

Decodificador Pocket MIPS - Parte 1

**Carlos Alberto de Jesus Pereira Neto RGA 2019.1907.046-0 ,
Ingracy Moraes RGA 2019.1907.039-7,
Heliton Oliveira da Silva RGA 2016.1907.055-3**

FACOM – Universidade Federal de Mato grosso do Sul - UFMS

carloshtcursos@gmail.com

Abstract. *The theme addressed in this work is about the MIPS Computing Instruction Set (Microprocessor Without Interlocked Pipeline Stages). We will make a brief description of MIPS and present a code example, in C language, containing instructions of type R, ADDI and BEQ.*

Resumo. *O tema abordado neste trabalho é sobre o Conjunto de Instruções Computacionais MIPS (Microprocessor Without Interlocked Pipeline Stages). Faremos uma breve descrição do MIPS e apresentarmos um exemplo de código, na linguagem C, contendo instruções do tipo R, ADDI e BEQ.*

1. Introdução

O MIPS é uma arquitetura de Conjunto de Instruções (Instruction Set Architecture - ISA), desenvolvida pela empresa MIPS Computer System, atualmente conhecida por MIPS Technologies.

Para uma maior compreensão do tema, descrevemos alguns conceitos (bases):

Conjunto de instruções: É um conjunto de sistemas com objetivos de executar ações específicas de responsabilidades individuais. Todos os subsistemas formam um dos elementos de todo Conjunto de instruções.

Arquitetura do Conjunto de Instruções: A arquitetura define os tipos de instruções, o formato das instruções, o modo de endereçamento e tipo de programas que ele será capaz de executar;

Tamanho da Instrução: Definido em bits e pode ser fixo ou variável. A arquitetura MIPS atual tem o tamanho fixo de 64 bits e se chama MIPS 64 Bits.

Formato da Instrução: Os projetistas da arquitetura é quem define utilizando o campo da instrução específico para isto com o nome OPCODE. O OPCODE indica qual é a operação que deverá ser executada.

Compilação: Um Conjunto de Instruções ISA necessita de um compilador para traduzir para Assembly o que está sendo executado, e em seguida para linguagem binária para assim ser reconhecida e executada pelo processador.

2. Execução da Instrução

A tabela abaixo apresenta algumas instruções básicas dos registradores do MIPS.

Mnemonic	Format	Opcode Field	Function Field	Instruction
Add	R	0 (00000)	8 (1000)	Add
Addi	I	8 (01000)		Add Immediate
Sub	R	0 (00000)	10 (1010)	Subtract
And	R	0 (00000)	12 (1100)	Bitwise And
Or	R	0 (00000)	13 (1101)	Bitwise Or
Slt	R	0 (00000)	14 (1110)	Set If Less Than
Lw	I	19 (10011)		Load Word
Sw	I	27 (11011)		Store Word
Beq	I	4 (00100)		Branch on Equal
Bne	I	5 (00101)		Branch on Not Equal
J	J	2 (00010)		Jump

O MIPS possui registradores que são as memórias rápidas e de alto custo para um projeto de um microprocessador.

A execução de qualquer instrução no processador deve ser convertida para Assembly MIPS e em seguida para binário.

Considerando uma simples instrução $a = b + c$ em linguagem C pode ser utilizado os registradores temporários de nome t do MIPS [$\$t0..\$t7$]. Esta instrução ficará da seguinte forma:

- Linguagem de montagem:

ADD $\$t0$, $\$s0$, $\$s1$ - **ADD** é o mnemônico para ADIÇÃO, $\$s0$ é o valor que está armazenado em b , $\$s1$ o valor da variável c e $\$t0$ é a variável a .

- Linguagem de máquina:

A linguagem de máquina troca o nome do registrador pelo seu número correspondente na tabela MIPS. A instrução acima ficará assim: **ADD $\$8$, $\$16$, $\$17$**

3. Aplicação demonstrando algumas instruções (mips-pocket-decoder)

O Decodificador Pocket MIPS é um processador simples e bem específico, que conta com 4 registradores, denominados **$r0$, $r1$, $r2$ e $r3$** , para a execução das instruções. Essa versão possui um conjunto de instruções menor que um MIPS de 64 bits, mas cumpre o mesmo objetivo.

O programa irá ler as instruções informadas e, caso estejam válidas, irá gerar uma saída em código de máquina contendo a instrução fornecida.

O projeto conta com cabeçalhos (arquivos **.h**) que contém o nome das operações suportadas e seus respectivos códigos. Esses cabeçalhos possuem Registros (struct) agrupando variáveis para implementação da arquitetura.

Ao ler uma linha de instrução o programa irá procurar na coleção de instruções o respectivo comando e irá executar o método de decodificação implementado para aquele caso.

Operações aceitas para Instruções do tipo R:

- ADD, SUB, MTA, MTB, MFA, MFB e HALT; (*informar o nome da instrução seguida do código do registrador*)

Entrada	Saída
ADD $r1$	000 01 000

Operações aceitas para Instruções do tipo I;

- ADDI; (*informar o nome da instrução seguida de um número inteiro de 0 a 31*)

Entrada	Saída
ADDI 6	001 00110

Operações aceitas para Instruções do tipo J;

- ADDI; (*informar o nome da instrução e em seguida o código de dois registradores*)

Entrada	Saída
BEQ $r0$ $r1$	011 00 01 0
BEQ $r1$ $r2$	011 01 10 0

4. Conclusão

O propósito deste trabalho foi expor como um conjunto de instruções é executado pelos processadores, qual linguagem “de baixo nível” utilizada e qual a linguagem compreendida pelos hardwares.

Os estudos também deixaram claros que há algumas décadas houve uma evolução considerável dos processadores mas que atualmente esta evolução tem chegado ao seu limite. Apesar de muitos esforços dos pesquisadores os mesmos não estão tendo muito sucesso para construir processadores mais rápidos utilizando as tecnologias atuais.

O projeto desenvolvido pode ser encontrado no repositório [Mips-Pocket-Decoder](#) no GitHub.

Referências

<https://www.embarcados.com.br/serie/mips/>

<https://chortle.ccsu.edu/assemblytutorial/index.html#part1>