# 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 metologia 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

Operação	Binario	Hexa
nop	000000000000000000000000000000000000000	00000000
addi	10110000000000100000000000000010	b0010002
addi	10110000000001000000000000000001	b0020001
add	0001000100100011000000000000000000	11230000
show	110000110000000000000000000000000000000	c3000000
sub	0010000100100100000000000000000000000	21240000
show	110001000000000000000000000000000000000	c4000000
and	001100010010010100000000000000000000000	31250000
show	110001010000000000000000000000000000000	c5000000
or	0100000100100110000000000000000000	41260000
show	110001100000000000000000000000000000000	c6000000
xor	0101000100100111000000000000000000	51270000
show	110001110000000000000000000000000000000	c7000000
not	011000100000100000000000000000000000000	62080000
show	110010000000000000000000000000000000000	c8000000
<<	011100010010100100000000000000000000000	71290000
show	110010010000000000000000000000000000000	c9000000
>>	1000000100101010000000000000000000	812a0000
show	110010100000000000000000000000000000000	cA000000
ori	100100100000101100000000000000001	920b0001
show	110010110000000000000000000000000000000	cb000000
xori	101000100000110000000000000000001	a20c0001
show	110011000000000000000000000000000000000	cc000000
branch	111000100001000000000000000000011	e2100003
addi	10110010000001000000000000000001	b2020001
jump	110100000000000011111111111111111	d000fffe
halt	111100000000000000000000000000000000000	f0000000

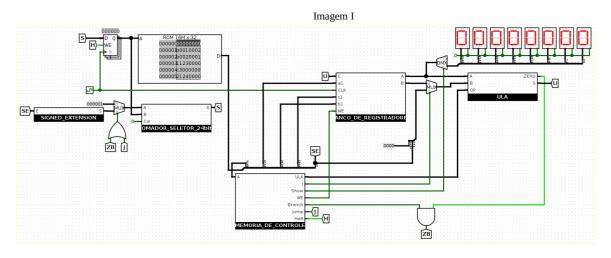
Os primeiros 4 bits do número binário representam a operação que está sendo realizada, os próximos 12 bits correspondem a utilização dos Registradores A, B e C respectivamente(4 bits para cada), e os 16 bits restantes correspondem a constante utilizada nas operações.

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.

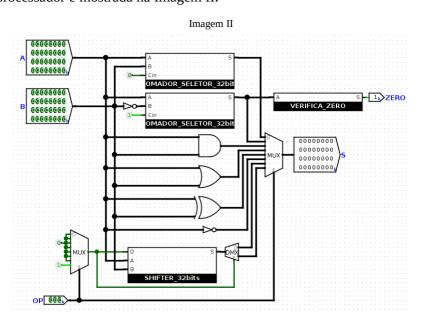
O primeiro passo na produção do programa foi adicionar dois valores na memória para realizar as operações, para isso foi utilizado a instrução addi, com os dois valores nos registradores o próximo passo foi realizar todas as demais instruções e analisar os resultados mostrados no display. Para o Branch 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.



A ULA do processador é mostrada na Imagem II.



IV. Referências

Curso de Projetos Digitais e Microprocessadores. Laboratório HIPES, Youtube, 2022. Disponível em: <a href="https://www.youtube.com/watch?v=tXdReO5AWO0&list=PL\_9px37PNj6oIJcyAxUSZ-Z8yLA2XS4gB">https://www.youtube.com/watch?v=tXdReO5AWO0&list=PL\_9px37PNj6oIJcyAxUSZ-Z8yLA2XS4gB</a>. Acesso em 09/01/2023 à 13/01/2023.

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

Binary Convert. Gronlin, François, 2009. Disponível em: <a href="https://www.binaryconvert.com/convert\_unsigned\_int.html">https://www.binaryconvert.com/convert\_unsigned\_int.html</a>. Acesso em 14/01/2023.