

Instituto Federal de Educação, Ciência e Tecnologia do Ceará Campus de Maracanaú

Curso de Bacharelado em Ciência da Computação

Arquitetura de Computadores

Prof. Otávio Alcântara

Prática de laboratório: caminho de dados do processador MIPS

Objetivos:

- Ter uma visão geral do caminho de dados do processador MIPS
- Entender o funcionamento dos principais componentes do processador MIPS
- Compreender como o circuito do processador executa instruções aritméticas simples

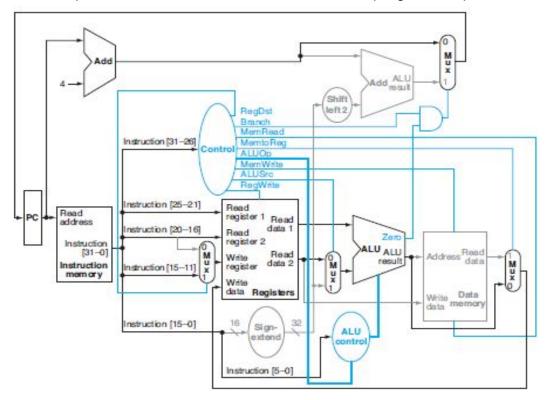
Ferramentas:

• Simulador Logisim Evolution

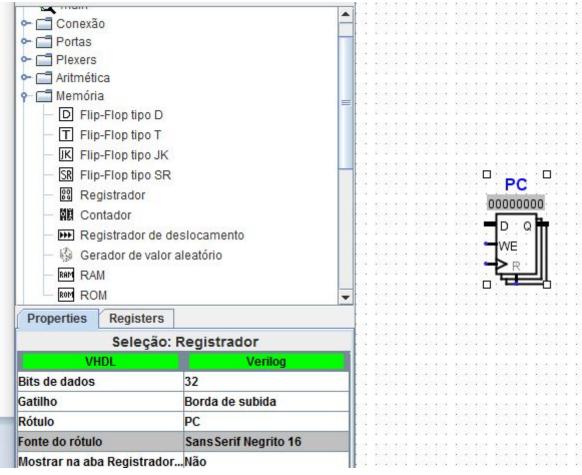
Descrição

 A meta da aula de hoje é reproduzir parte do caminho de dados do MIPs (figura abaixo) usando os componentes do simulador. Uma descrição (english only, sorry folks!) dos componentes pode ser encontrada no link:

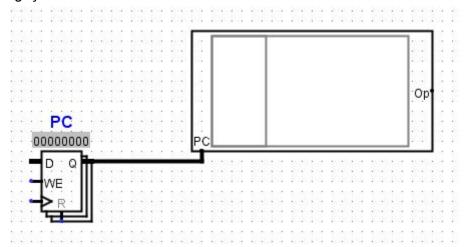
http://www.cs.cornell.edu/courses/cs3410/2017sp/logisim/components.html



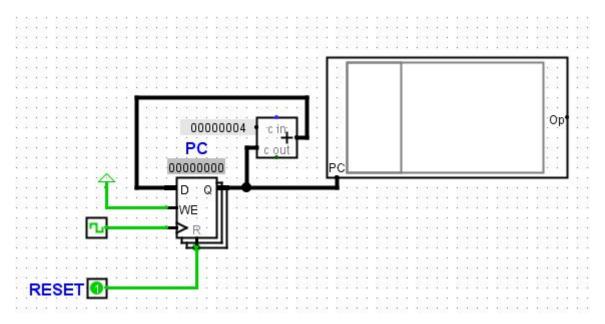
2. Vamos iniciar adicionando o registrador PC. Selecione o componente registrador na pasta Memória e configure-o para 32 bits.



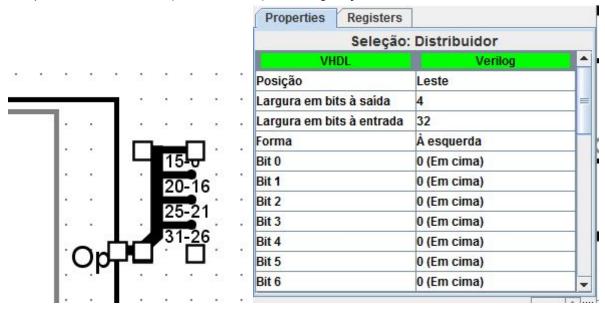
3. Adicione o componente MIPS Memory ROM (da aba CS3410 Components) e faça as ligações.



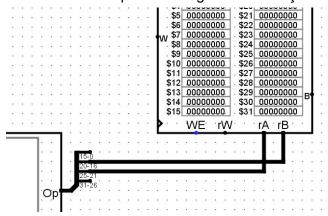
4. Adicione o componente Somador e faça as ligações abaixo. Simule o circuito e descreva o seu funcionamento.



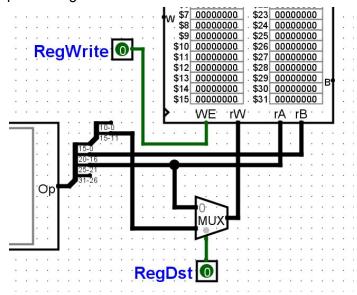
5. Agora vamos dividir a saída da memória de instruções em quatro grupos de bits usando o componente Distribuidor (aba Conexão). A configuração é mostrada abaixo.



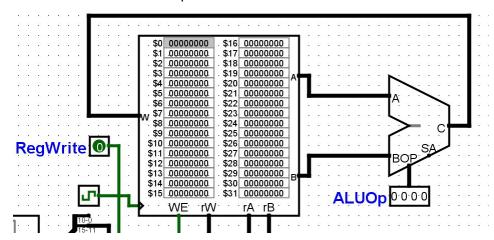
6. Adicione o componente Register File e faça as ligações.



7. O endereço do registrador de escrita pode vir de locais dependendo do formato da instrução. Adicione o Distribuidor e um Multiplexador para fazer as ligações abaixo. Atende para a largura de bits dos sinais.



8. Vamos adicionar o componente ULA.

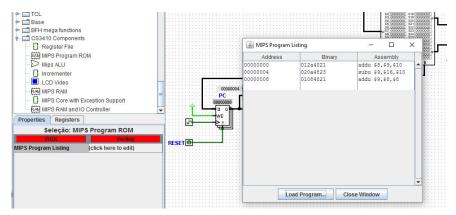


9. O circuito construído até agora está longe da versão final, mas já é capaz de executar instruções aritméticas simples. Contudo, os sinais de controle devem ser gerados manualmente. Para testar devemos carregar a memória de instruções, crie um arquivo de texto com o conteúdo abaixo.

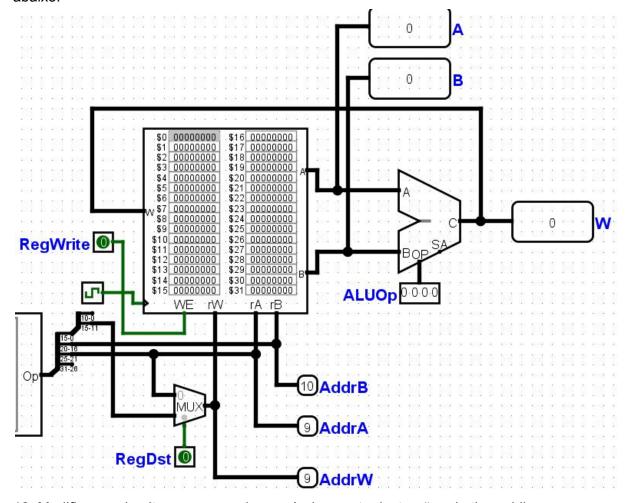
addu \$8,\$9,\$10 subu \$9,\$16,\$10 addu \$9,\$8,\$8

Este texto é o dump de um programa compilado usando o MARS, o código assembly desse programa segue abaixo.

.text addu \$t0, \$t1, \$t2 sub \$t1, \$s0, \$t2 add \$t1, \$t0, \$t0 10. Carregue o arquivo com o dump no componente MIPS Program ROM através da opção MIPS Program Listing. Então, clique em Load Program e selecione o arquivo de dump.



11. Agora podemos acompanhar a execução dessas três instruções. Gere os pulsos de clock e os sinais de controle da ULA para que as três instruções executem corretamente. Dê uma olhada no link no início do roteiro de prática para entender como operar a ULA. Para facilitar a visualização dos barramentos, sugerimos adicionar vários componentes Pino, ajustando a largura de bits e a base de numeração para decimal, conforme figura abaixo.



- 12. Modifique o circuito para que seja possível executar instruções do tipo addi.
- 13. Faça um belo relatório da prática para entrega no dia da prova!