

# Relatório

# Trabalho Semestral – Mico X1

Lucas Emanuel de Oliveira Santos - GRR20224379

Universidade Federal do Paraná – UFPR

Curitiba, Brasil

## I. Escolha do Somador

Para realizar a soma foi escolhido o somador seletor de carry, buscando um equilíbrio entre custo, área e a velocidade do circuito, assim como uma fácil e compreensível implementação. O somador seletor proporciona um aumento significativo de velocidade( se comparado ao somador comum) e uma adição moderada de área e custo. Como a metodologia utilizada na escolha do somador foi o equilíbrio, um somador que fosse muito rápido, porém com uma área muito maior e consequentemente um custo muito maior não faria sentido. Com todos esses fatores, aliados a uma implementação simples, o somador seletor de carry se mostrou o ideal para a situação.

Como este é um somador de 4 bits, foram dispostos 8 blocos em série do mesmo para formar um somador de 32 bits. Vale destacar que como o primeiro somador não importa se for ou não seletor, foi utilizado um somador comum para economizar área.

## II. Programa de Teste

O programa de teste foi feito em binário e posteriormente transformado para hexadecimal como é mostrado na tabela I. Para isso, foi utilizado um conversor online de binário para hexadecimal e um arquivo .hex para adicionar o programa na memória do processador.

Tabela I

[illegible]

Os primeiros 4 bits do número binário representam a operação que está sendo realizada, os próximos 12 bits correspondem a utilização dos Registradores A, B e C respectivamente (4 bits para cada), e os 16 bits restantes correspondem a constante utilizada nas operações.

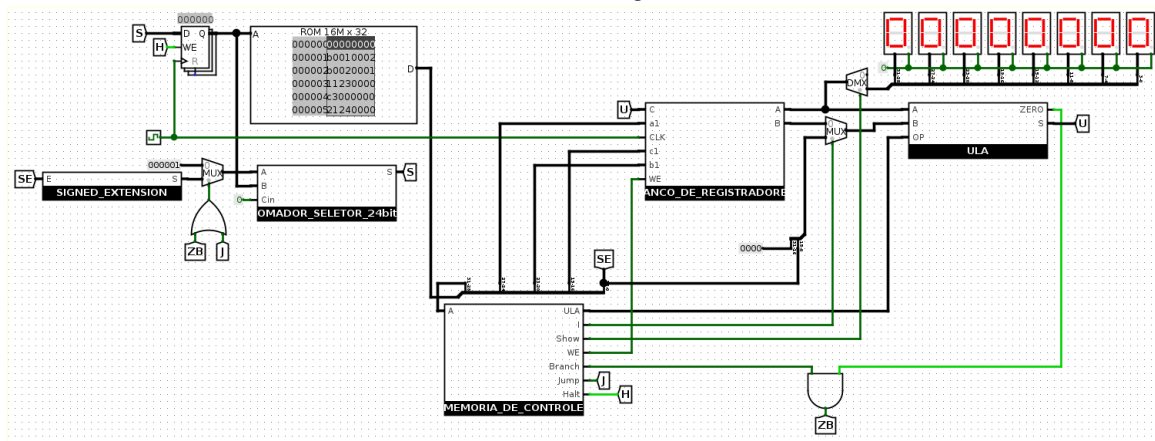
O programa teste teve como objetivo realizar todas as operações do processador e mostrar o resultado no display para analisar a solução e corrigir qualquer eventual erro.

O primeiro passo na produção do programa foi adicionar dois valores na memória para realizar as operações, para isso foi utilizado a instrução addi, com os dois valores nos registradores o próximo passo foi realizar todas as demais instruções e analisar os resultados mostrados no display. Para o Branch e o Jump foi utilizado outro addi, assim o programa faria o número da constante adicionada no addi de Saltos Incondicionais. Por fim foi utilizado o Halt para encerrar o programa.

### III. Bloco Operativo

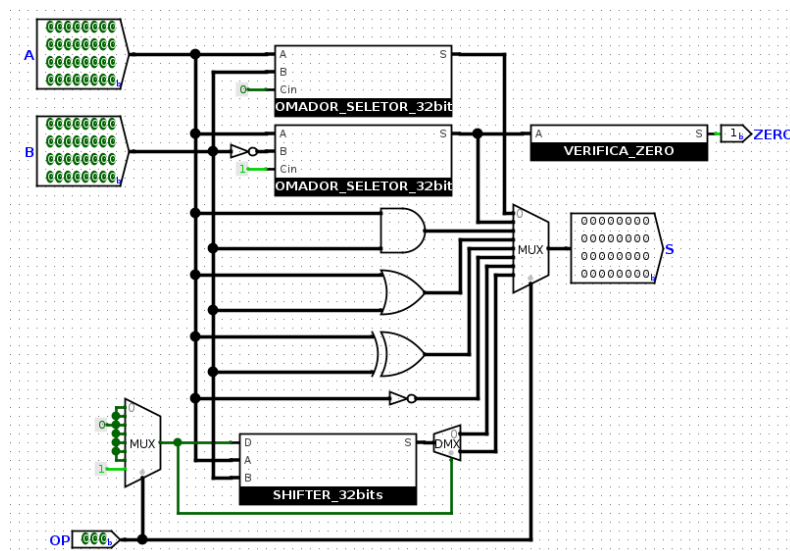
O Circuito principal do processador é mostrado na imagem I.

Imagem I



A ULA do processador é mostrada na Imagem II.

Imagem II



### IV. Referências

Curso de Projetos Digitais e Microprocessadores. Laboratório HIPES, Youtube, 2022. Disponível em: [https://www.youtube.com/watch?v=tXdReO5AWO0&list=PL\\_9px37PNj6oIJcyAxUSZ-Z8yLA2XS4gB](https://www.youtube.com/watch?v=tXdReO5AWO0&list=PL_9px37PNj6oIJcyAxUSZ-Z8yLA2XS4gB). Acesso em 09/01/2023 à 13/01/2023.

Rapid Tables. Disponível em: <https://www.rapidtables.com/convert/number/binary-to-hex.html>. Acesso em 14/01/2023.

Binary Convert. Gronlin, François, 2009. Disponível em: [https://www.binaryconvert.com/convert\\_unsigned\\_int.html](https://www.binaryconvert.com/convert_unsigned_int.html). Acesso em 14/01/2023.