

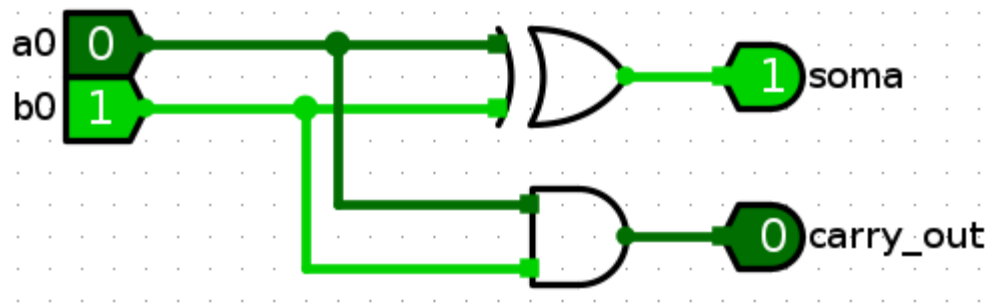
**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

GABRIEL VARGAS BENTO DE SOUZA

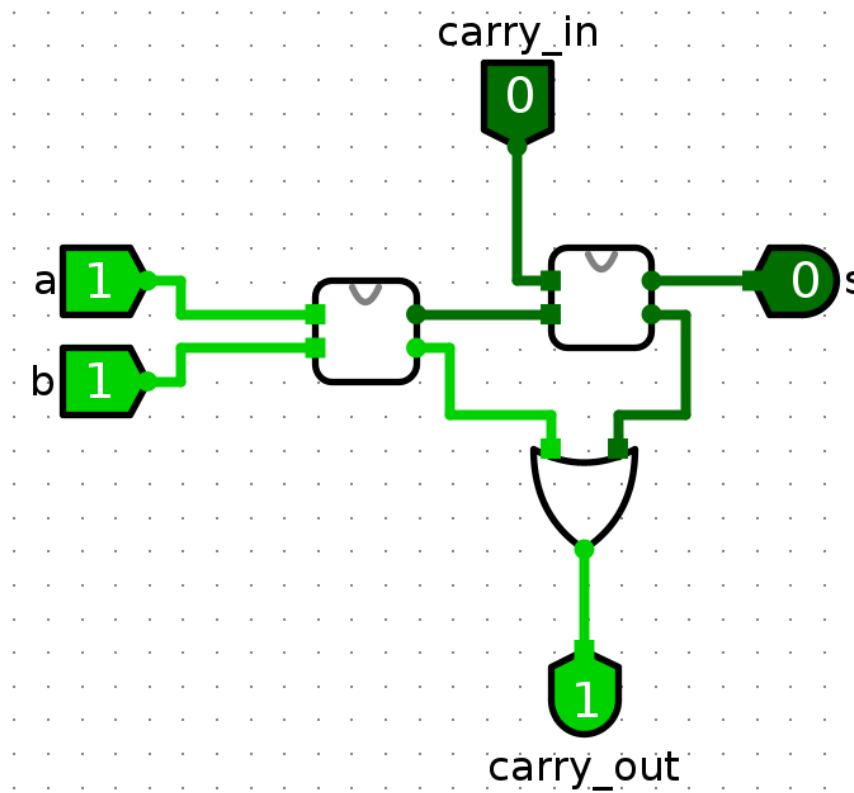
**EXERCÍCIO PRÁTICO 01**  
Arquitetura de Computadores II

Belo Horizonte - MG  
**2023**

### Exercícios:



**Figura 01** : ½ somador no Logisim



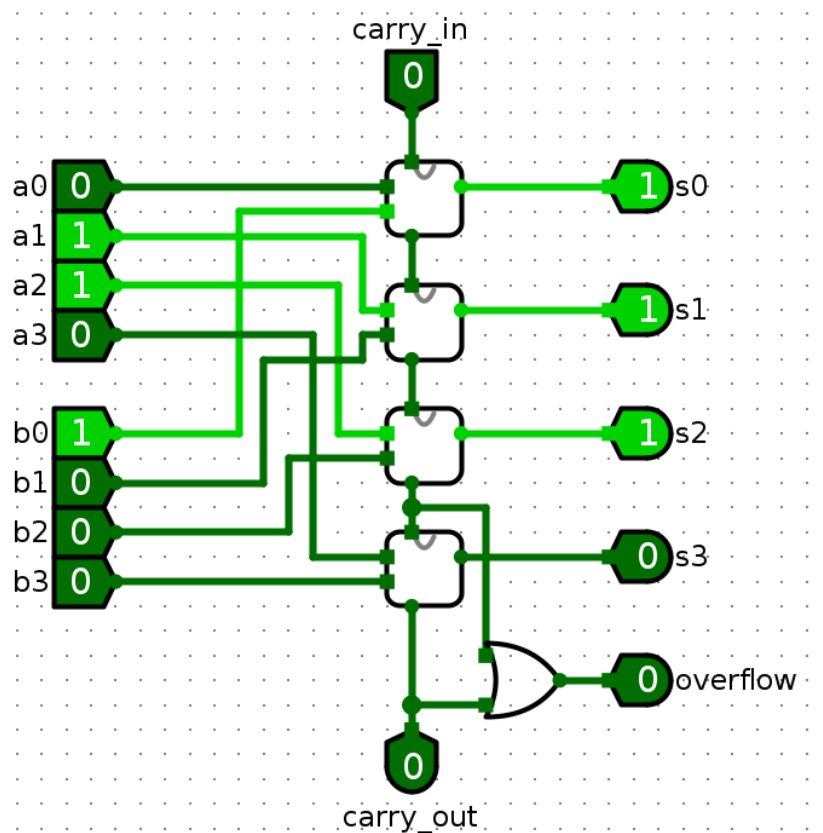
**Figura 02** : Somador completo no Logisim

a0	b0	soma	carry_out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

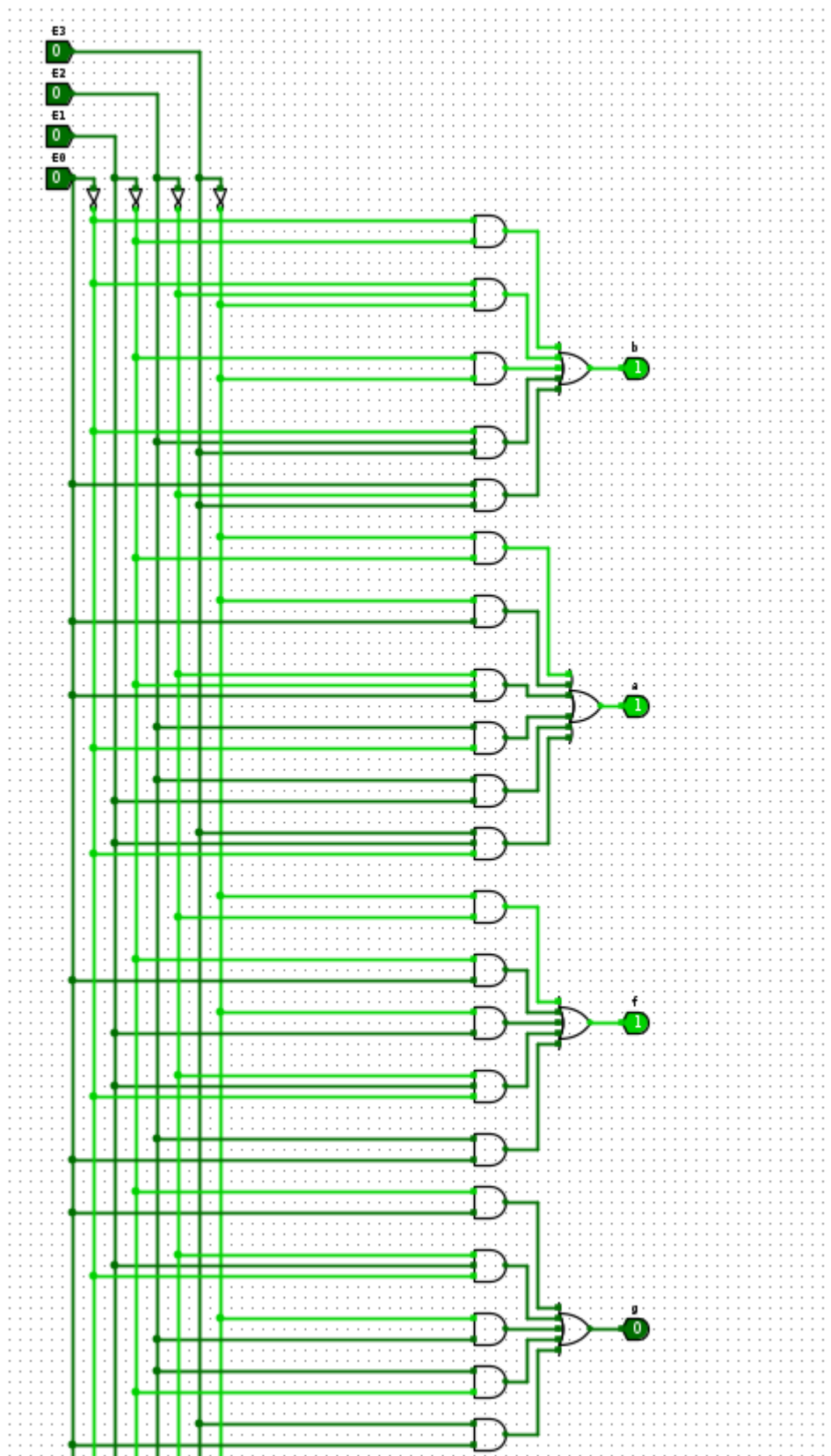
**Figura 03** : Tabela verdade ½ somador

carry_in	a	b	s	carry_out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

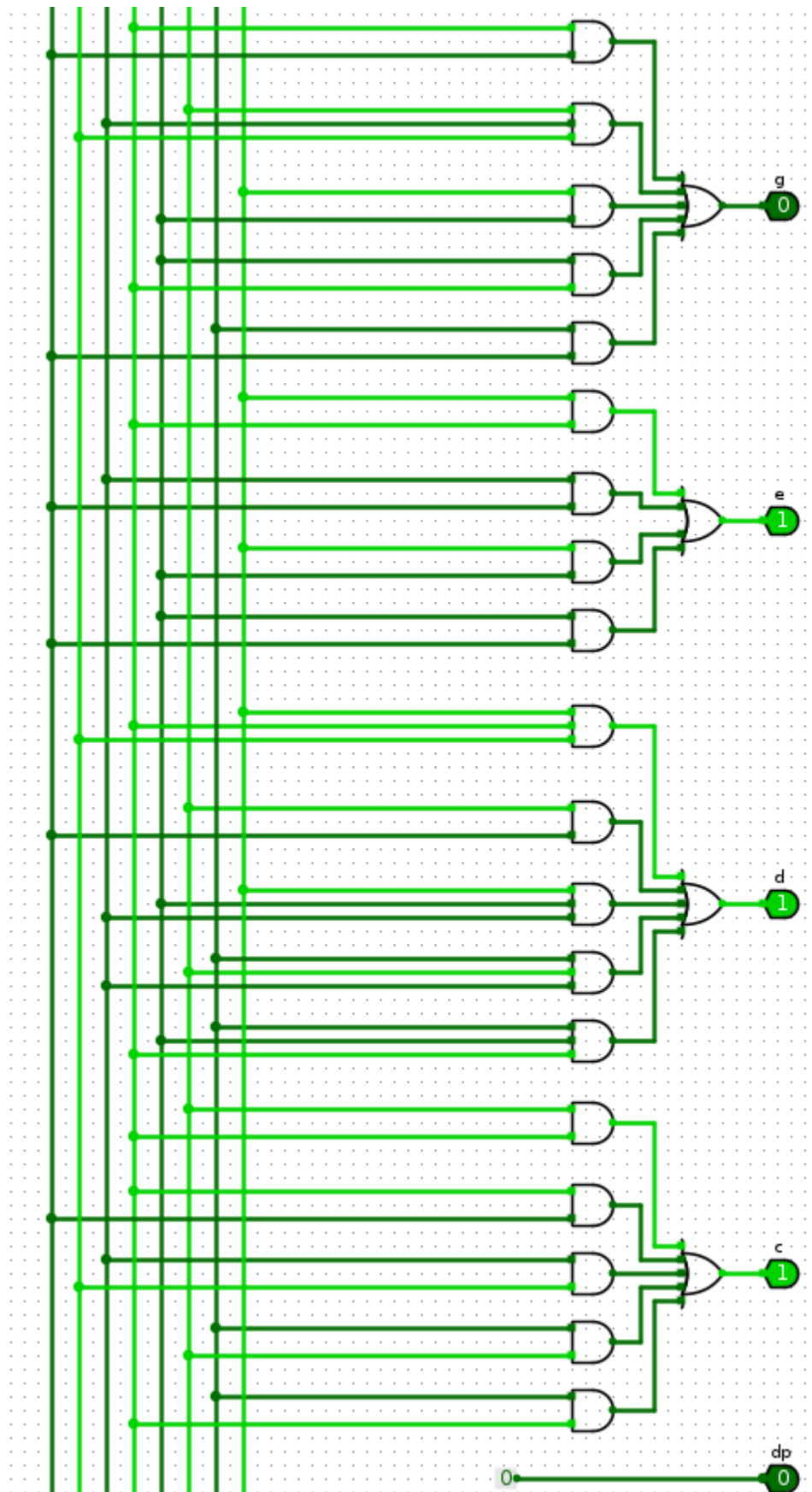
**Figura 04** : Tabela verdade somador completo



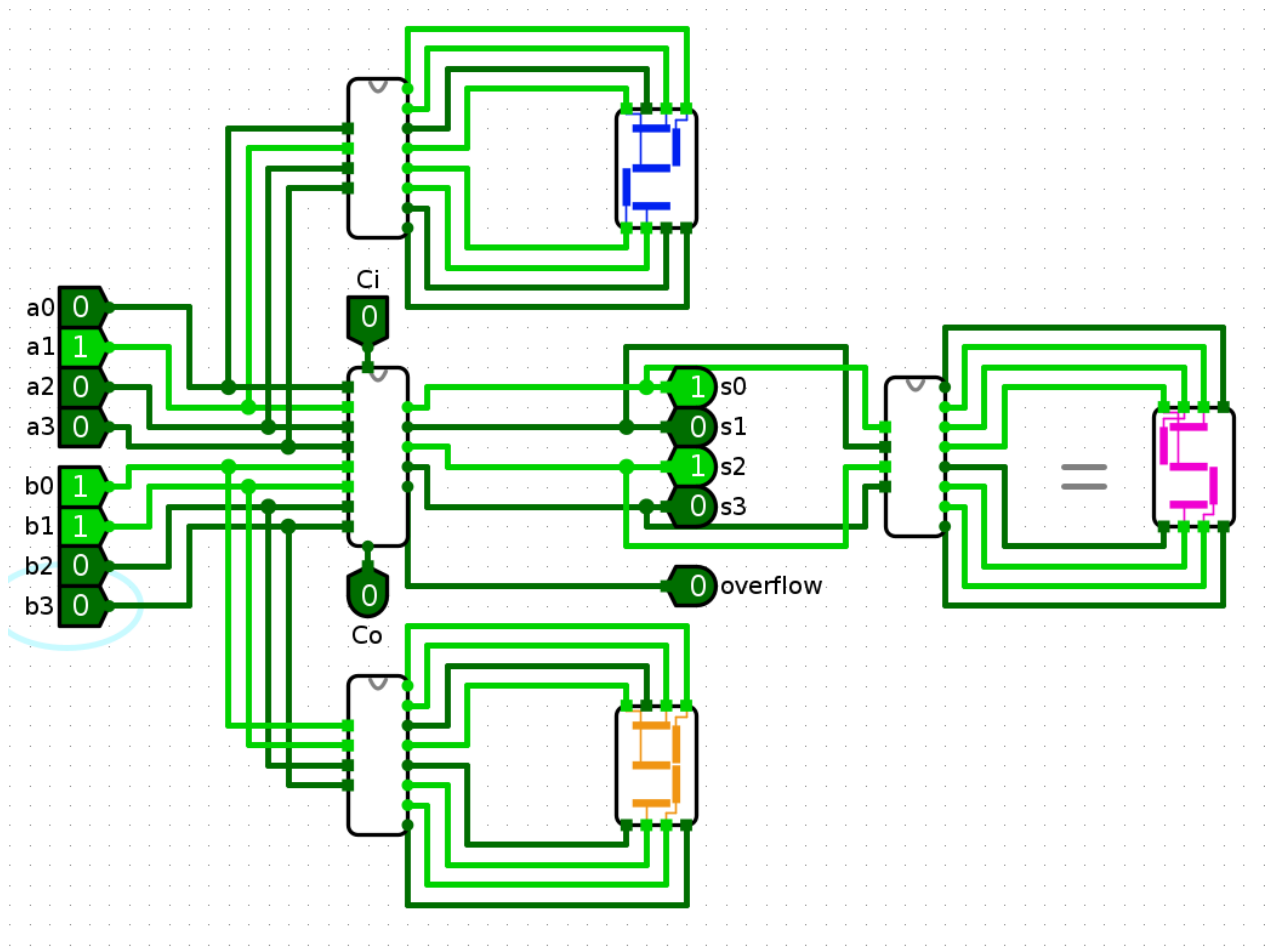
**Figura 05** : Somador completo para 4 bits no Logisim



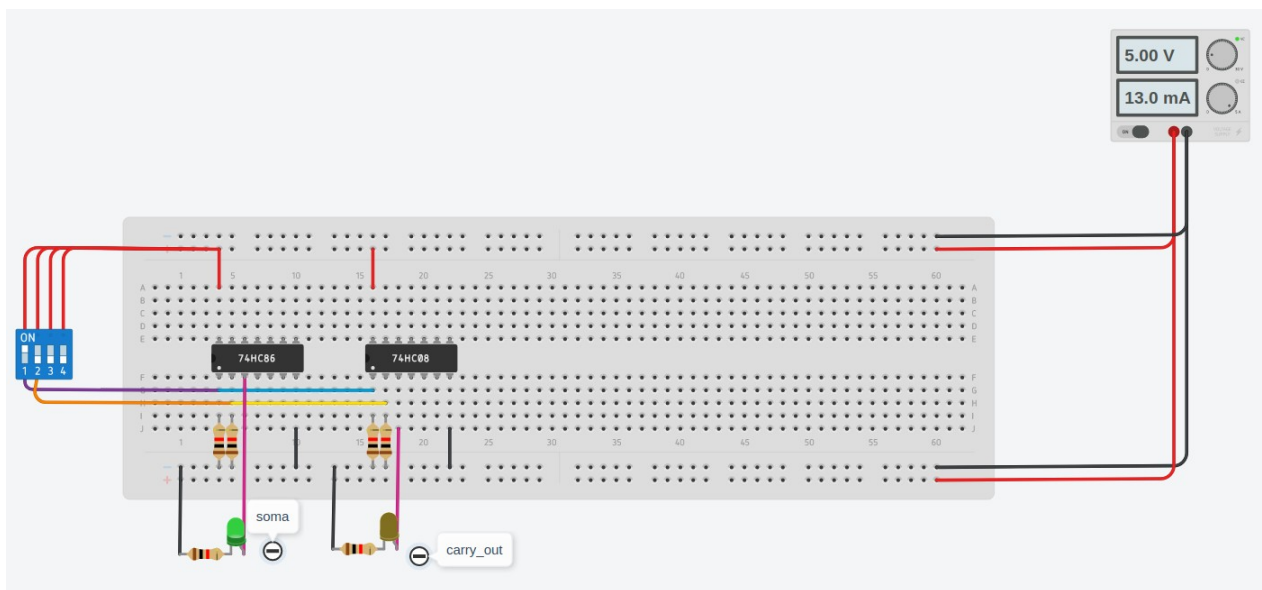
**Figura 06** : Decodificador hexadecimal no Logisim [1/2]



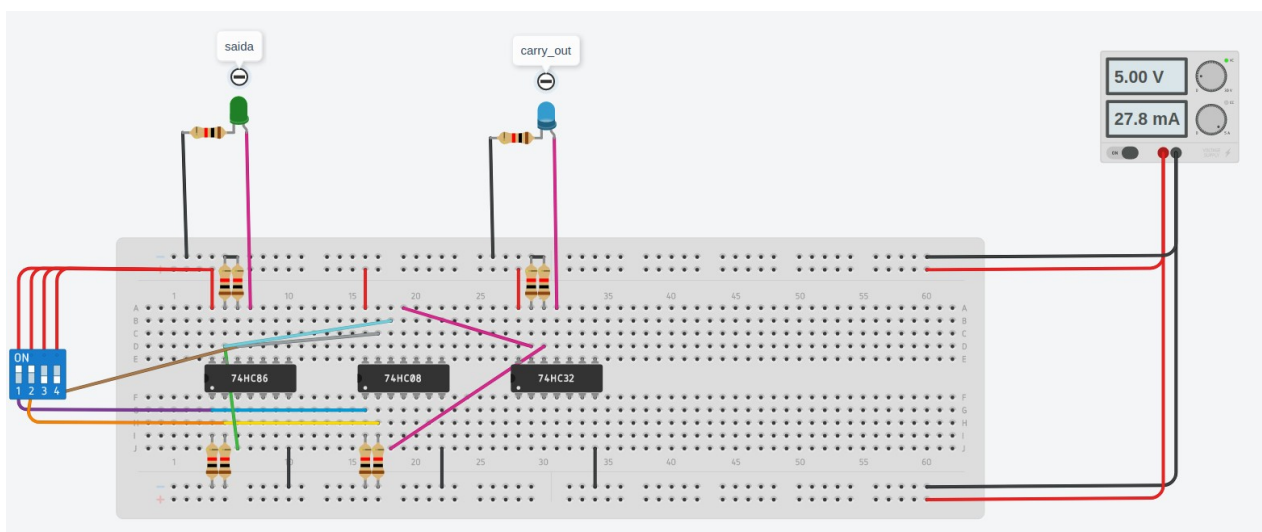
**Figura 07** : Decodificador hexadecimal no Logisim [2/2]



**Figura 08 :** Calculadora de 4 bits no Logisim



**Figura 09:**  $\frac{1}{2}$  somador no Tinkercad



**Figura 10:** Somador completo no Tinkercad

## Perguntas:

01.

Cada uma das portas lógicas possui uma tabela verdade pré-definida para cada entrada possível. Todavia, caso um dos terminais de uma porta lógica não estiver conectada a um valor válido (nem a 0 nem a 1), potencialmente, poderá haver um erro na saída, sendo ela imprevisível. Isso ocorre, uma vez que a porta lógica é sensível a pequenas interferências elétricas, assim, estas podem influenciar e levar a resultados inesperados.

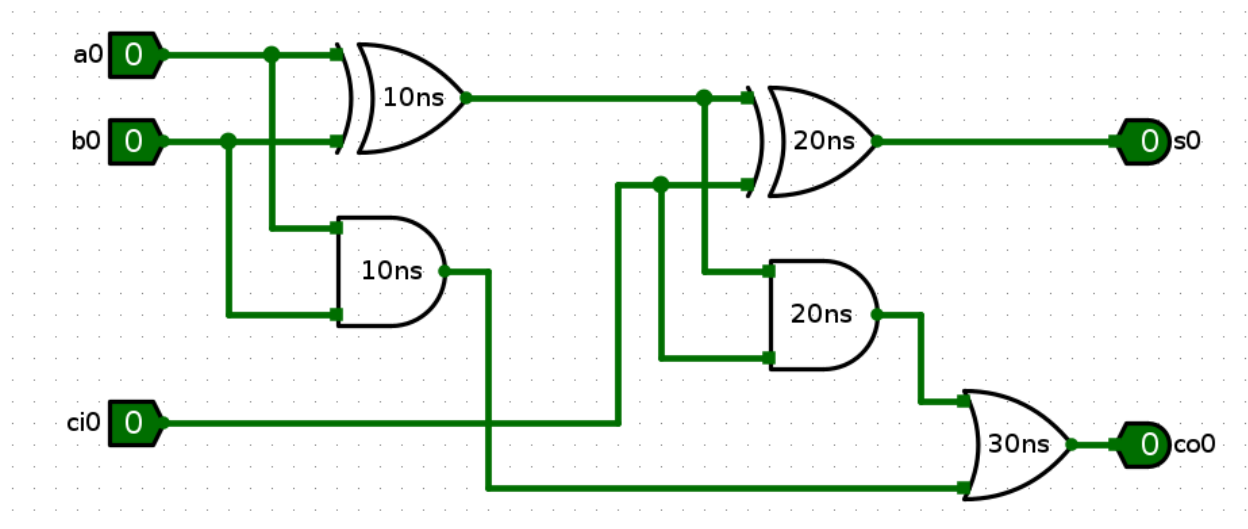
02.

Para somar os 4 bits, inicialmente somente alguns valores já são conhecidos:  $a_0$ ,  $a_1$ ,  $a_2$ ,  $a_3$ ,  $b_0$ ,  $b_1$ ,  $b_2$ ,  $b_3$  e  $ci_0$ . Estes são as 2 palavras de 4 bits e o carry in inicial. Para passar no primeiro somador dos primeiros bits, o tempo que leva é de 30ns. O mesmo poderia ocorrer em cada um dos outros 3 somadores completos. Todavia, para os seguintes, é necessário que o carry out da porta anterior envie o sinal para o carry in da atual. Em outras palavras: o  $ci_{n+1}$  depende do  $co_n$ .

Portanto, o somador não leva 30ns para somar tudo, pois é necessário o somador anterior finalizar para completar o próximo ligado a ele.

03.

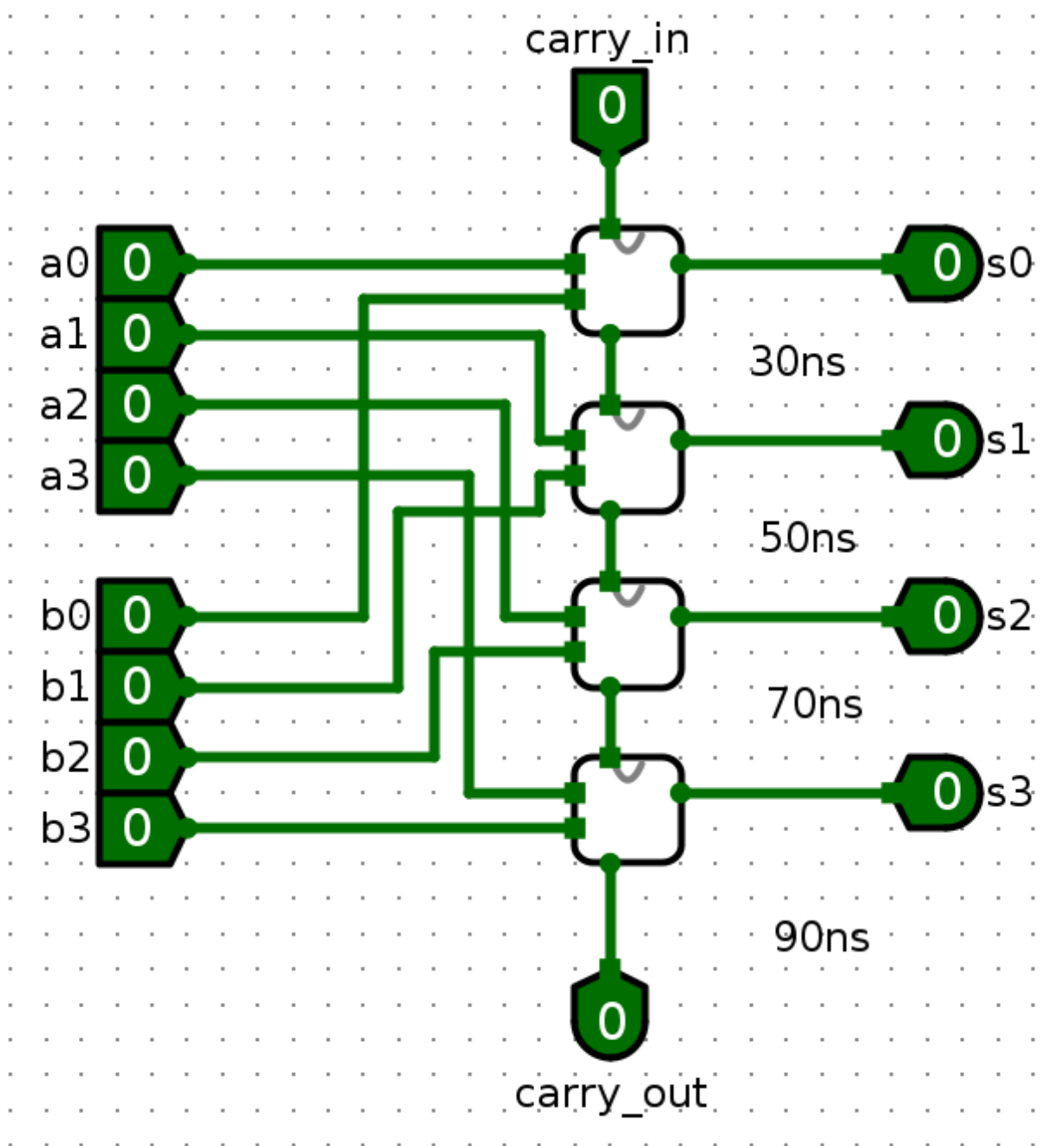
Tal como explicado no item anterior, o tempo total, tendo em vista o somador apresentado, pode ser calculado da seguinte forma:



No somador para 4 bits, o processo é semelhante. Todavia, nota-se que o  $co_0$  só esteve concluído após 30ns. Então, o sinal do  $co_1$  só estará disponível após este tempo.



Depois disso, precisará de mais 20ns para fornecer o sinal do  $\text{co}_2$ . Tal como na imagem abaixo:



Logo, para o somador de 4 bits acima, seriam necessários 90ns para que a soma fosse efetuada por completo.

**04.**

Seguindo o padrão do somador de 4 bits, seria necessário, para o de 32 bits, 32 somadores completos. Em outras palavras:

- $32 * 2 * \text{portas XOR} = 64 \text{ XOR}$
- $32 * 2 * \text{portas AND} = 64 \text{ AND}$
- $32 * 1 * \text{portas OR} = 32 \text{ OR}$

Além, claro, das 32 entradas e saídas, bem como o carry out e o overflow caso necessário.

A operação toda de soma, neste caso, ocorreria em 650ns:  $1 * 30\text{ns}$  para a primeira parte com  $co_0$  +  $31 * 20\text{ns}$  para os  $co_n$ .

**05.**

$$F = 1 / T$$

$$F = 1 / 650\text{ns}$$

$$F = 1 / 650 * 10^{-9} \text{ s}$$

$$F = 10^9 / 650 \text{ Hz}$$

$$F \approx 1.54 * 10^6 \text{ Hz}$$

$$F \approx 1.54 \text{ MHz}$$

Logo, a frequência de operação de um somador para 32 bits é de, aproximadamente, 1.54MHz.

**06.**

A partir dos itens anteriores, percebe-se uma significativa ineficácia deste somador completo para  $n$  bits. Por se tratar de operações lógicas com portas lógicas, uma possível solução para minimizar esta perda de tempo se encontra na possibilidade de simplificação do circuito, de tal sorte que não seja necessário todo o tempo para se obter o resultado esperado, utilizando técnicas de simplificação da tabela verdade e da sua equação.