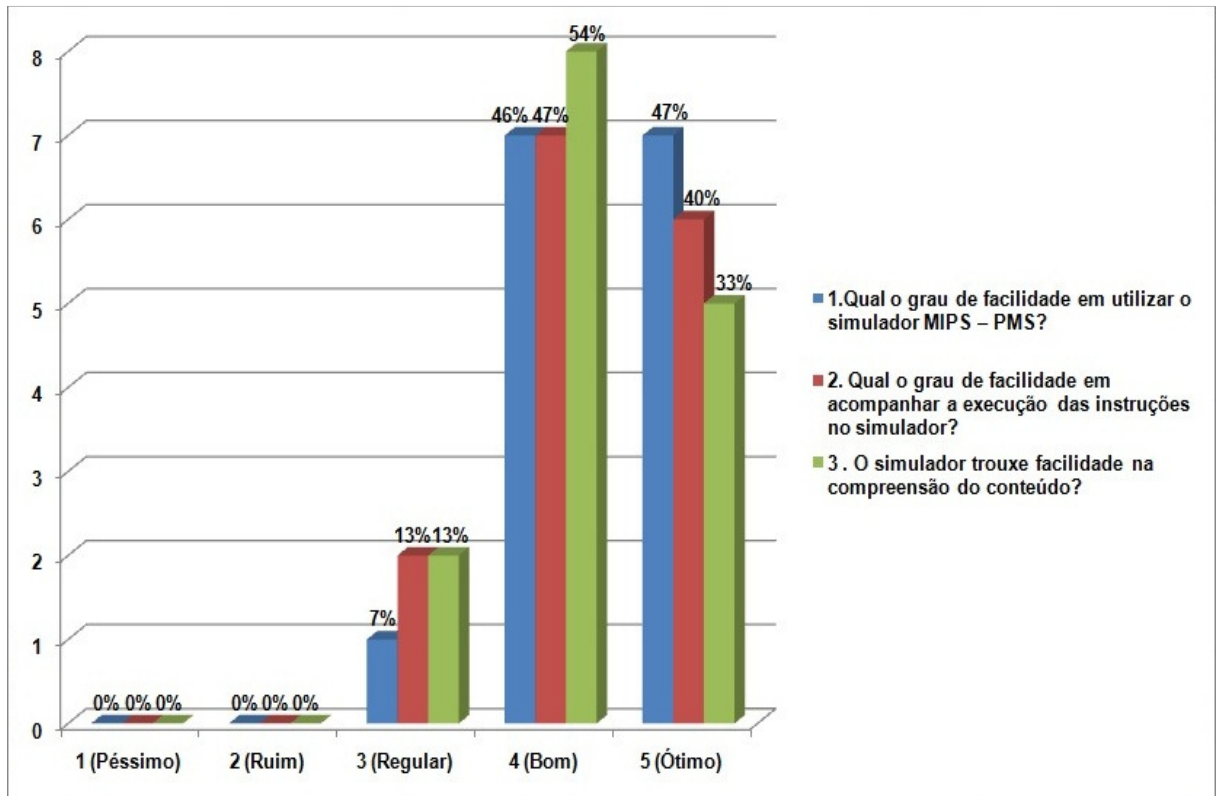


Figura 4.1. Gráfico dos dados referentes às questões um, dois e três.

Com base na Tabela 4.1 foram criados gráficos para demonstrar o resultado dos testes de forma mais intuitiva.

Pode-se observar na Figura 4.1 que o desempenho do simulador MIPS – PMS, em relação à facilidade de uso foi muito bem avaliada, onde mais de 90% dos entrevistados afirmam que o grau de facilidade em utilizar o simulador é bom ou ótimo. Pode-se então concluir dessa análise que as funcionalidades criadas para que o simulador fosse fácil de ser manipulado foram aprovadas pelos entrevistados, onde se vê a necessidade de criar algumas outras funcionalidades que facilitem ainda mais a utilização do software.

Quanto a facilidade em acompanhar a execução e facilidade na compreensão do conteúdo, representada na Figura 4.1, foi muito satisfatória. Na avaliação, 40% dos entrevistados responderam ótimo e 47% responderam bom, demonstrando que os alunos acharam que aprovaram as funcionalidades colocadas no software para que os mesmo pudessem acompanhar com facilidade a execução da simulação das instruções. A principal funcionalidade criada para trazer essa facilidade de acompanhar a execução foi o painel contendo uma imagem que é atualizada a cada passo da execução de um instrução, mostrando quais componentes da arquitetura

do processador MIPS Multiciclo esta sendo usada. Além disso, os painéis contendo os valores dos registradores temporários, o banco de registradores e a memória foram criados para mostrar o detalhamento destes componentes e como seus valores são atualizados. Toda essa estrutura foi criada para ajudar o aluno e trazer facilidade para acompanhar a execução da simulação.

De acordo com a avaliação dos alunos presente exposto na Figura 4.1, 54% dos alunos consideram que o simulador foi bom em trazer facilidade na compreensão do conteúdo, 33% afirmam que o simulador foi ótimo nesse mesmo quesito, e 13% afirmam que o simulador foi regular em trazer facilidade na compreensão do conteúdo. Analisando estes dados pode-se concluir que o simulador ajuda na compreensão do conteúdo ministrado nas aulas, porém não foi uma avaliação totalmente boa, os 13% dos entrevistados que afirmaram que o simulador foi regular nesta questão, vêem a necessidade de melhorias no software, dentre as quais pode se considerar a visualização dos sinais de controle, onde apenas alguns podem ser visualizados.

Esse resultado demonstra que o simulador satisfaz o objetivo proposto, servindo como complemento de estudos em Arquitetura de Computadores, trazendo para o aluno facilidade na compreensão do conteúdo, além de ser um software fácil utilização, todas essas facilidades ajudam ao aluno a compreender o conteúdo com mais facilidade, existem quesitos que podem ser modificados para obter uma avaliação ainda melhor, como por exemplo, os sinais de controle que podem ser mostrados em um painel disposto junto com os demais.

Conforme se observa na Figura 4.2, que trata sobre a interface do simulador MIPS – PMS desde o grau de intuição até simplicidade da interface, o simulador possui um ótimo desempenho onde 80% dos entrevistados afirmam que o simulador possui uma boa ou ótima intuição do conteúdo presente na interface. Esta intuição esta presente devido a grande quantidade de informação que esta presente na execução, que tem por objetivo ajudar os alunos na compreensão do conteúdo. Essa questão teve também uma avaliação onde 20% dos entrevistados afirmam que o simulador é regular ou ruim questão da intuição do simulador. Com essa avaliação pode-se notar que se deve acrescentar mais informações para que a interface fique ainda mais intuitiva.

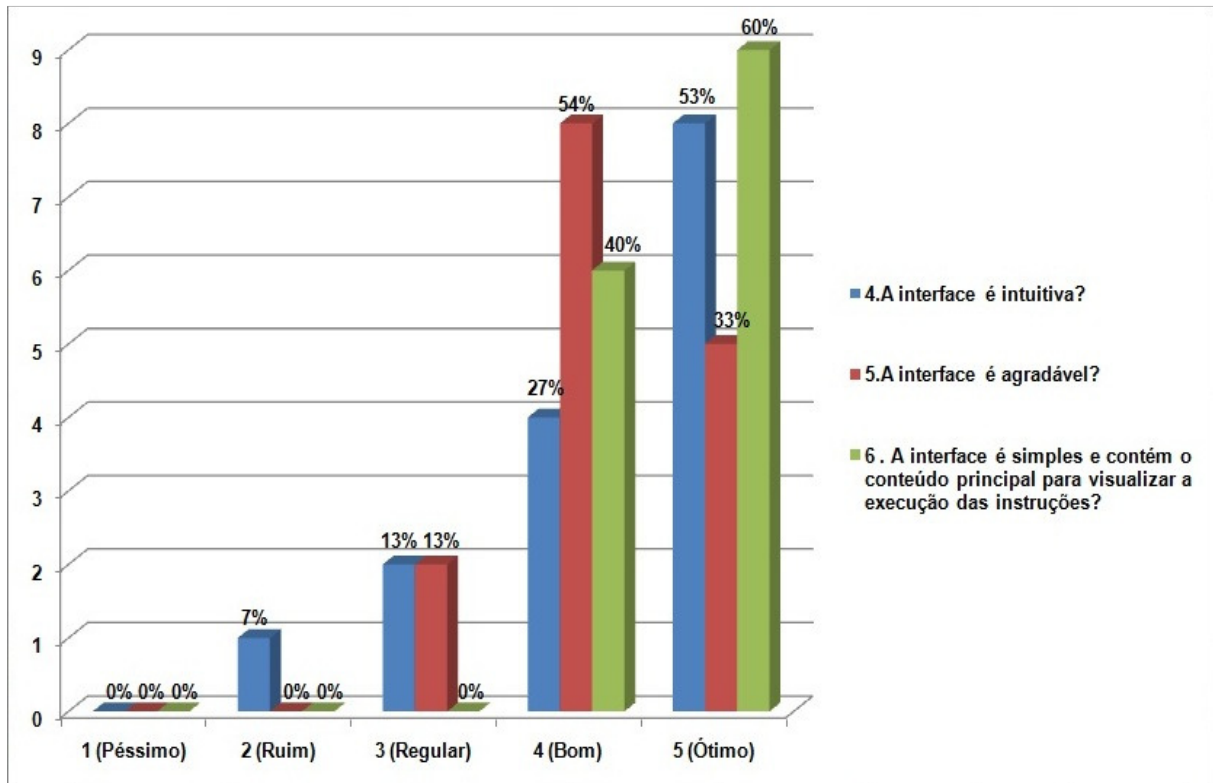


Figura 4.2. Gráfico dos dados referente às questões quatro, cinco e seis.

Na Figura 4.2 expõe também que 87% dos entrevistados afirmam que a interface é agradável respondendo essa questão como ótimo ou bom. Isso mostra que a disposição dos componentes e das informações que estes componentes apresentam agradou a maioria dos entrevistados. Apenas 13% dos alunos afirmaram que o simulador é regular quanto a possuir uma interface agradável. Para que esta avaliação possa possuir 100% de aproveitamento bom ou ótimo, deve-se então mudar a fonte das letras e números de forma que agrade aos alunos, além de fazer uma pesquisa sobre os ícones que para mudança.

Quanto a questão que avalia a simplicidade da interface o simulador foi muito bem aceito, onde 60% dos entrevistados afirmaram ser ótimo e os outros 40% acham que a simplicidade da interface é boa, demonstrando que o simulador atendeu ao objetivo esperado, ser simples. Essa simplicidade tem como objetivo atrair a atenção do aluno para execução, e não distraí-lo com outras informações sem necessidade. Em questão de simplicidade o simulador não precisa de modificações já que alunos afirmam que a interface é ótima ou boa, o único cuidado a ser tomado em trabalhos futuros é manter esse nível de simplicidade na hora de colocar alguma outra funcionalidade.

Com toda essa sondagem sobre a interface do simulador, nota-se que os componentes ficaram muito bem distribuídos de forma simples e intuitiva, além disso, nota-se também que boa parte dos alunos achou a interface agradável, sendo assim o simulador é agradável de utilizar e simples de ser manipulado. Algumas alterações podem ser feitas nos trabalhos futuros para tornar a interface ainda mais intuitiva, como por exemplo, adicionar algumas informações sobre o controle do processador MIPS Multiciclo.

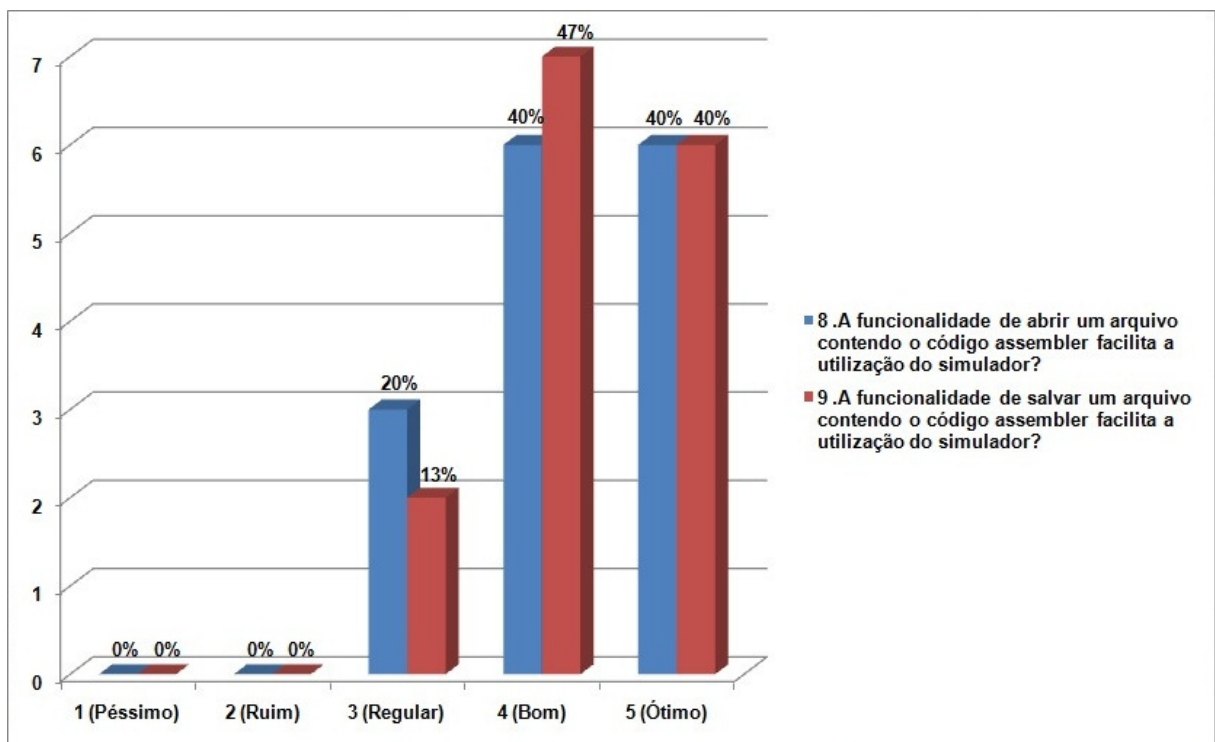


Figura 4.3. Gráfico referente às questões oito e nove.

De acordo com os dados apresentados na Figura 4.3 pode-se observar que as funcionalidades, de abrir um código existente foi avaliada em 80% com sendo bom ou ótimo, mostrando que essa funcionalidade teve uma boa avaliação, facilitando a utilização do simulador evitando que o usuário necessite redigitar o código para executá-lo. Deve-se observar também que 20% dos entrevistados avaliarem essa funcionalidade como regular, com isso percebe-se que deve ser feitas alterações nessa funcionalidade para facilitar ainda mais sua utilização.

A funcionalidade de salvar um código inserido no simulador foi muito bem aceita pelos entrevistados já que os mesmos avaliaram essa funcionalidade em 87% como sendo ótima ou boa, demonstrando que é de grande valia salvar o código que



está sendo utilizado na simulação para possíveis futuras utilizações. Com relação aos outros 13% dos entrevistados que afirmaram que esta funcionalidade é regular, percebe-se que pode aprimorar essa funcionalidade para facilitar ainda mais a reutilização do código *assembler* digitado no simulador.

Com esses dados nota-se que tais funcionalidades são importantes em um simulador, pois evitam o retrabalho em ter que redigitar códigos para realizar novas simulações. Ainda pode-se concluir que essas funcionalidades podem ser alteradas para proporcionar para o usuário ainda mais facilidade na utilização do simulador.

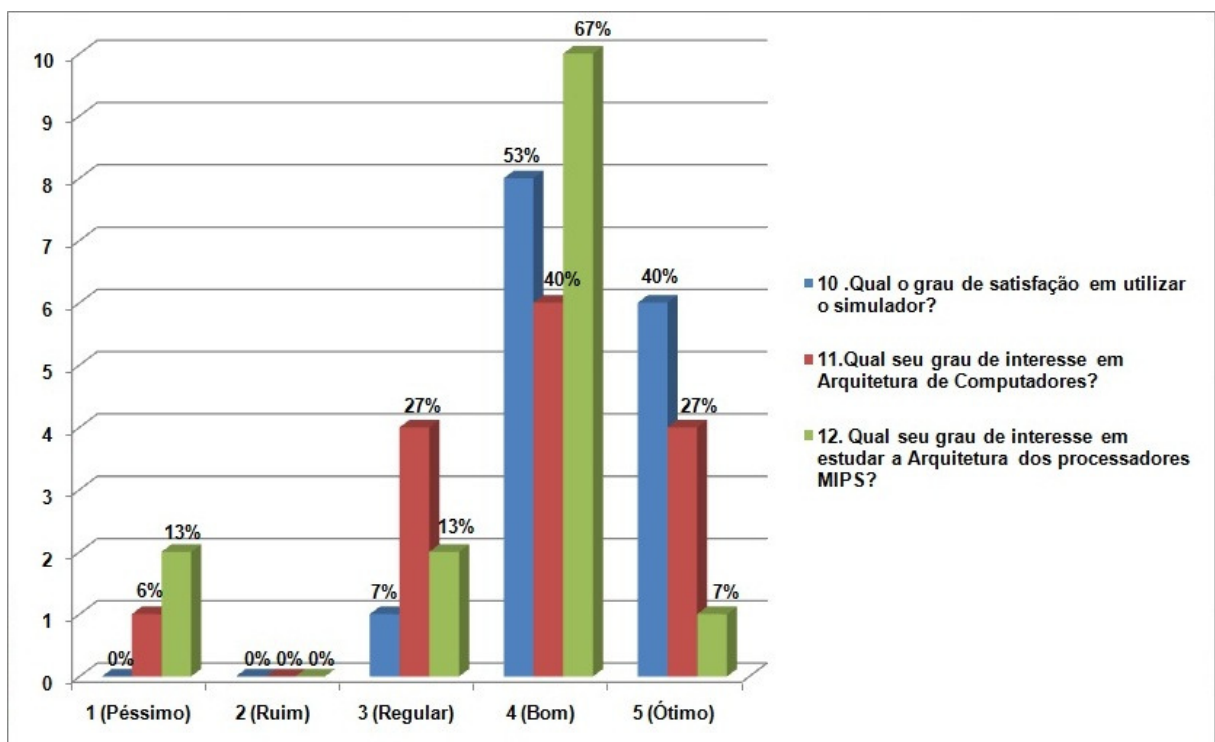


Figura 4.4. Gráfico referente às questões dez, onze e doze.

Os dados apresentados pelo gráfico da Figura 4.4 expõem que 93% dos alunos afirmam que a satisfação em utilizar o simulador é ótima ou boa, confirmando que o simulador foi muito bem aprovado para ser utilizado como complemento no estudo da disciplina de Arquitetura de Computadores. Os outros 7% dos entrevistados afirmaram que a satisfação em utilizar o simulador é regular, demonstrando que o simulador foi muito bem desenvolvido, mas podem ser feitas algumas alterações para alcançar a satisfação de 100% dos alunos.

A figura 4.4 mostra também o interesse dos alunos na disciplina de Arquitetura de Computadores, onde 67% afirmam possuir um bom ou ótimo

interesse na disciplina, o que é um número razoável. Os demais entrevistados que somam 33% afirmam que o interesse na disciplina é regular ou ruim, onde pode-se concluir que muitos alunos estão desinteressados na disciplina. O simulador foi desenvolvido pensando nesses alunos desinteressados também, fazendo o estudo de MIPS Multiciclo mais iterativo e fácil, tudo isso para tentar atrair o interesse desses alunos e ajudar a todos na compreensão do conteúdo do MIPS Multiciclo.

Analisando a questão de número 12, ainda a Figura 4.4, pode-se observar que 74% dos alunos afirmam ter um bom ou ótimo interesse em estudar a Arquitetura dos processadores MIPS, o que é um ótimo número. Para esses alunos o simulador irá servir como complemento de estudo, para que os mesmos possam entender conceitos básicos e avançados do funcionamento de um processador MIPS e seus componentes. Já os 26% dos entrevistados que afirmaram que possuem um interesse péssimo ou regular em estudar a Arquitetura dos processadores MIPS, o simulador tem como objetivo facilitar o estudo e atrair a atenção deste alunos para estudo dessa arquitetura.

Avaliando todas as questões presentes no gráfico da Figura 4.4 pode-se observar que a maioria dos alunos obtiveram satisfação em utilizar o software, em contraste com isso percebe-se que mesmo alguns alunos, que não tem interesse na disciplina de Arquitetura de Computadores e também não possuem interesse em estudar a arquitetura dos processadores MIPS, afirmam que estão satisfeito em utilizar o simulador. Essa comparação demonstra que o simulador chamou a atenção de vários alunos, inclusive alunos que não estão interessados na disciplina de Arquitetura de Computadores, o que é um ótimo resultado, onde comprova-se a eficácia de sua utilização de acordo com o objetivo de incentivar o estudo na disciplina.

Tabela 4.2. Tabela contendo os dados referentes à questão sete.

	7 .Você utilizaria esse simulador como complemento para o estudo em arquitetura de computadores?
Sim	15
Não	0

De acordo com a Tabela 4.2 todos entrevistados, que fizeram os testes com o software simulador, utilizariam o simulador MIPS – PMS para complementar o estudo em Arquitetura de Computadores. Este resultado mostra que o software simulador foi muito bem aceito pelos alunos que o testaram, tanto que os mesmo afirmam que utilizariam como o complemento de estudos.

Comparando a Tabelas 4.2 e a Tabela 4.1 percebe-se que o software simulador foi muito bem aceito pelos alunos, ressalvo algumas mudanças na interface que devem ser realizadas em trabalhos futuros, como por exemplo, a adição de painéis que mostrem o funcionamento de outros componentes do processador MIPS Multiciclo. O simulador atingiu todos os objetivos propostos, que foi levar para o aluno um software simples, agradável, que possua todo o conteúdo, que possua funcionalidades que ajudem na utilização do mesmo e que possua fácil utilização, onde os alunos em sua grande maioria aprovaram as funcionalidades do software.

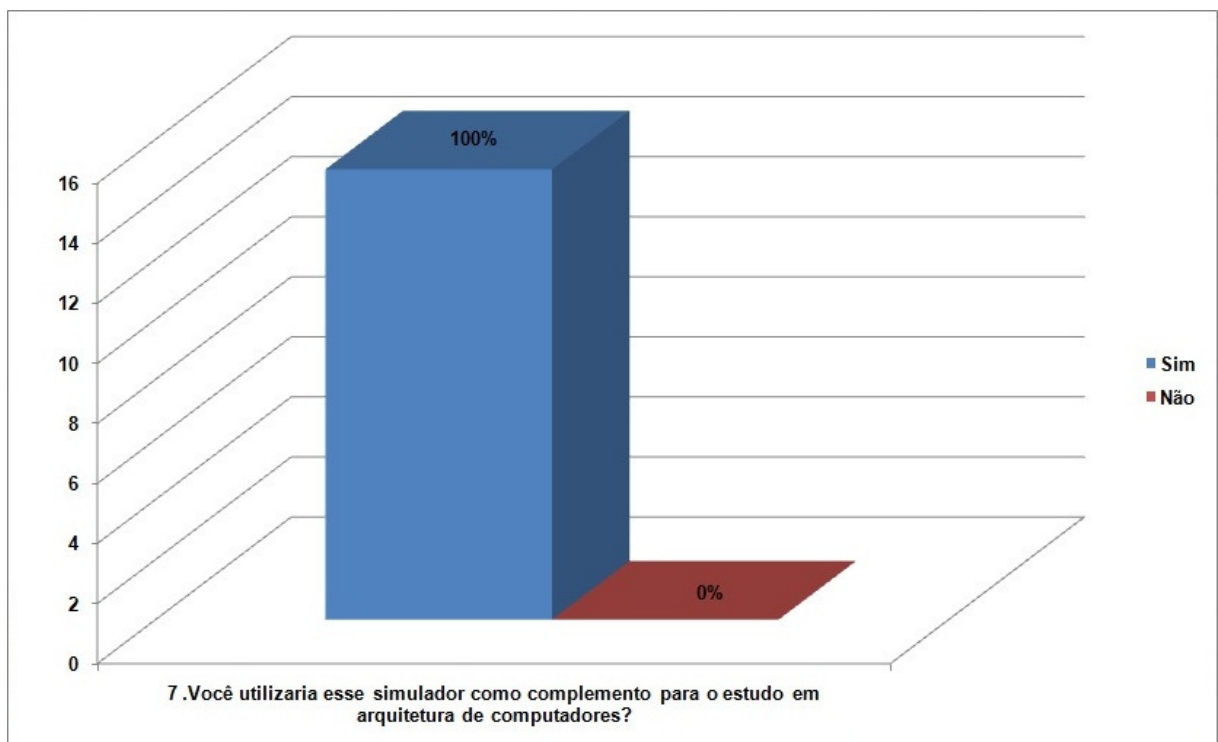


Figura 4.5. Gráfico referente a questão sete.

Na Figura 4.5 onde apresenta o gráfico referente a questão de número sete, onde nota-se que 100% dos alunos entrevistados afirmaram que utilizariam o simulador MIPS – PMS para o complementar o estudo em Arquitetura de

Computadores. Comparando os números presentes no gráfico presente na Figura 4.5 e os números presentes no gráfico presente na Figura 4.4, percebe-se que mesmos os alunos que afirmam não terem interesse na disciplina de Arquiteturas de Computadores e no estudo da arquitetura do processador MIPS, afirmam que utilizariam o simulador, isso mostra que o simulador atende ao objetivo proposto que é ser utilizado pelos alunos como apoio no estudo da arquitetura do processador MIPS Multiciclo, e ainda incentivando alunos que não tem interesse no estudo dessa disciplina.

Visualizando a pesquisa no geral, pode-se chegar a conclusão que em média as respostas foram sempre positivas em relação aos itens perguntados. O ponto que deve ser modificado em trabalhos futuros é em questão da adição de mais painéis contendo o conteúdo de outros componentes da arquitetura dos processadores MIPS Multiciclo, isso para que o programa se torne ainda mais intuitivo.

O simulador MIPS - PMS tem sido muito proveitoso para a comunidade acadêmica, salvo alguns pontos que ainda precisam evoluir para chegar a um ponto satisfatório.

## 5. CONCLUSÃO

A interatividade e praticidade sempre são itens muito desejados por todos os estudantes, por proporcionar aulas mais dinâmicas, intuitivas e atrativas. Em todos os cursos existem disciplinas que não atraem a atenção de todos os alunos, o que é o caso da disciplina de Arquitetura de Computadores. Pode-se analisar de acordo com a Figura 4.4 que mais de 30% dos alunos avaliam que possuem pouco ou nenhum interesse em estudar a disciplina e ainda de acordo com a Figura 4.4, 26% dos alunos afirmam que possuem pouco ou nenhum interesse em estudar a arquitetura dos processadores MIPS. Contudo a Figura 4.5 exemplifica que todos os alunos utilizariam o simulador como complemento de estudo.

Pode-se ser analisado então que mesmo os alunos que possuem pouco ou nenhum interesse na disciplina de Arquitetura de Computadores e no estudo da arquitetura do processador MIPS, utilizaria o simulador pelo mesmo ser simples e agradável como mostra o gráfico presente na Figura 4.2, onde os alunos afirmam na sua grande maioria que o simulador possui uma interface realmente simples, o que torna sua utilização muito agradável.

A criação do simulador MIPS – PMS foi uma iniciativa de promover uma aula mais prática, onde os alunos aprendem o conteúdo trabalhando com os comandos e realizando testes em tempo real, analisando como conteúdo aprendido com o professor é utilizado na prática. Isso tudo facilita a compreensão do conteúdo da disciplina, já que o aluno aprende a teoria e coloca isso na prática de forma simples e interativa.

O presente trabalho mostra como o simulador MIPS – PMS foi desenvolvido, como e o porquê as funcionalidades foram criadas e estruturadas. Foram mostrados também os requisitos para a instalação, como a instalação pode ser feita e como se utiliza o simulador.

Para avaliar o simulador foi aplicado um questionário, contendo 12 questões de avaliação e uma questão para comentar sobre o simulador, para alunos do curso de Ciência da Computação, nos quais 15 alunos responderam. Nesta avaliação o

simulador teve uma boa aceitação onde 100% dos alunos definiram que utilizariam o simulador como complemento no estudo da disciplina de Arquitetura de Computadores.

Concluindo, o simulador foi muito bem aceito pelos alunos entrevistados, alcançando o objetivo de ser utilizado como apoio no estudo da disciplina de Arquitetura de Computadores, além de incentivar alunos que não possuem interesse na disciplina de Arquitetura de Computadores e tão pouco em estudar a arquitetura do processador MIPS.

Verificando a aceitação do simulador MIPS-PMS, pode-se concluir também que a criação de novos simuladores se faz necessária, pois estimula a interação do aluno com a disciplina para que o mesmo foi desenvolvido, auxiliando o aluno na aprendizagem da mesma.

## 6. TRABALHOS FUTUROS

Com base nos testes realizados com o simulador MIPS – PMS alguns trabalhos futuros podem ser desenvolvidos para melhorar a interface e o desempenho do software simulador, dentre eles têm-se:

- Acrescentar as funcionalidades undo/redo utilizando padrões de projeto, para uma maior interatividade com o usuário, colocando um controle ainda maior nas mãos destes.
- Adicionar painéis mostrando os valores de controle da arquitetura do processador MIPS Multiciclo.
- Deixar mais visíveis as funcionalidades de abrir/fechar código *assembler*.
- Adicionar mais instruções ao software simulador para que o mesmo possa simular outras instruções, como por exemplo, instruções de desvio condicional e incondicional (*beq, j* e etc.).
- Fazer uma análise de usabilidade utilizando avaliação com heurísticas de IHC.
- Realizar testes com um número maior de alunos.
- Realizar testes com professores da disciplina de Arquitetura de Computadores para obter uma avaliação de funcionalidades e praticidade de uso.

## REFERÊNCIAS

AVELAR, Cíntia P.; et al. **MPDSim: Simulador didático do Pipeline do MIPS de 32 bits**. In: Workshop sobre Educação em Arquitetura de Computadores (WEAC), 2008, Campo Grande.

BARCELLOS, César da Cunha; SIMÕES, Henrique Magalhães. **Ultimate Pipeline Simulator: Simulador do Pipeline do MIPS**. In: Workshop sobre Educação em Arquitetura de Computadores (WEAC), 2009, São Paulo.

CONSENDEY, Michael Martins; BRASIL, Rafael Henrique; MARTINS, Carlos A. Paiva da Silva. **SIPPIS MIPS: Um Novo Simulador de Pipeline do MIPS 32 bits para Auxílio à Educação em Arquitetura de Computadores**. In: Workshop sobre Educação em Arquitetura de Computadores (WEAC), 2009, São Paulo.

FELIX, Ariane F.; POUSA, Christiane V.; CARVALHO, Milene B. **DIMIPSS: Um simulador didático e interativo do MIPS**. In: Workshop sobre Educação em Arquitetura de Computadores (WEAC), 2006, Ouro Preto.

GoogleDocs – **Google docs: Crie e compartilhe seu trabalho on-line com o Google Docs**. Disponível em: <<http://docs.google.com>>. Acessado em: 06 out. 2010.

Güntzel, José Luís. **Arquitetura e Organização de Computadores II, Aula 6**. Universidade Federal de Pelotas. Disponível em: <[http://www.inf.ufsc.br/~guntzel/ine641400/AOC2\\_aula6.pdf](http://www.inf.ufsc.br/~guntzel/ine641400/AOC2_aula6.pdf)>. Acessado em: 15 jun. 2010.

JAVA – **Saiba mais sobre a tecnologia Java**. Disponível em: <[http://www.java.com/pt\\_BR/about/](http://www.java.com/pt_BR/about/)>. Acessado em: 05 de out. 2010.

LARUS, James R..**SPIM S20: A MIPS R2000 Simulator**. Disponível em: <<http://www.cs.wisc.edu/~larus/spim.html>>. Acessado em: 15 jun. 2010.



LUCENA, M. **A Gente é uma Pesquisa: Desenvolvimento Cooperativo da Escrita Apoiado pelo Computador**; Dissertação de Mestrado; Departamento de Educação, PUC - Rio; Rio de Janeiro: 1992.

MAIA, Davidson W. N.; VIEIRA, Marcelo M.; PESSOA, Ramon F. **PS – CAS MIPS: Um Simulador De Pipeline Do Processador MIPS 32 Bits Para Estudo de Arquitetura de Computadores**. In: Workshop sobre Educação em Arquitetura de Computadores (WEAC), 2009, São Paulo.

MIPS – **MIPS Technologies**. Disponível em: <<http://www.mips.com>>. Acessado em: 06 de out. 2010.

NETBEANSORG - **NetBeans Org**. Disponível em: <[http://netbeans.org/community/releases/69/index\\_pt\\_BR.html](http://netbeans.org/community/releases/69/index_pt_BR.html)>. Acessado em: 05 de out. 2010.

PATTERSON, D. A.; HENNESSY J. L. **Organização e Projeto de Computadores – A Interface Hardware/Software**. 2ª Edição, Editora Livros Técnicos e Científicos, 2000.

PATTERSON, D. A.; HENNESSY J. L. **Organização e Projeto de Computadores – A Interface Hardware/Software**. 3ª Edição, Editora Campus, 2005.

SBC – **Sociedade Brasileira de Computação**. Disponível em: <<http://www.sbc.org.br/>>. Acessado em: 20 abr. 2010.

SOUSA, Bruno F. De; et al. **WebSimple – MIPS: Simulador Web-based do Pipeline do MIPS**. In: IX Simpósio em Sistemas Computacionais WSCAD – SSC, 2008, Campo Grande.

TEIXEIRA, Adriano Canabarro; BRANDÃO, Edemilson Jorge Ramos. **Software Educacional: O Difícil Começo**. In: Revista Novas Tecnologias na Educação (RENOTE), 2003, UFRGS – Universidade Federal do Rio Grande do Sul.

WEAC – **Workshop Sobre Educação em Arquitetura de Computadores**. Disponível em: <<http://www.ppgee.pucminas.br/weac/2009/>>. Acessado em: 20 abr. 2010.

**WEI – Workshop Sobre Educação em Computação.** Disponível em: <[http://csbc2009.inf.ufrgs.br/index.php?option=com\\_content&task=view&id=31&Itemid=76](http://csbc2009.inf.ufrgs.br/index.php?option=com_content&task=view&id=31&Itemid=76)>. Acessado em: 20 abr. 2010.

## ANEXO 1 – QUESTIONÁRIO

# Avaliação do Simulador MIPS - PMS

Arquitetura de Computadores

Nome:

Curso / Período:

1. Qual o grau de facilidade em utilizar o simulador MIPS – PMS?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

2. Qual o grau de facilidade em acompanhar a execução das instruções no simulador?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

3. O simulador trouxe facilidade na compreensão do conteúdo?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

4. A interface é intuitiva?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

5.A interface é agradável?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

6 . A interface é simples e contém o conteúdo principal para visualizar a execução das instruções?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

7.Você utilizaria esse simulador como complemento para o estudo em arquitetura de computadores?

- Sim
- Não

8 .A funcionalidade de abrir um arquivo contendo o código assembler facilita a utilização do simulador?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

9 .A funcionalidade de salvar um arquivo contendo o código assembler facilita a utilização do simulador?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

10. Qual o grau de satisfação em utilizar o simulador?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

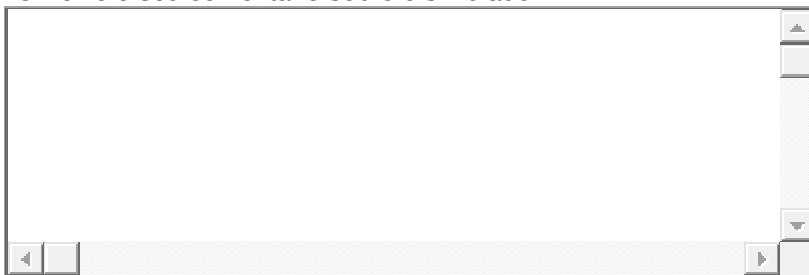
11. Qual seu grau de interesse em Arquitetura de Computadores?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

12. Qual seu grau de interesse em estudar a Arquitetura dos processadores MIPS?

- 1 (Péssimo)
- 2 (Ruim)
- 3 (Regular)
- 4 (Bom)
- 5 (Ótimo)

13. Deixe o seu comentário sobre o simulador.

A large empty text input field with a scroll bar on the right and navigation arrows at the bottom.