UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
ARQUITETURA DE COMPUTADORES - 2023.1
PROF. DR. FRANCISCO HERON DE CARVALHO JUNIOR

ALUNOS:
542562 - BRUNO AGUIAR CARNEIRO SILVA
537606 - ERIK BAYERLEIN

Alterações na Unidade Lógica Aritmética (ULA)

Adotamos o código em python desenvolvido em sala de aula e fizemos nossas alterações para reduzir passos e executar melhor as operações realizadas por nosso processador. Assim implementamos as seguintes alteraçoes na ULA:

1. Ligação do barramento A a todos os registradores de B
Essa alteração foi efetuada na função read_regs, encontrada no arquivo do processador (ufc2x.py). Acrescentamos o parâmetro referente ao barramento A (linha 137), a fim de ler os registradores já imediatamente, sem passar os valores para o registrador H, economizando assim um (01) passo. Essa mudança requereu o acréscimo de mais 3 bits à leitura do barramento final, sendo 3 para A e 3 para B. Sendo necessário alterar também as linhas: ● linha 5: mudando o tamanho do array do firmware de 'L' = Long unsigned int de 32 bits para 'Q' = inteiro sem sinal de 64 bits ● linha 311-316: a execução de: ○ read_regs precisou receber os novos bits nos parametros ● o deslocamento de bits de alu, write_regs, memory_io, nex_instruction ○ todos receberam +3 no deslocamento >>
2. Acréscimo do circuito multiplicador
A ULA proposta pelo livro-texto foi implementada conforme a figura 1 no circuito anexado, identificada no arquivo circuitos.circ como ULA Livro. Pensando a partir dela, vimos que nem todas as combinações de 6 bits estão ocupadas por instruções válidas. A

primeira ideia que tivemos foi aumentar o número de bits do seletor de funções (conforme criado no circuito "ULA 1: 2bits com multi"). No entanto, essa solução se apresentou muito custosa, sendo necessário alterar o numero de bits das instruções da ula para acrescentar o multiplicador.

Como as questões pedidas não exigiam operações lógicas, resolvemos substituir uma delas para acrescentar o multiplicador (figura 2). Assim, removemos a instrução ~B, cuja operacao na ULA-BE.circ e no ufc2x era: 101100 (F0_F1_ENA_ENB_INVA_INC) pelo circuito multiplicador da figura 3 (disponivel para testa no arquivo ULA-BE.circ, circuito MULT 8BITS).

Ao fim da aplicação, temos uma ULA de 4 BITS funcionando como multiplicador de 8. Como se trata apenas da abstração, assim é possível termos uma ULA de 32 bits funcionando com um multiplicador de 32 bits.

3. Instruções do Firmware + Registrador Y

O registrador Y já estava pronto para uso, mas precisava de suas funções implementadas. Assim o fizemos.

No firmware, implementamos as operações:

- O registrador Y e suas operações
  - Memory[address] <-Y, (linha 69)
  - JAM Z, (linha 86)
  - Y <- Y + memory[address], (linha 60)
  - Y <- Y- memory[address], (linha 78)
- X = X * memory[address] (linha 48)

Como retiramos do registrador H a obrigatoriedade de receber as operações no barramento A, alteramos as funções que exigiam essa passagem H <- MDR, etc. Reduzindo uma linha de instrução.

4. Questões escolhidas

Escolhemos trabalhar com as questões:

- Q1: Potenciação, implementada no arquivo 'Q1pot.asm'
- Q3: CSW, implementada no arquivo 'Q3csw.asm'
- Q4: Fatorial, implementada no arquivo 'Q4fatorial.asm'

Abaixo as figuras listadas no texto e printadas do arquivo criado no Logisim.

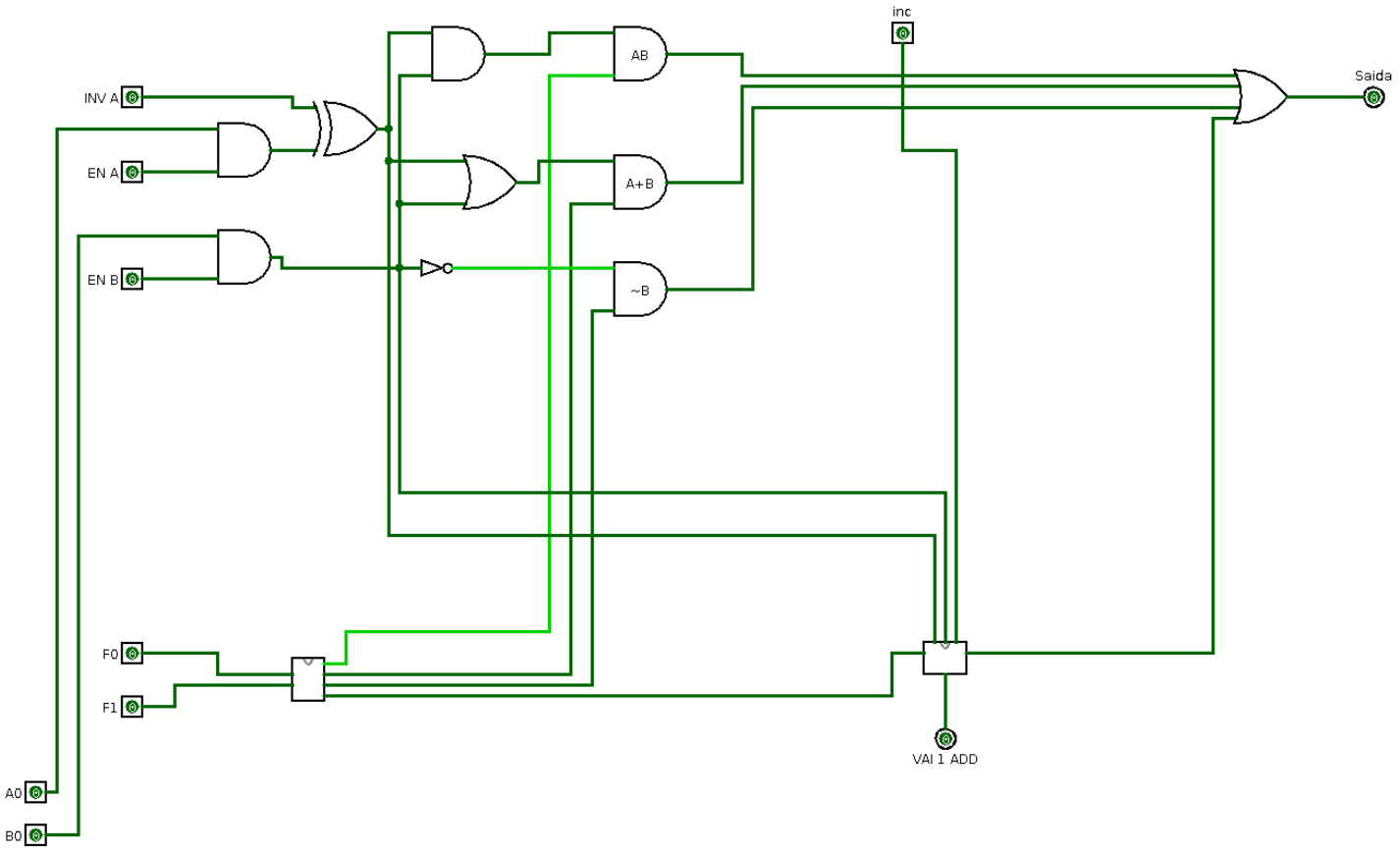

Figura 1 - ULA Livro implementada no Logisim

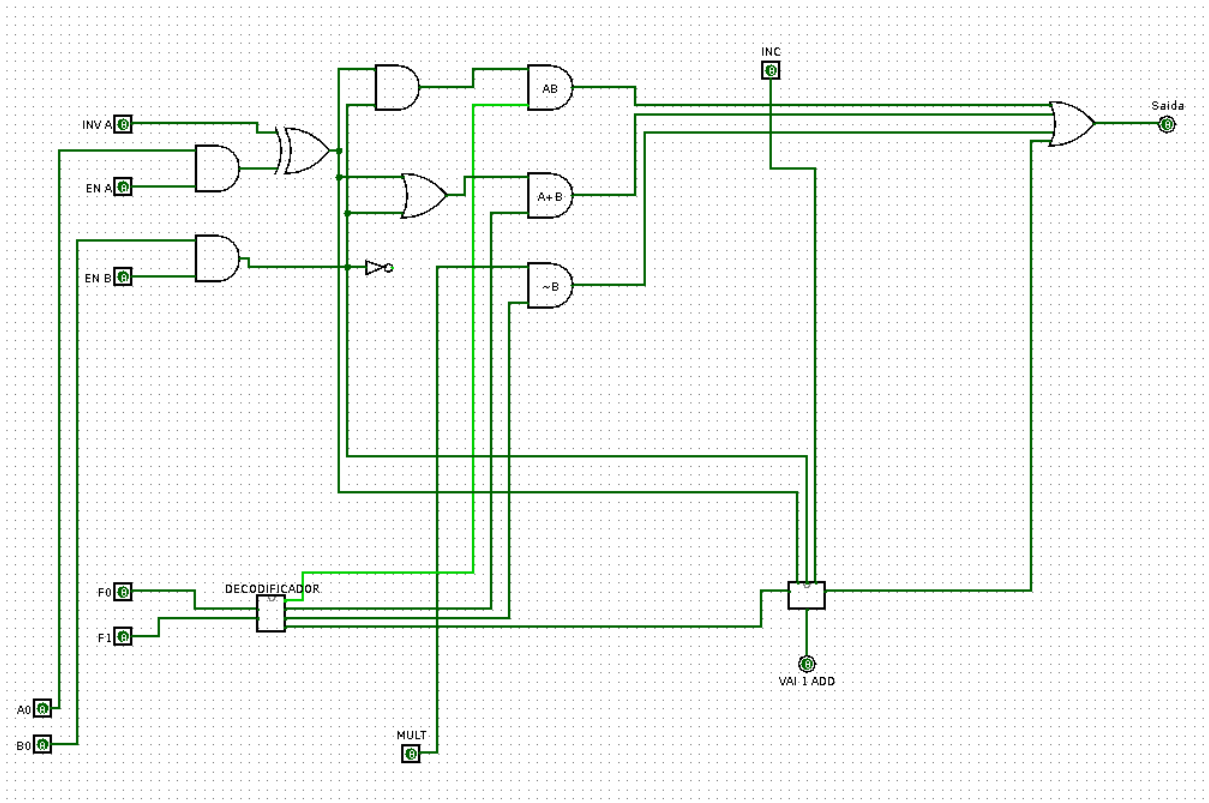

Figura 2 - operacao ~B removida e substituida pelo multiplicador

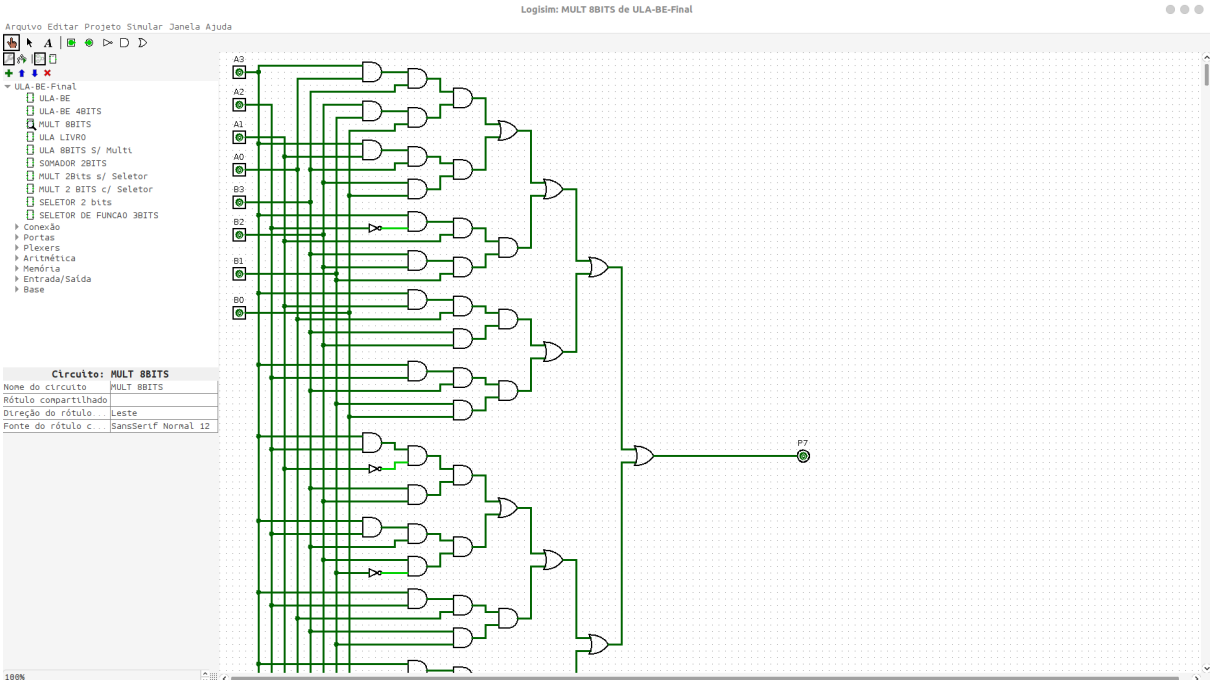

Figura 3 - circuito multiplicador em vista parcial. No arquivo enviado junto com a microarquitetura é possível visualizar e testar o circuito.

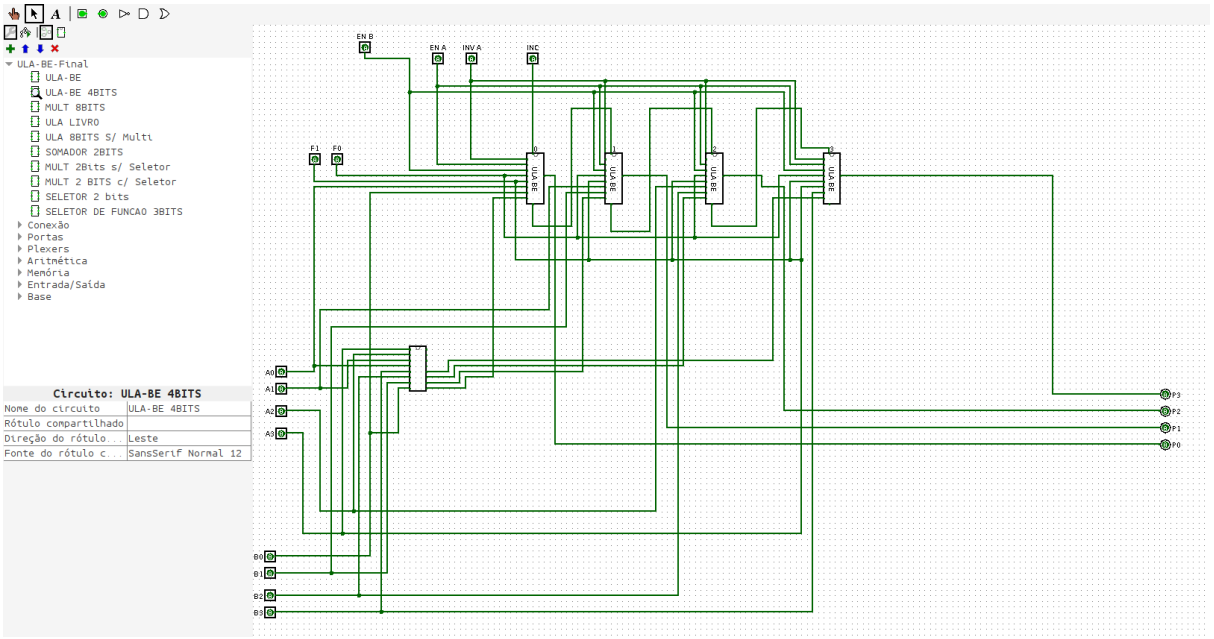

Figura 4 - ULA de 4 bits implementada com o multiplicador no lugar do ~B. As demais saídas do multiplicador podem ser incorporadas às outras ULAs a fim de concluir a arquitetura.