****

**Laboratório IV – Projeto CPU Monociclo**

Nome: Bruno Santos Fernandes Matrícula: 2021100784

Nome: João Victor Moraes Matrícula: 2021100244

Data: 04/12/22

**1. OBJETIVOS**

Entender os passos necessários para estender uma CPU para que ela dê suporte a execução de novas instruções na arquitetura (Instruction Set Architecture ou ISA) que não haviam sido implementadas no projeto inicial da CPU monociclo.

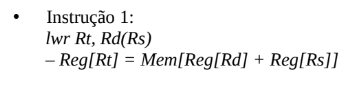
**2. ATIVIDADES**

A implementação da CPU monociclo da figura a seguir implementa apenas uma pequena parte do conjunto de instruções MIPS. Novas instruções devem ser adicionadas, mas a decisão depende, dentre outras coisas, do custo e complexidade da inclusão do suporte a estas instruções no caminho de dados e no controle do processador.

**3. QUESTÕES:**

**Para cada uma das 3 novas instruções apresentadas a seguir, responda:**

**a) Quais blocos existentes (se existirem) podem ser reaproveitados para esta nova instrução?**

****

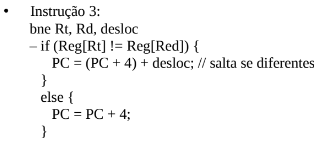
Na instrução 1 será necessários os bloco:

1. Pc
2. Memória de instruções
3. Registradores
4. Somador (add)
5. Alu
6. Mux
7. Memória de Dados
8. Controle



Na instrução 2 será necessários os bloco:

1. Pc
2. Memória de instruções
3. Somador (add)
4. Alu
5. Mux
6. Registradores
7. Controle



Na instrução 3 será necessários os bloco:

1. Pc
2. Memória de instrução
3. Somador (add)
4. Alu
5. Registradores
6. Mux
7. Controle
8. Porta And

**b) Quais novos blocos devem ser adicionados para esta nova instrução?**

**-> Instrução 1: lwr Rt, Rd(Rs)**

1. Acredito que não necessita de novos blocos

**-> Instrução 2: addi Rt, Rd, imed.**

1. Extensor de Sinal (16 bits -> 32 bits)
2. Mux

**-> Instrução 3: bne Rt, Rd, desloc**

1. Extensor de Sinal (16 bits -> 32 bits)
2. Deslocador ( << 2 bits)
3. Mux

**c) Quais sinais são necessários para se executar esta nova instrução?**

**-> Instrução 1: lwr Rt, Rd(Rs)**

1. Não foi adicionado novos Blocos

**-> Instrução 2: addi Rt, Rd, imed.**

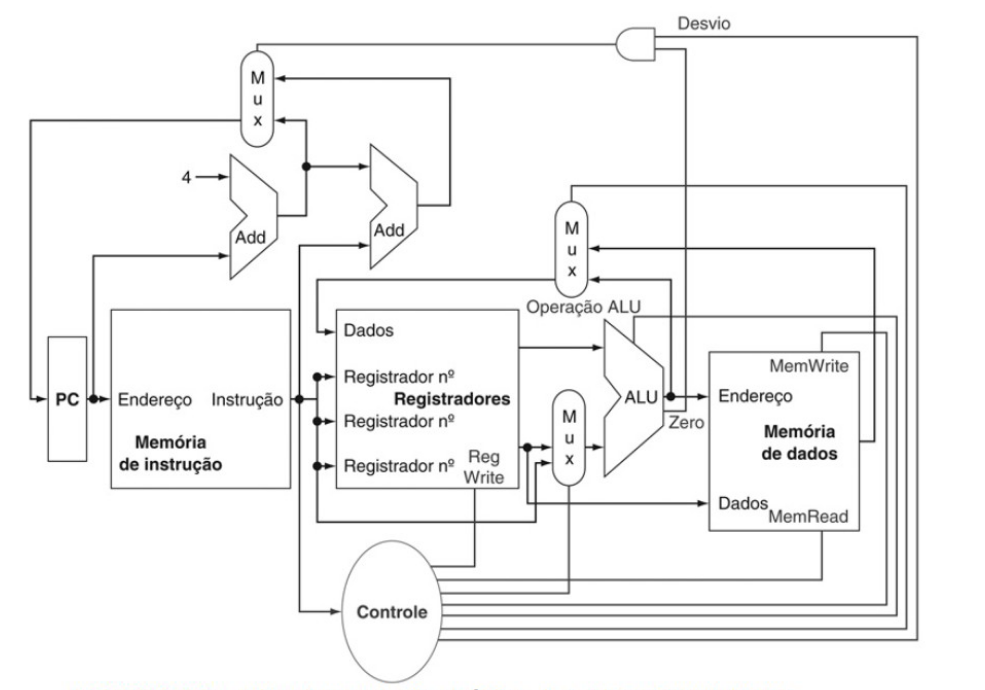
1. RegWrite para escrever no registrador de destino
2. Alusrc para saber se passa o valor de dados do registrador 2 ou o sinal estendido de outra operação, isso tudo através do mux
3. AluOp para controlar a operação que vai acontecer dentro da Alu, que seria uma add, do tipo R
4. MemReg para controlar o mux, será decidido se irá passar o valor do resultado da ALU ou os dados da memória

**-> Instrução 3: bne Rt, Rd, desloc**

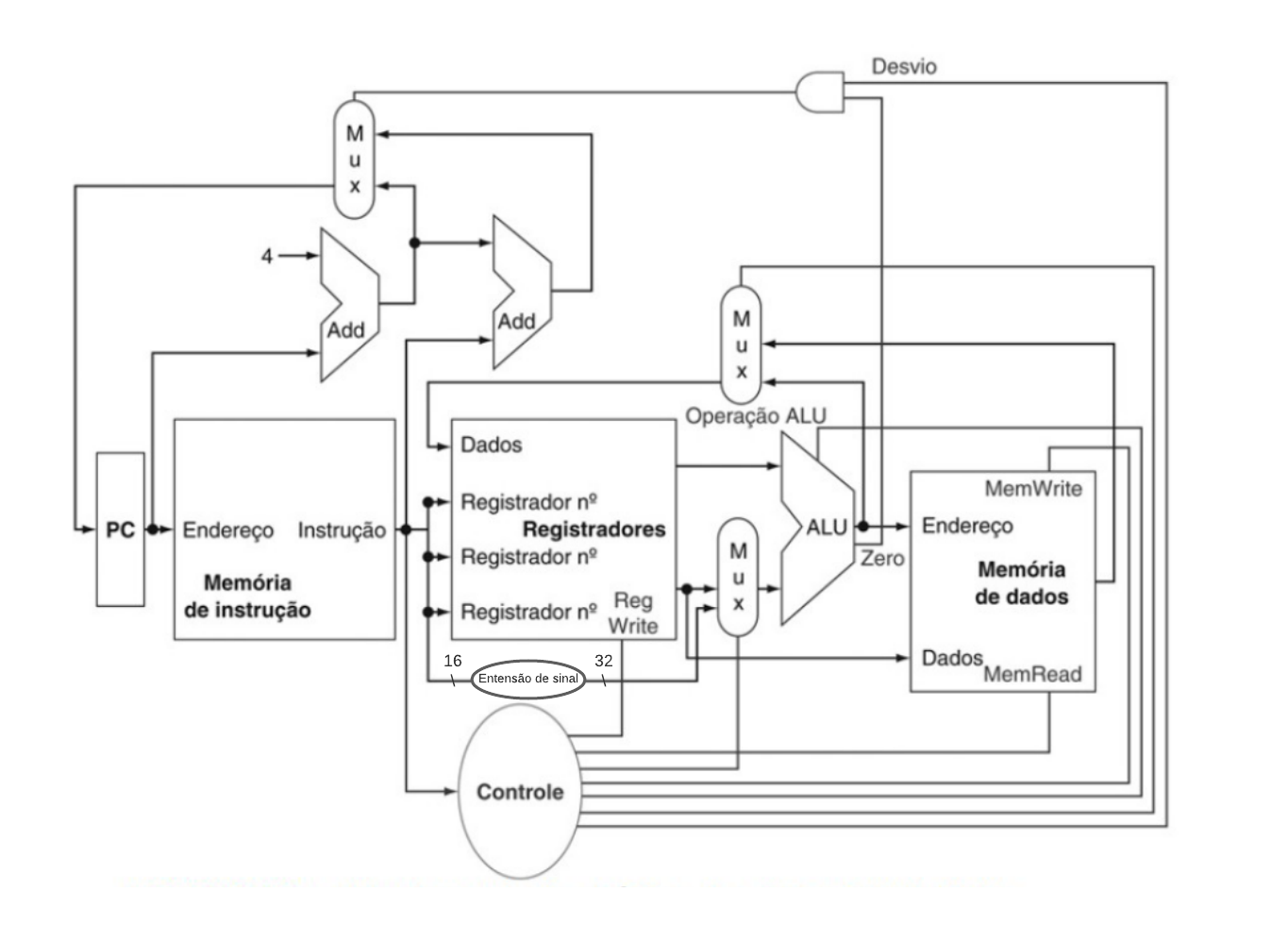
1. Criamos um sinal SelectBranch, para saber se o desvio é do tipo Beq ou Bne
2. AluOp, que vem da operação da ALU
3. Um sinal de Branch, para saber que é um desvio no mux
4. Alusrc para saber se passa o valor de dados do registrador 2 ou o sinal estendido de outra operação, isso tudo através do mux

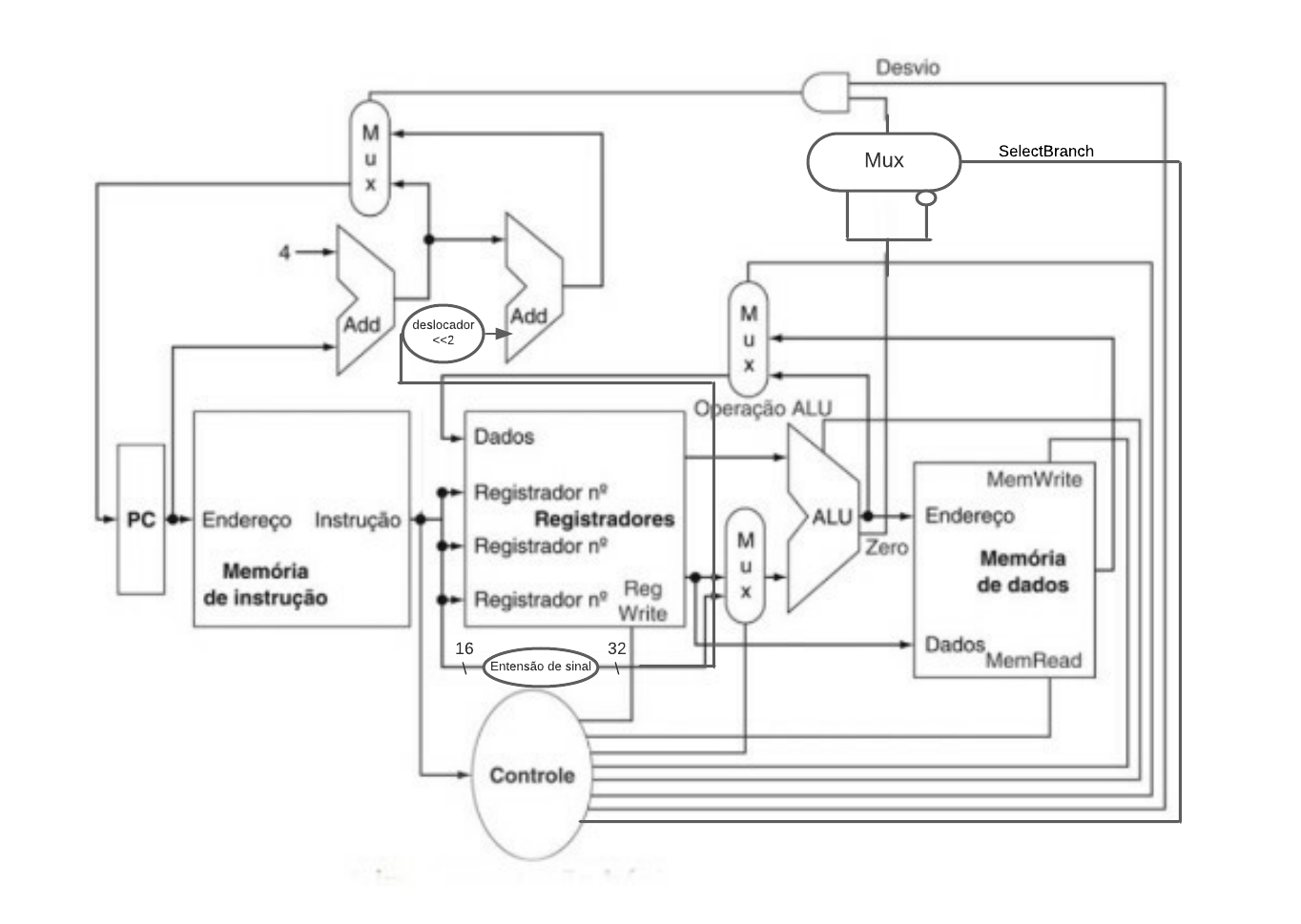
**d) Redesenhe o projeto com o suporte de cada 1 das novas instruções (serão 3 novos projetos).**

**-> Instrução 1 :**

****

**-> Instrução 2 :**

****

**-> Instrução 3 :￼**