

Projeto – Mini Simulador Mips

Objetivo:

O objetivo deste projeto é desenvolver um simulador de processador MIPS que permita aos usuários carregar código assembly MIPS, executar passo a passo e visualizar os efeitos da execução de cada instrução nos registradores, memória e outros elementos relevantes do processador. Exercitar e fixar os conhecimentos adquiridos sobre linguagem assembly e arquitetura de computadores. Reforçar o conhecimento sobre decisões de projeto e funcionamento de hardware. Exercitar conceitos sobre integração software e hardware por meio de simulação.

Requisitos Funcionais:

1. Interface Gráfica de Usuário (GUI):
 - a. Desenvolver uma GUI amigável que permita aos usuários carregar, editar e executar código assembly MIPS.
 - b. Fornecer opções para carregar, salvar e editar programas assembly.
 - c. Exibir uma janela que represente o estado atual dos registradores e memória.
 - d. Permitir que os usuários executem o programa passo a passo (passo único) e em velocidade ajustável.
2. Simulador de Processador MIPS:
 - a. Implementar um núcleo de simulação que interprete as instruções MIPS e atualize o estado de cada registrador.
 - b. Simular os registradores (32 registradores GPRs), a memória de dados e a memória de instruções.
 - c. Implementar as instruções do conjunto de instruções MIPS descritas na seção **Subconjunto de Instruções**.
3. Interpretação de Código Assembly:
 - a. Decodificar e executar o código assembly
 - b. Desenvolver um analisador que traduza código assembly em Código hexadecimal
 - c. Lidar com diretivas como .data e .text.
4. Visualização de Saída:
 - a. Atualizar a GUI com informações em tempo real sobre o estado do processador (registradores, memória, PC, etc.) a cada etapa da execução.
 - b. Fornecer informações de depuração, como valores dos registradores, valores da memória e instruções em execução.

Requisitos Técnicos:

1. Linguagem de Programação: Use uma linguagem de programação como C,

- C++, Java ou Python para implementar o simulador e a GUI.
2. Bibliotecas Gráficas: Utilize bibliotecas gráficas (por exemplo, PyQt, Tkinter, JavaFX) para criar a interface do usuário.
 3. Estruturas de Dados: Utilize estruturas de dados eficientes para representar o estado do processador (registradores, memória).
 4. Testes: Realize testes abrangentes para garantir a precisão da simulação.

Cronograma:

Divida o projeto em marcos com datas de conclusão para acompanhar o progresso. Isso ajudará a manter o projeto organizado e dentro do prazo.

Documentação:

Documente todo o projeto, incluindo a arquitetura, algoritmos, estruturas de dados, API e guias de uso.

Subconjunto de Instruções:

1. Lógica e aritméticas

ADD ADDI SUB SUBI MULT DIV AND OR XOR NOR SLT SLL SRA
LI

2. Chamadas de Sistema Sistema

IMPRIMIR INTEIRO
LER INTEIRO
IMPRIMIR STRING
SAIR (EXIT)

3. Acesso a memória

exemplo: LW SW LB SB

4. Condicionais e Saltos condicionais

exemplo: BEQ BNE BGT BEZ

5. Saltos incondicionais

exemplo J JR JAL

O trabalho poderá ser realizado em dupla, onde cada dupla deverá utilizar em sua apresentação pelo menos 3 programas de testes para mostrar o funcionamento do simulador. Os códigos dos programas devem ser armazenados em arquivos **.ass**.

Conteúdo a ser entregue

Código fonte compactado.
Documentação do Projeto (formato .pdf)

Prazo:

O professor irá combinar com a turma a apresentação do cronograma (Valendo 1 ponto)

A partir do cronograma o professor marcará datas específicas para o acompanhamento deste (2 pontos)

O prazo para a entrega e apresentação final é: **06/12/2023**