

**Nome: Tobias Viero de Oliveira**

**Data: 03/11/2023**

## **Relatório sobre o Trabalho do Decoder de Instruções de Máquina MIPS em Assembly MIPS\*\***

### **Introdução:**

Este relatório descreve o processo de desenvolvimento de um programa em Assembly MIPS da disciplina de Organização de Computadores ministrada pelo professor Giovani Baratto para decodificar instruções de máquina binárias em linguagem Assembly MIPS. O objetivo principal do projeto foi criar um decodificador capaz de ler instruções em formato binário, interpretá-las e imprimir a representação em linguagem Assembly MIPS correspondente. O desenvolvimento envolveu várias etapas desafiadoras, aplicação de conceitos de Assembly MIPS, manipulação de bits, estruturas de controle, leitura e escrita de arquivos, vários dias, e muita calma.

### **Objetivos:**

O objetivo principal do trabalho foi criar um decodificador de instruções de máquina binárias para a linguagem Assembly MIPS. Mais especificamente, os objetivos incluíram:

1. Ler instruções de máquina em formato binário de um arquivo e imprimir o Program Counter(PC) atual, representação da instrução em hexadecimal e representação em linguagem Assembly Mips em um arquivo de texto.
2. Decodificar as instruções binárias, identificando os campos opcode, rs, rt, rd, imm, target e funct.
3. Implementar a lógica de tradução para a linguagem Assembly MIPS, considerando diferentes tipos de instruções.
4. Garantir a precisão na manipulação dos campos de instrução, incluindo a extensão de sinal de números quando necessário.
5. Identificar e lidar com instruções desconhecidas ou não mapeadas.

### **Revisão Bibliográfica:**

O desenvolvimento desse projeto exigiu uma boa compreensão da arquitetura MIPS, seus registradores, instruções e formatos. Além disso, foram necessários conhecimentos de programação em Assembly MIPS, manipulação de bits e uso de estruturas de controle e manipulação de arquivos. Para isso foi necessário consultar sites de documentação, slides do professor e a parte de ajuda do software MARS.

## **Metodologia:**

A metodologia adotada para atingir os objetivos incluiu as seguintes etapas:

1. Leitura das instruções de máquina em formato binário de um arquivo.
2. Impressão no arquivo o PC atual.
3. Conversão dos valores binários em representações hexadecimais e escrita em um arquivo de texto separado.
4. Decodificação das instruções, identificando os campos opcode, rs, rt, rd, imm, target e funct usando operações de shift left e shift right.
5. Comparação do opcode com valores conhecidos para determinar o tipo de instrução.
6. Implementação da lógica específica para cada tipo de instrução, incluindo o tratamento de imediatos, registradores e extensão de sinal quando necessário.
7. Impressão da representação em linguagem Assembly MIPS no arquivo de texto.

## **Experimento:**

O projeto foi desenvolvido em três arquivos principais: o arquivo "main" que controla o fluxo do programa, o arquivo "hexa" que realiza a conversão binário-hexadecimal e o arquivo "decoder" que contém a lógica de decodificação. No início comecei fazendo todas as operações em somente um arquivo e fazia pouco uso de funções, à medida que fui avançando percebi que a utilização de diversos arquivos e funções facilitaria e muito o trabalho.

## **Resultados:**

Os resultados do experimento foram promissores. O programa foi capaz de decodificar com sucesso várias instruções de máquina binárias em linguagem Assembly MIPS. As representações em linguagem Assembly MIPS foram precisas e correspondentes às instruções de entrada. Instruções desconhecidas foram identificadas e tratadas adequadamente. O programa foi executado com diferentes instruções de máquina binárias para testar a eficácia da decodificação.

## **Discussão:**

O desenvolvimento deste decodificador envolveu desafios significativos, como a manipulação de bits, extensão de sinal e a necessidade de lidar com diferentes tipos de instruções. O uso de funções separadas para lidar com registradores e conversão de binário para hexadecimal facilitou a organização do código. No entanto, a manutenção do código exigiu atenção detalhada devido ao grande número de registradores e instruções diferentes, quando mudava alguma função e esquecia de mudar um registrador por exemplo, passava muito tempo tentando achar o erro.

## **Conclusões e Perspectivas:**

Tive bastante dificuldade inicial e somente posteriormente com mais calma, dividindo o problema em procedimentos e arquivos consegui me orientar de forma mais tranquila, no final percebi que já estava minimizando os erros e fazendo o trabalho mais rápido.

Entretanto, o projeto atingiu com sucesso seus objetivos de criar um decodificador de instruções de máquina binárias para a linguagem Assembly MIPS. O programa é capaz de decodificar uma variedade de instruções e lidar com instruções desconhecidas de forma apropriada.