

Instruções do processador pipeline:

Instrução	Formato assembly	Comentários
soma	soma rd, r1, r2	Soma dois números.
subtrai	subtrai rd, r1, r2	Subtrai o primeiro número pelo segundo.
pega	pega rd, imediato	Pega um imediato e guarda em um registrador.
vai	vai destino	Pula para um endereço específico.
multiplica	multiplica rd, r1, r2	Multiplica dois números.
maior	maior r1, r2, destino	Salta se um número é maior do que o outro.
igual	igual r1, r2, destino	Salta se os dois números são iguais.
menor	menor r1, r2, destino	Salta se um número é menor do que o outro.
ler	ler rd, ro	Lê uma posição da memória.
escrever	escrever ro, rd	Escreve em uma posição da memória.
fodaSe	fodaSe rd	Sorteia um número entre 0 e 127.
aMimir	aMimir	Não faz nada.
somaFoda	somaFoda rd, ro	Soma o número com o resultado da operação anterior.
multiplicaFoda	multiplicaFoda rd, ro	Multiplica o número com o resultado da operação anterior.
lerFoda	lerFoda rd	Lê a posição de memória calculada na operação anterior.
escreverFoda	escreverFoda ro	Escreve na posição de memória calculada na operação anterior.

Conceitos básicos do assembler

Para evitar a necessidade de programar diretamente em linguagem de máquina, desenvolvemos um programa em Python para converter códigos de assembly em linguagem de máquina. Isso permite uma programação mais acessível e eficiente, utilizando assembly.

Inicialmente, é necessário inserir os endereços do arquivo de entrada e saída na memória do seu computador. Se o programa não localizar o arquivo de entrada, será encerrado.

O nosso processador contém 16 registradores de 32 bits, identificados pelo símbolo £ (libra esterlina), seguido de um número de 00 a 15.

O assembler é capaz de detectar erros de sintaxe. Após a detecção de um erro, a escrita no arquivo de saída é interrompida, e o programa é encerrado, exibindo uma mensagem indicando a linha em que o erro ocorreu. Além disso, o assembler permite comentários, iniciados por //, que são ignorados durante a leitura e tradução para a linguagem de máquina.

As instruções estão apresentadas na tabela acima, enfatizando que o uso da vírgula não é obrigatório. Ela serve apenas para melhorar a legibilidade do código. No entanto, é necessário o uso de espaços, mesmo se a vírgula for utilizada. O formato básico da instrução consiste no seu nome e nos respectivos argumentos, podendo ser registradores, imediatos ou endereços de salto.

Por fim, é importante destacar que o assembler possui a capacidade de detectar erros de sintaxe. Quando um erro é identificado, o processo de escrita no arquivo é interrompido imediatamente. O assembler então exibe uma mensagem indicando não apenas em qual linha o erro ocorreu, mas também o conteúdo específico dessa linha.