

Grupo: Dionatas Santos, Otávio Sales e Maria Julia Nolasco

1) Primeira Instrução

- a) Quais blocos existentes (se existirem) podem ser usados para esta nova instrução?
Essa instrução usa os blocos **instruction memory**, o **register file** (uma das portas de leitura existentes), o caminho que passa o imediato para o **ALU** e a porta de registro de escrita.
- b) Quais novos blocos devem ser adicionados para esta nova instrução?
Precisamos estender o **ALU** existente para fazer shifts também(adicionar operação SLL ALU).
- c) Quais sinais são necessários para se executar esta nova instrução?
Precisamos modificar os sinais de controle da **ALU** para que tenhamos suporte para SSL.

2) Segunda Instrução

- a) Quais blocos existentes (se existirem) podem ser usados para esta nova instrução?
A instrução addi usa os blocos **instruction memory**, **register file** (uma porta de leitura e uma de escrita), **ALU** (somar o valor do reg com o imediato) e o **control block**.
- b) Quais novos blocos devem ser adicionados para esta nova instrução?
Precisamos adicionar um **sign-extend** para transformar o imediato de 16 bits para 32 bits (antes de enviar para o mux que controla o segundo input da ALU).
- c) Quais sinais são necessários para se executar esta nova instrução?
Precisamos do sinal ALUSrc ativo para que o segundo input da ALU seja o imediato. MemtoReg inativo para que o valor enviado para o input Write Data venha da ALU.
RegWrite ativo para que o registrador Rt seja escrito com o valor do resultado da operação.
ALUOp 0010 para realizar uma adição.

3) Terceira Instrução

- a) Quais blocos existentes (se existirem) podem ser usados para esta nova instrução?
A instrução bne usa os blocos **instruction memory**, **register file** (duas portas de leitura de registrador, **ALU** (subtrair os valores e utilizar o bit zero para saber se são iguais ou não) e o **control block**.
- b) Quais novos blocos devem ser adicionados para esta nova instrução?
Precisamos adicionar um **sign-extend** para transformar o imediato 16 bits desloc em um imediato de 32 bits. Também precisamos adicionar um bloco "Shift Left 2" logo após o valor do imediato 32 bits para que o valor represente o valor real do Program Counter (já que os endereços são de 4 em 4 bits). Além disso, também precisamos de uma porta AND, uma porta OR e uma porta NOT.
- c) Quais sinais são necessários para se executar esta nova instrução?

ALUSrc inativo (para usar as duas saídas de dados de leitura do register file)
Branch inativo (não é possível utilizar Branch para BEQ e BNE ao mesmo tempo, por isso vamos criar um novo sinal de controle).

ALUOp 0110 para realizar uma subtração.

Novo sinal: BNE, ativo. Dessa forma, teremos o MUX que ativa o desvio quando Branch = 1 e Zero = 0 ou BNE = 1 e Zero = 1 (not equal).