

Vinícius Jeremias dos Santos (GRR20232361)

Profº Marco A. Zanata Alves

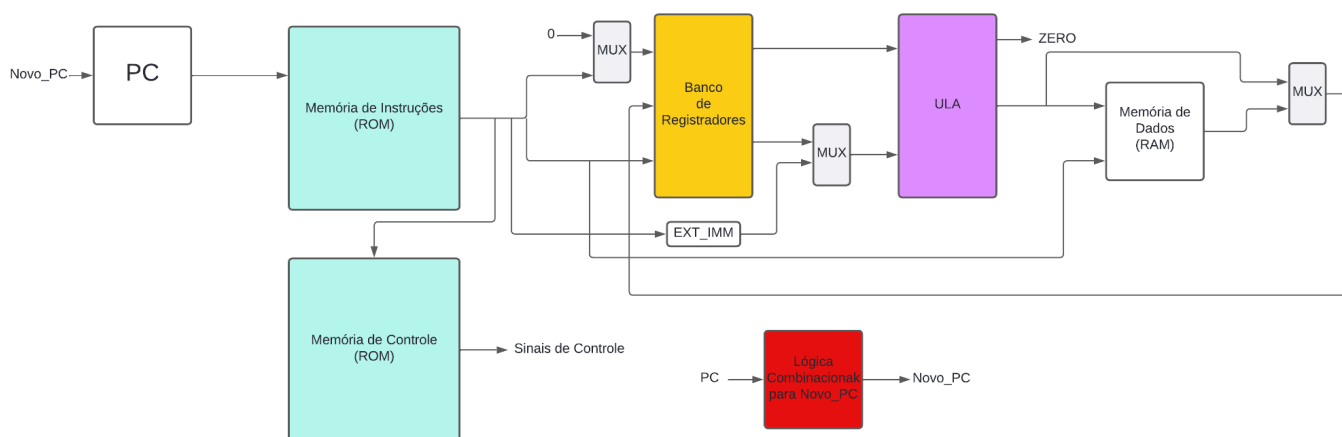
Arquitetura de Computadores

Trabalho - Arquitetura Sagui

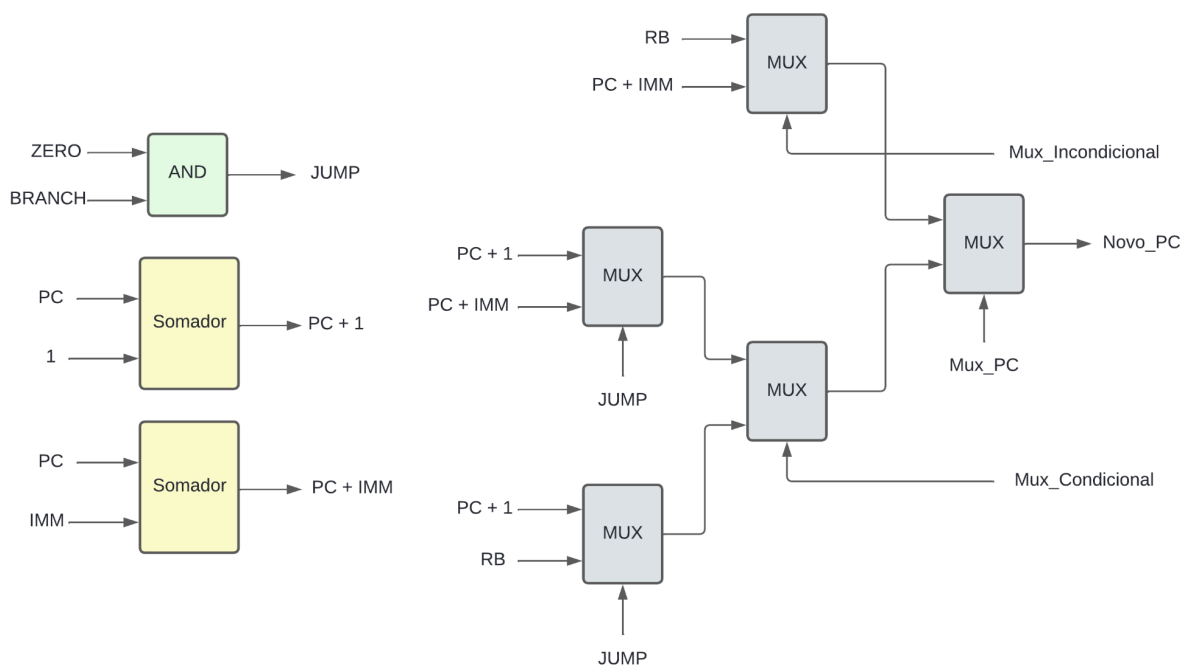
Relatório de desenvolvimento do processador Sagui, um processador simplificado capaz de executar uma série de operações aritméticas para calcular uma soma de vetores de 10 posições e outros programas simples.

Diagrama do Sagui

O diagrama do Sagui se assemelha bastante ao modelos conhecidos de diagramas para arquiteturas como MIPS e RISC-V, porém com suas próprias particularidades para executar devidamente o conjunto de instruções que foi proposto para essa arquitetura, incluindo alguns Mux adicionais e um componente particular na própria ULA.



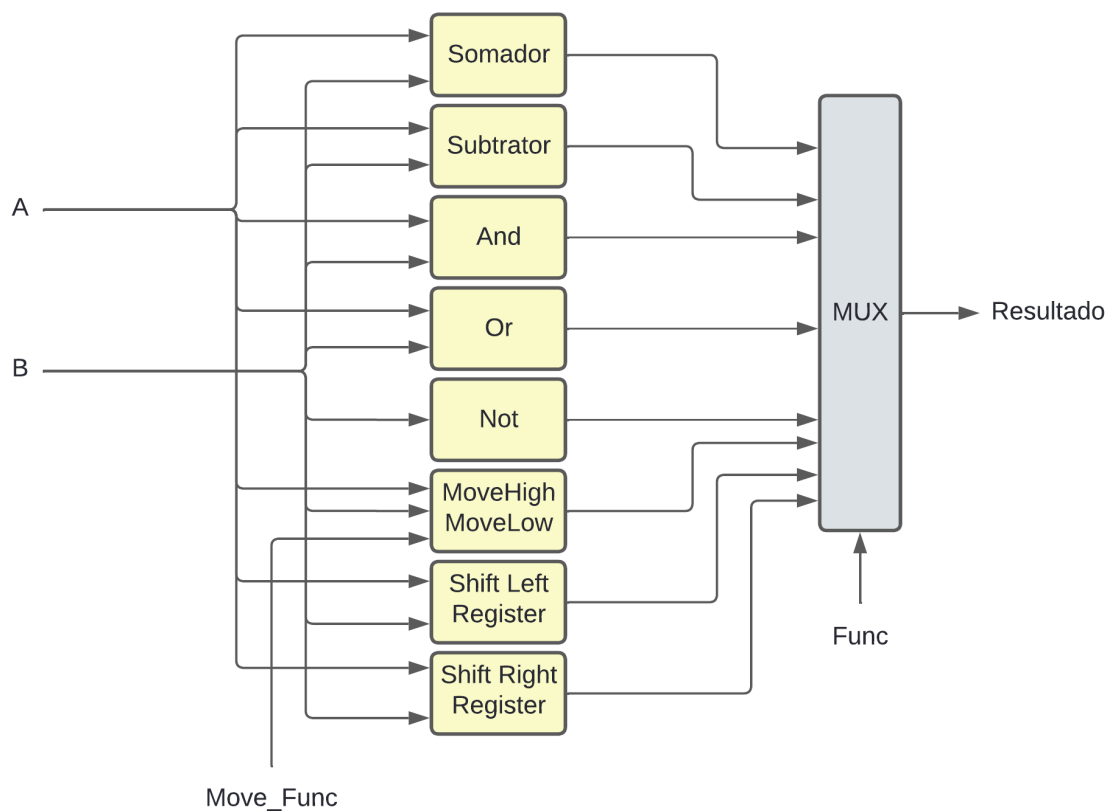
Em função de atender a algumas instruções de saltos condicionais e incondicionais, foi implementada uma lógica combinacional para gerar o próximo endereço apontado pelo registrador PC corretamente, pois instruções como brzr (Branch On Zero Register) e brzi (Branch On Zero Immediate) não poderiam ser implementadas de maneira trivial, dessa forma, se ampliarmos para observar o conteúdo que gera o novo endereço, obtemos a seguinte lógica:



Utilizando de um sinal de controle para Branch e alguns Mux, essas instruções podem ser utilizadas da forma correta, evitando saltos aleatórios durante a execução do programa inserido no processador.

O Projeto da ULA

Para a ULA, além das operações mais simples, para a implementação de instruções como movh (Move High) e movl (Move Low), foi adicionado uma lógica combinacional para gerar os valores a serem escritos no registrador, utilizando distribuidores para redirecionar os devidos bits aos campos apropriados. Este componente também permitiu que a instrução movr (Move Register) fosse implementada com facilidade, utilizando apenas dois bits para alternar entre os valores de A, B, movh e movl.



Por fim, junto com o .circ referente a arquitetura Sagui implementada no simulador Logisim Evolution, alguns arquivos úteis também estão incluídos, como:

- Data_ProgramaTeste: Os dados carregados na memória de dados utilizados pelo programa de teste de instruções.
- Data_ProgramaVetor: Os dados carregados na memória de dados utilizados pelo programa de soma de vetores.
- InstrucaoProgramas_asm: A codificação em Assembly do programa de testes e do programa de soma de vetores.
- TesteInstrucoes: Arquivo que contém o programa de testes que deve ser carregado na memória de instruções.
- ProgramaVetorInstrucoes: Arquivo que contém o programa de soma de vetores que deve ser carregado na memória de instruções.
- Sagui_Mem_Control: Arquivo que contém a codificação da memória de controle que deve ser carregado na memória de controle.