Este montador gera um arquivo .mif para o quartus para uso no projeto da disciplina de Infra de Hardware para Engenharia da Computação da Universidade Federal de Pernambuco.

Estes arquivos se referem a um projeto que foi criado e compilado usando a NetBeansIDE, que compila os arquivos Montador.java e Arquivo.java, lê as instruções em Assembly.txt e gera um arquivo binário de memória (instrucoes.mif) que é o que vai ser copiado para uso no projeto da disciplina.

Este projeto já implementa todo o conjunto de instruções que está na especificação do projeto no site da disciplina.

Anexo: Instruções para a configuração e carregamento da memória e uso do montador de instruções

Memória:

A memória, usada no projeto e disponibilizada nos componentes, usa a estratégia de armazenamento Big-endian, que é a estratégia usada pelos processadores Mips.

Exemplo de Big-Endian: Armazenar a palavra $90AB12CD_{16}$ (base hexadecimal).

Na implementação Big-Endian, o byte mais significativo é armazenado nos menores

Address	Value
1000	90
1001	AB
1002	12
1003	CD

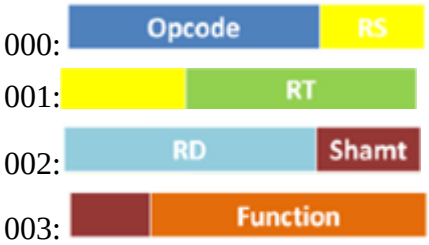
endereços da memória, $90AB12CD_{16}$ em **Big Endian** escrito na memória ficaria assim:

Address	Value
1000	CD
1001	12
1002	AB
1003	90

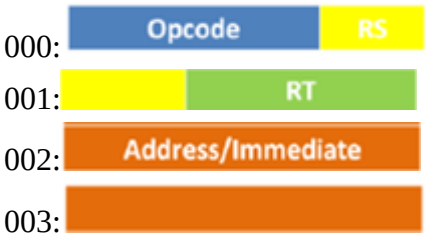
Enquanto na estratégia **Little-Endian** para a mesma palavra, seria:

Para o teste do projeto o arquivo .mif deverá ser configurado com um conjunto de instruções específico. Isso é feito com a inserção das instruções nas posições de memória que

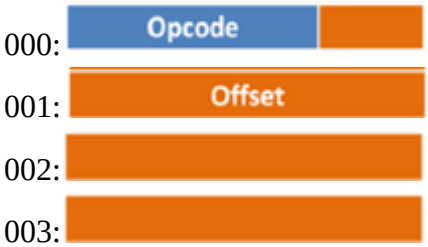
estão marcadas na margem esquerda de cada linha do arquivo .mif. As figuras abaixo mostram como dispor as instruções de cada tipo nas posições de memória.



Instruções do tipo R



Instruções do Tipo I



Instruções do Tipo J

Na posição do arquivo Memoria.vhd destacada abaixo insira o nome do arquivo .mif para ele passar a ser parte do seu projeto, onde por padrão está configurado com o nome instrucoes.mif.

```
package ram_constants is
constant DATA_WIDTH : INTEGER := 8;
constant ADDR_WIDTH : INTEGER := 8;
constant INIT_FILE : STRING := "instrucoes.mif";
end ram_constants;
```

Faça a síntese de seu projeto e os testes. Caso seja necessário mudar o arquivo .mif, uma nova Netlist(simulação) deverá ser gerada.

Criando um arquivo .mif usando o montador

É possível criar um arquivo.mif com instruções usando o montador cedido pela monitoria e que está no site da disciplina.

Regras do montador: No arquivo de nome Assembly.txt devem ser colocadas as instruções que se quer simular no projeto, onde a partir destas instruções será gerado um arquivo .mif.

A **primeira linha** deve conter a **quantidade de instruções** que você irá colocar, seguido dessas instruções que devem ser no máximo 57.

No arquivo .mif as posições a partir da 228 contém as rotinas de tratamento das exceções que não devem ser alteradas mas implementadas conforme a especificação.

Não deve ser usado $ para expressar o número de um registrado **nem** ( ) nas instruções de Load e Store.

Nas instruções de Desvio condicional, desvio incondicional, loads e stores, os endereços que não devem ser labels (Ex.; beq $3,$5, a) mas endereços físicos.

Nas instruções de desvios condicionais e incondicionais, deve ser levado em conta que o endereço ainda será multiplicado por 4, o que faz parte do cálculo de endereço.

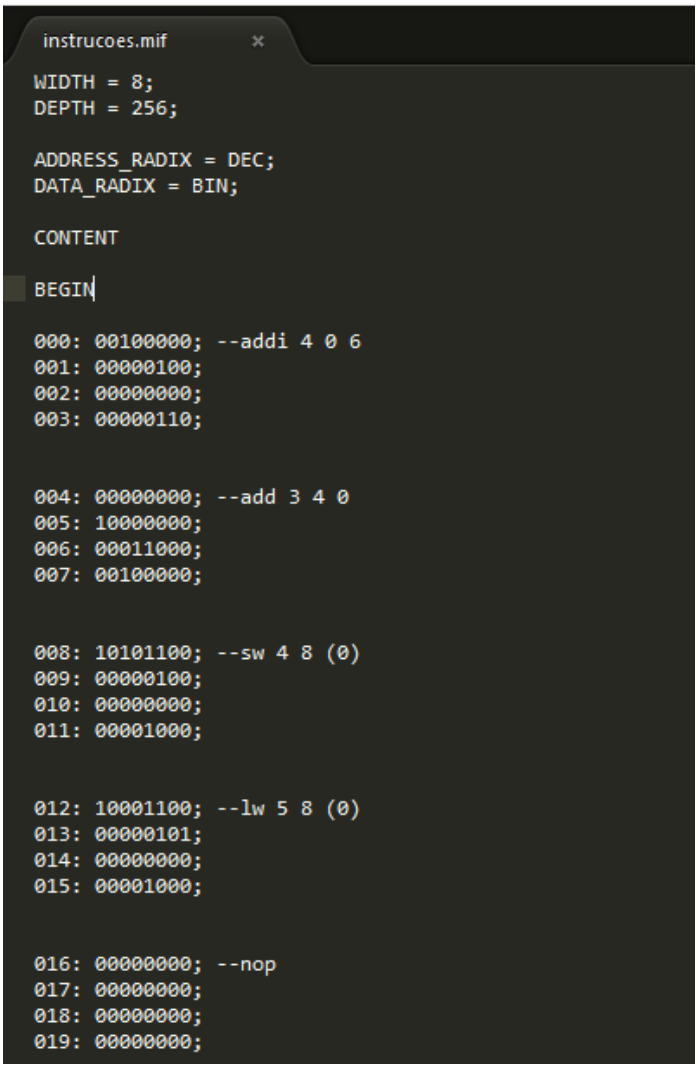
Nas instruções de Load/Store o cálculo do endereço é o mostrado na especificação onde o endereço final é o valor do **registrador base multiplicado por 4 + deslocamento.**

O MONTADOR FUNCIONA APENAS COM NÚMEROS POSITIVOS !

Exemplo de entrada válida:

4
addi 4 0 6
add 3 4 0
sw 4 8 0
lw 5 8 0

Saída:



O montador não tem um console com resultados dos erros nem acertos.
Se o arquivo Assembly.txt estiver correto e você executar o montador ele vai gerar um arquivo instrucoes.mif, caso haja algum erro o arquivo instrucoes.mif não será gerado e você deve consertar o erro e executar de novo o montador.

O arquivo instrucoes.mif deve ser copiado e colocado na pasta onde está a memoria.vhd para então o projeto ser compilado e simulado.

Nesse período a monitoria irá disponibilizar alguns arquivos de memória .mif com as instruções que estão neles detalhadas para que o projeto seja testado por vocês na confecção do mesmo.

LINK PARA DOWNLOAD:
http://www.cin.ufpe.br/~if674/arquivos/2017.1/Projeto/Montador%20v1.3.rar
https://github.com/leamorim/Montador-de-Instrucoes-MIPS

VERSIONAMENTO DO MONTADOR:

Versão 1.1: Alteração para instrução xor que não funcionava
Versão 1.2: Correção das instruções sllv e srav cujos valores de rs e rt estavam trocados
Versão 1.3: Correção de bugs nas instruções subu, addu e xor, opcode rte