

Implementação da instrução Store Triple Word na arquitetura MIPS

João Wallace Lucena Lins

Bacharelado em Engenharia de Computação, Centro de Informática – Universidade Federal da Paraíba
jwallace.lucena@gmail.com

INTRODUÇÃO

Antes de tudo, faz-se necessário a apresentação das siglas que serão usadas durante esse relatório.

RISC Reduced Instruction Set Computer
MIPS Microprocessor without Interlocked Pipeline Stages
ALU Arithmetic Logic Unit
STW Store Triple Word

O MIPS é uma arquitetura de processadores que segue o padrão RISC e que não apresenta pipelines, de modo que seu caminho de dados é simples e didático, sendo uma ótima introdução para estudantes ao mundo das arquiteturas. Com isso em mente, e tomando como base os conhecimentos adquiridos nas aulas da disciplina Arquitetura de Computadores, faremos a implementação da instrução STW em um processador hipotético com arquitetura MIPS.

METODOLOGIA

Para alcançar nosso objetivo, primeiramente realizamos um estudo sobre o caminho de dados de nosso processador, e também de como se dava o funcionamento das instruções que referenciavam a memória, no caso do MIPS, as instruções LW e SW. A partir disso, encontramos uma solução de simples compreensão que faria uma pequena alteração no caminho de dados, tal alteração é exposta na figura abaixo.

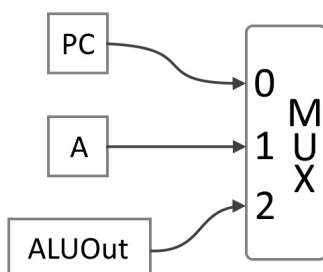


Figura 1: Alteração no MUX da primeira entrada da ALU.

Com isso, conseguimos usar o ALUOut como uma das entradas da ALU, tornando possível que façamos a soma de quatro bits – por meio da segunda entrada da ALU – a um endereço de memória que se encontrar no ALUOut, ou seja, conseguimos

calcular os endereços seguintes a um computado anteriormente.

Assim, nos resta inserir novos estados que executem a instrução STW, visível na figura abaixo, em nossa máquina de estados.

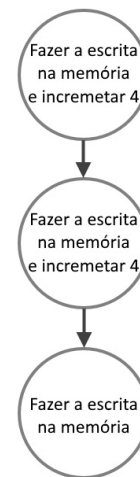


Figura 2: Estados da STW.

Vale ressaltar que esses estados devem ser inseridos após o "Memory Address Computation", uma vez que tal estado já aplicará o offset no endereço de memória a ser gravado. E com essa modificação, também será necessário alterar a tabela de despacho 2, que é justamente a tabela de despacho de acesso à memória. Ao final disso, teremos uma nova máquina de estados representada pela figura 3.

Por fim, temos de fazer a implementação propriamente dita, que terá o label STW2, por estar na segunda tabela de despacho. Antes de prosseguirmos, vale lembrar que o registrador ALUOut guarda a saída da ALU do ciclo anterior, portanto devemos computar o endereço a ser utilizado no próximo ciclo durante o ciclo atual. O resultado é visível na tabela 1.

CONCLUSÃO

Enfim, foi possível alcançar nosso objetivo inicial sem realizar modificações significativas no caminho de dados. Dessa forma, conseguimos manter o mínimo de alterações em outras instruções, que agora precisariam considerar o MUX da primeira entrada da ALU com dois bits de controle, ao invés de um, como foi exposto na figura 3.

Label	ALU Control	SRC1	SCR2	Register Control	Memory	PCWrite Control	Sequencing
STW2	Add	ALUOut	4		Write ALU		Seq
	Add	ALUOut	4		Write ALU		Seq
					Write ALU		Fetch

Tabela 1: Implementação da instrução STW.

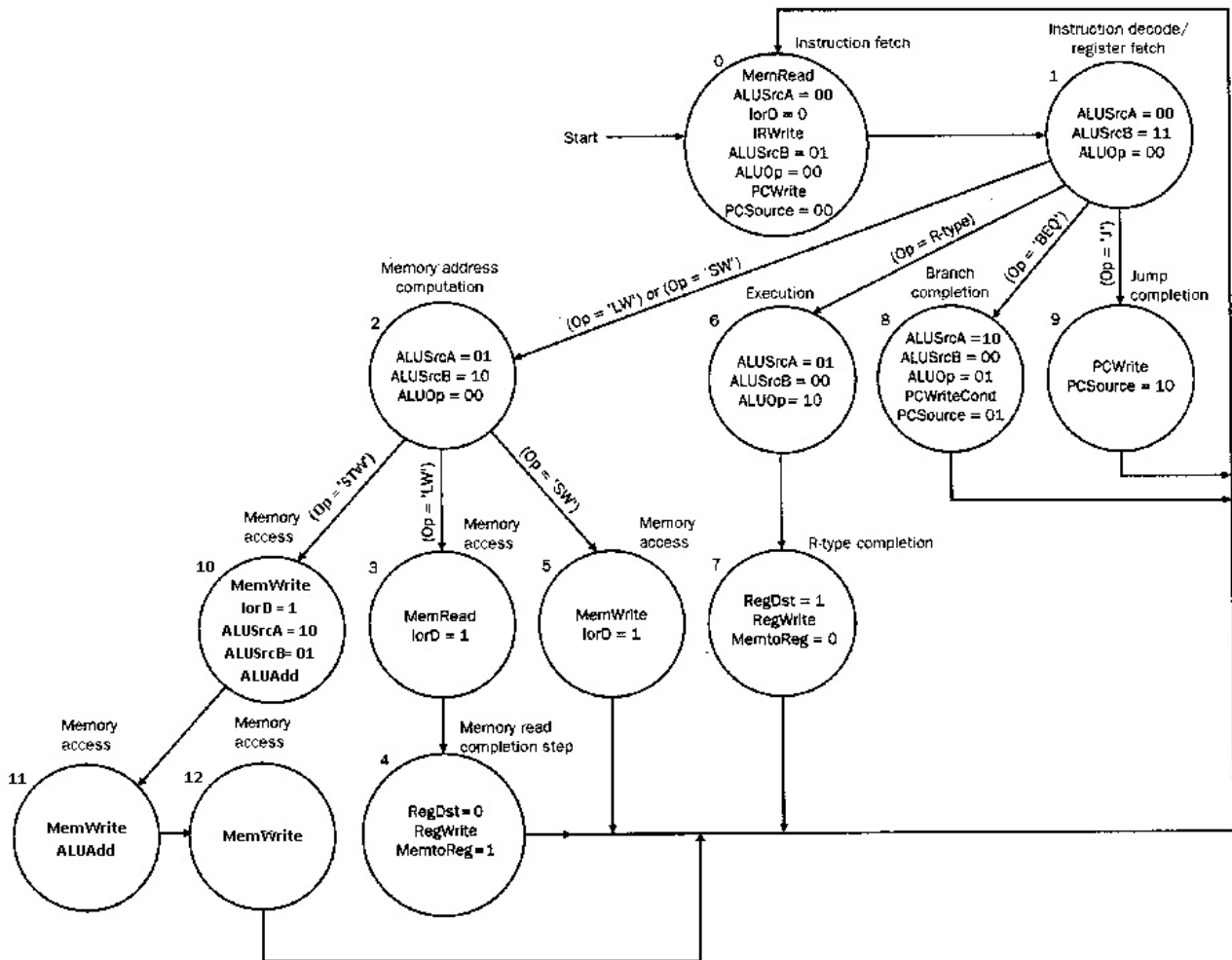


Figura 3: Nova máquina de estados com devidas alterações em ALUSrcA.