



UNIVERSIDADE FEDERAL DA PARAÍBA
CAMPUS I - UNIDADE ACADÊMICA REITOR LYNALDO CAVALCANTI DE
ALBUQUERQUE
CENTRO DE INFORMÁTICA

RELATÓRIO DE PROJETO DE MICROARQUITETURA COM
MICROPROGRAMAÇÃO

LEONARDO DO NASCIMENTO PEIXOTO DA SILVA

JOÃO PESSOA
2022

LEONARDO DO NASCIMENTO PEIXOTO DA SILVA

**RELATÓRIO DE PROJETO DE MICROARQUITETURA COM
MICROPROGRAMAÇÃO**

Projeto final realizado para obtenção de
nota parcial da disciplina Arquitetura de
computadores I

Professor: Ewerton Monteiro Salvador.

JOÃO PESSOA

2022

1 INTRODUÇÃO

O seguinte relatório tem como objetivo mostrar o desenvolvimento e aplicação dos conceitos de microarquitetura e microprogramação vistos durante a disciplina de Arquitetura de Computadores I por meio da adição de uma nova instrução (a décima) na arquitetura MIPS.

Para isso, foi elaborada, baseando-se no caminho de dados, “tabelão” das microinstruções e diagrama de estados, a instrução BNE (*branch if not equal*) para o microprograma. Tal nova instrução possui a característica de realizar um novo desvio condicional no caso de os dois valores contidos na ULA (Unidade de Lógica e Aritmética) do computador sejam diferentes entre si.

A seguir, será apresentado todos os processos e alterações realizados, a partir do modelo base fornecido pelo professor, para que essa instrução pudesse estar presente em tal microarquitetura de computadores.

2 DESENVOLVIMENTO

Em primeiro lugar, para adição dessa nova instrução, BNE, o caminho de dados MIPS foi alterado. Conforme as figuras 1 e 2 abaixo, pode-se notar três modificações:

- 1) Criação de um novo sinal de controle (PCWriteCond2);
- 2) Adição de uma porta NOT para a flag ZERO (oriundo da ULA) em uma nova ramificação;
- 3) Adição de uma porta AND para unir o PCWriteCond2 e o inverso da flag ZERO.

[illegible]

Fonte: própria, 2022.

Um exemplo de Microarquitetura

O Caminho de Dados - MIPS

The diagram illustrates the MIPS data path, focusing on the control unit and its interactions with other components. The control unit is represented by a central oval labeled "Control" with "Op [5-0]" inside. It has several inputs and outputs:

- Inputs to Control:**
 - PCWriteCond (from a handwritten box labeled "PCWrite Cond 2")
 - PCWrite (from a handwritten box labeled "PCWrite")
 - MemRead (from a handwritten box labeled "MemRead")
 - MemWrite (from a handwritten box labeled "MemWrite")
 - MemtoReg (from a handwritten box labeled "MemtoReg")
 - IRWrite (from a handwritten box labeled "IRWrite")
 - Instruction [25-0] (from a handwritten box labeled "Instruction [25-0]")
- Outputs from Control:**
 - ALUOp (to a handwritten box labeled "ALUOp")
 - ALUSrcB (to a handwritten box labeled "ALUSrcB")
 - ALUSrcA (to a handwritten box labeled "ALUSrcA")
 - RegWrite (to a handwritten box labeled "RegWrite")
 - RegDst (to a handwritten box labeled "RegDst")

Handwritten annotations include a box labeled "PCWrite Cond 2" connected to the PCWriteCond input, a box labeled "PCWrite" connected to the PCWrite input, a box labeled "MemRead" connected to the MemRead input, a box labeled "MemWrite" connected to the MemWrite input, a box labeled "MemtoReg" connected to the MemtoReg input, a box labeled "IRWrite" connected to the IRWrite input, a box labeled "ALUOp" connected to the ALUOp output, a box labeled "ALUSrcB" connected to the ALUSrcB output, a box labeled "ALUSrcA" connected to the ALUSrcA output, a box labeled "RegWrite" connected to the RegWrite output, and a box labeled "RegDst" connected to the RegDst output. There is also a handwritten box labeled "Instruction [25-0]" at the bottom, connected to the Instruction [25-0] input.

Fonte: própria, 2022.

A lógica encontrada, então, para inserir a nova condição, mediante a valores diferentes, nesse caminho de dados foi a seguinte: (Observação: \wedge equivale a porta AND)

- Se a flag ZERO da ULA for 1 (está ativado), logo os valores na unidade lógica e aritmética são iguais ($A - B = 0$, logo $A = B$). Assim, o desvio condicional de valores iguais, $PCWriteCond \wedge \text{flag ZERO} = 1$, é acionado pela porta AND. Já o desvio condicional de valores diferentes resulta em um valor 0 pois a flag ZERO, que está 1, passa pela porta NOT e inverte seu número para 0, fazendo com que, caso o $PCWriteCond2$ esteja ligado também, $(\text{inv}(\text{flag ZERO}) = 0) \wedge (PCWriteCond2 = 1) = 0$.

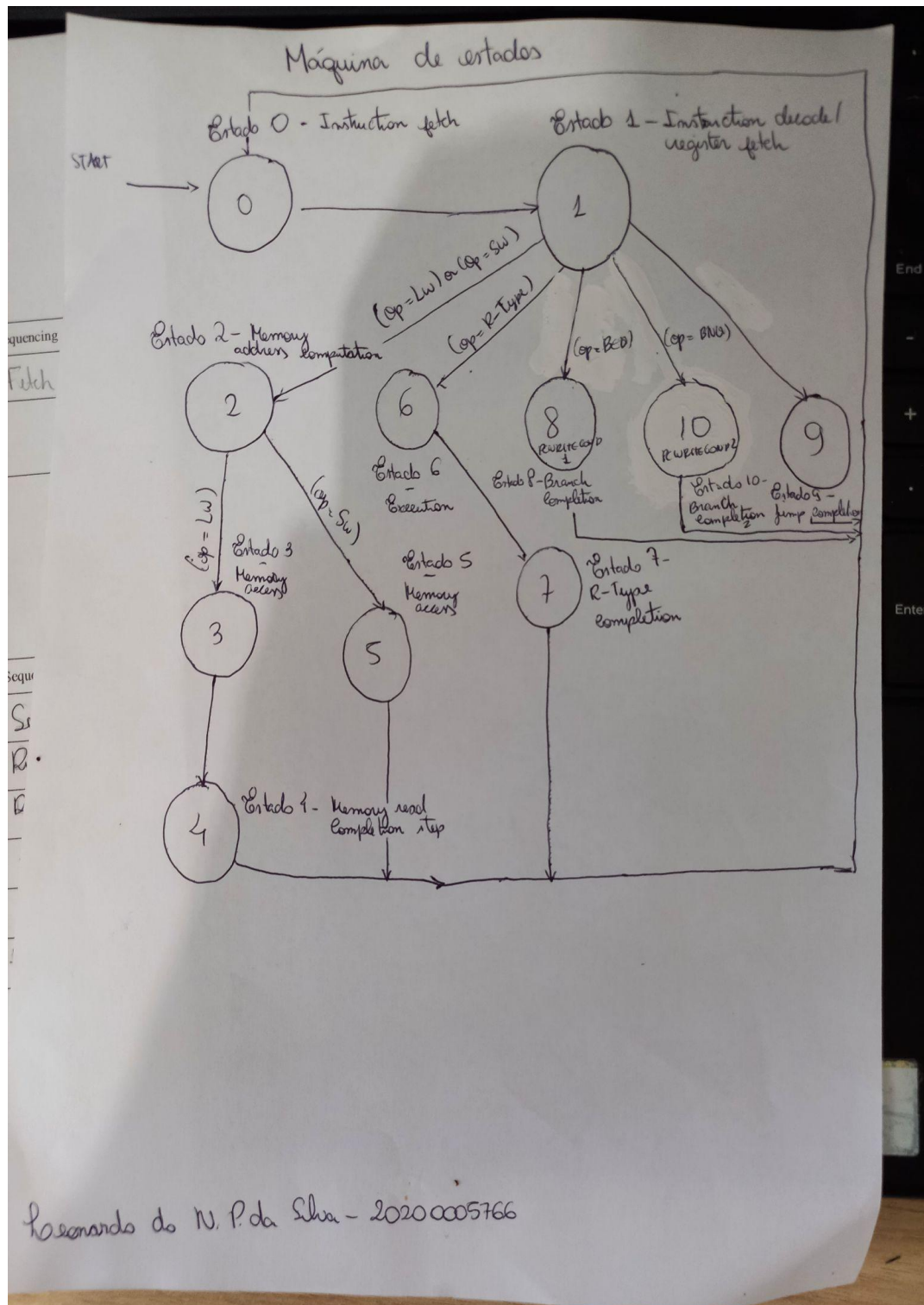
- Já caso no caso da flag ZERO for 0 (está desativada), os valores da ULA são diferentes, ou seja, ele tem que passar pelo outro caminho com o sinal de controle $PCWriteCond2$. A flag ZERO, que está 0, vai para porta NOT e fica como 1, e, com o sinal de controle $PCWriteCond2$ acionado também, vai para a porta AND e essa é ativada, pois $(\text{inv}(\text{flag ZERO}) = 1) \wedge (PCWriteCond2 = 1) = 1$.

Com o mecanismo dessa nova ramificação, PORTA NOT + PORTA AND, cria-se uma nova condição, fazendo, desse modo, que o registrador PC tenha três modos de ativação.

Implementada a instrução BNE no caminho de dados, teve-se que, por consequência, remodelar a máquina de estados e o “tabelão” das microinstruções:

Na máquina de estados, figura 3, foi adicionado um novo estado, o estado 10 (branch completion 2). Esse estado tem como o $op = BNQ$ e, ao invés de $PCWriteCond$ está ativado como no estado 8 - branch completion, o $PCWriteCond2$ é o sinal de controle ativado. Desse jeito, ficou-se com duas possibilidades para fazer desvio condicional com os valores da ULA: uma comparando se são iguais e outra comparando se são diferentes, por meio da operação BEQ ou da operação BNQ respectivamente.

Figura 3: Máquina de estados atualizada com o novo estado 10 correspondente ao BNQ.



Fonte: própria, 2022.

Já o tabelão de microinstruções foi modificado em relação a mais uma possibilidade na seção PCWRITE CONTROL, aparecendo, agora, uma nova área:

ALU: Escreve a saída da ALU no PC.

ALUOut-cond: Se a saída Zero da ALU estiver ativada, escreve em PC o conteúdo do registrador SaídaALU.

ALUOut-cond2: Se a saída Zero da ALU estiver desativa (valores contidos na ULA diferentes entre si), escreve em PC o conteúdo do registrador SaídaALU.

Jump Address: Escreve no PC o endereço de desvio da instrução.

Por último, o microprograma completo também foi modificado. Acrescentando mais uma linha de instrução formatada como se pode ver na figura 4:

Label: BNQ1

ALUControl: SUBT

SRC1: A

SRC2: B

RegisterControl:

Memory:

PCWriteControl: ALUOut - Cond2

Sequencing: Fetch

Figura 4: Microprograma completo com a microinstrução BNQ.

Leonardo do N. P. da Silva - 20200005166

MICROPROGRAMA COMPLETO

	Label	ALU Control	SRC1	SRC2	Register Control	Memory	PCWrite Control	Sequencing
E①	Fetch	add	PC	4		Read PC	ALU	Seq
E②	No label	add	PC	Bitshift	Read			Repatch
E③	Mem1	add	A	Extend				Repatch 2
E③	2w2					Read ALU		Seq
E④					Write MDR			Fetch
E⑤	SW2					Write ALU		Fetch
E⑥	R-format Func code		A	B				Seq
E⑦					Write ALU			Fetch
E⑧	Reg 1	SUBT	A	B			ALUOut cond	Fetch
E⑨	Jump1						Jump Address	Fetch
E⑩	BNQ1	SUBT	A	B			ALUOut Cond2	Fetch

(4)

3 CONCLUSÃO E CONSIDERAÇÕES FINAIS

Durante o andamento do projeto, notou-se alguns pontos mais difíceis para que a instrução BNE fosse pensada e implementada. Aqui estão eles:

- O primeiro é uma questão que o professor sempre abordou em sala de aula: atender os detalhes presentes em todo conteúdo de microarquitetura (além da disciplina em geral). Cada parte do trabalho (caminho de dados, máquina de estados, tabelão, microprograma) possui um universo de detalhamento do qual precisa ser entendido para que o todo dessas partes seja conectado e, assim, possa modificá-lo. Apenas pôde-se progredir no projeto ao entender (ou pelo menos ter uma noção básica) do papel que cada um deles exerce no projeto como um todo.
- O segundo é em relação ao ato de justamente conectar essas partes e tentar montar uma linha de raciocínio utilizando as informações de tudo. É como se fosse um quebra cabeça no qual tem que imaginar tudo funcionando, se movendo, ao mesmo tempo que precisa montar as “peças”.
- O terceiro se refere ao fato de saber a complexidade do problema em criar uma nova instrução. Em diversos momentos, houve um questionamento sobre o que estava se fazendo. Se era por esse caminho mesmo ou era algo mais fácil ou mais difícil.

Diante de tais comentários, observa-se que trabalhar com microarquitetura é uma atividade bastante desafiadora e, paralelamente, bagunçada. É uma mistura de hardware e software em que não há níveis bem definidos e uma fronteira estabelecida para cada um deles.