

Trabalho Final - MIPS em Verilog

Matheus Diniz, Davi Assis, Breno Gabrich e Breno Rodrigues

December 18, 2015

Passos necessários para executar uma instrução MIPS

1. Busca da instrução na memória, utilizando o PC para indicar a posição da instrução a executar
2. Decodificação da instrução e leitura dos registradores necessários
3. Executar a operação na ULA, calculando o endereço de memória a acessar no caso do lw e sw, executar as operações correspondentes nas instruções R, e efetuar a comparação por meio de subtrações nas instruções de beq.
4. Acesso a memória ou escrita do resultado.
5. Escrita do resultado em registros no caso do lw.

Controle:

O módulo do controle tenta implementar a máquina de estados do processador. A tarefa do controle é coordenar todos os outros componentes do processador para que as instruções possam ser executadas sem erro. Na implementação feita, há um contador que marca em qual etapa de execução o processador está.

Inicialmente ocorre a leitura de uma nova instrução, incluindo o incremento do PC. Em seguida, esses sinais são desligados, já que ela deve ser executada antes de receber outra. Nesse ponto o controle altera os sinais relevantes com base no tipo de instrução (R, BEQ, Jump, LW/SW). No caso do Jump, ocorre o desvio imediatamente, enquanto nos outros a ULA recebe os sinais relevantes. No ciclo seguinte a ULA já realizou suas operações e ocorrem os acessos à memória, no caso de LW ou SW, e a escrita no registrador, no caso de R. A execução da instrução chega ao fim, e o processador volta ao estágio de fetch.

Buscando a instrução na memória:

Como dito antes, para acessar a instrução que deve ser executada, primeiro utiliza-se o PC(program counter) para dizer em qual posição da memória de instruções está a memória que deve ser executada. Para isso, ele utiliza uma

variável do controle chamada PCesq que define se o PC deve ser incrementado para adquirir a próxima instrução que deve ser executada, ou permanecer na instrução atual.

Decodificação:

O passo de decodificação funciona da seguinte forma: após ser definida a instrução que deve ser executada, são buscados no bando de registradores, os registros necessários para que sejam feitas as operações com os mesmos. Além disso, existe uma variável definida pelo controle chamada EscReg que define se um registro deve ser alterado seu valor para um novo valor de registro(dado_a_ser_escrito) ou não. Após isso, esses registradores selecionados terão seus valores assimilados à duas variáveis de saída chamadas de dado_a e dado_b.

Execução:

Após possuir a instrução e os dados necessários, deve-se verificar qual operação deseja ser feita. Para isso, é verificado no módulo chamado Controle da Alu, que define qual operação deve ser feita. Esse módulo separa as operações R, das outras, e assim, define como a ALU deve se comportar. Essa operação é passada pelo controle à partir da variável ALUOp. O módulo ALU, por sua vez, possui somente operações básicas aritméticas. O resultado que a ALU retornada o resultado dos dados com a operação desejada.

Acesso a memória:

Para executarmos esse acesso precisamos de um banco de memória, este recebe dois sinais (EscreveMem e leMem) que informam a unidade de memória qual operação será feita, precisamos também do endereço da memória que queremos acessar, e no caso de querermos salvar um dado(sw) precisamos também do dado a ser escrito.

Os sinais vem do controle, o endereço da memória é calculado por meio da adição entre um offset e o endereço no registrador.

Calculo do “beq”:

As instruções de salto têm 3 parâmetros: os registradores cujo conteúdo deve ser comparado e o endereço do salto, usamos a ULA para subtrair os dois valores e verificar se são iguais a zero.

Quando isso acontece um sinal é validado e enviado para o PC, assim ele pode saber onde está a próxima instrução de acordo com o label utilizado.

Write back:

A etapa de write back , apenas escreve os resultados das instruções de volta nos registradores que escolhemos, tem apenas um MUX que retorna o resultado

onde precisamos que ele seja escrito no caso de uma operação R e pulamos essa etapa no caso de uma operação sw.