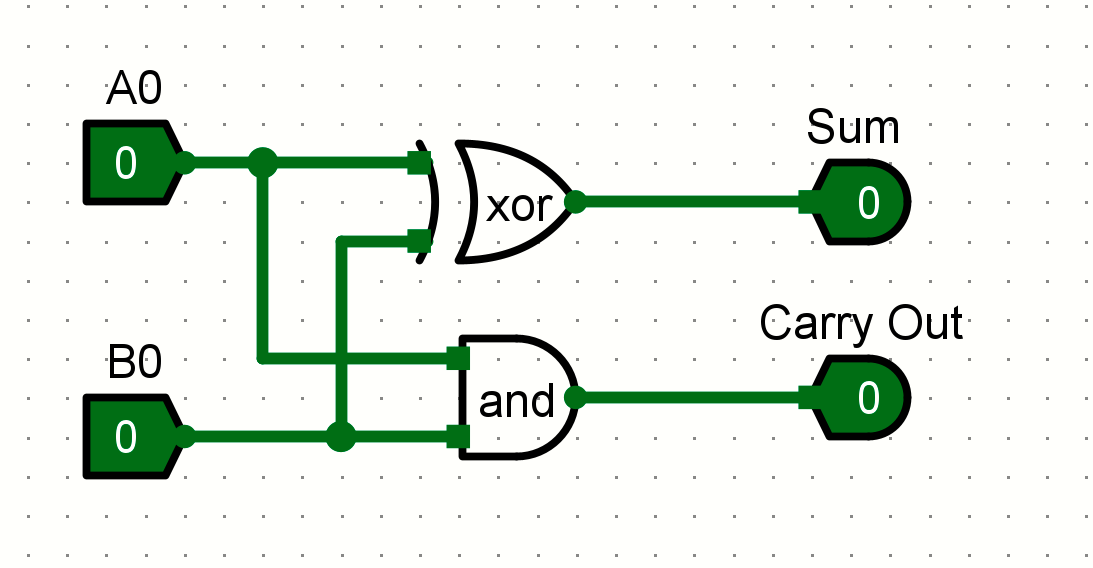
**Arquitetura de Computadores 2**

**Exercício Prático 01 – Prof. Romanelli**

**812839 - Vinícius Miranda de Araújo**

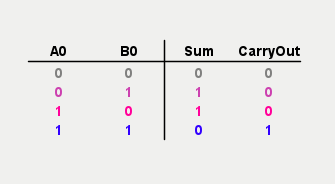
1. **½ (Meio) Somador no *Logisim*:**

Figura 1 - Meio Somador no Logisim



1. **Tabela Verdade do ½ somador:**

Figura 2 - Tabela Verdade do Meio Somador



1. **Componentes que possuem portas lógicas necessárias para a construção de um meio somador:**

Figura 3 - Porta XOR - 74LS86

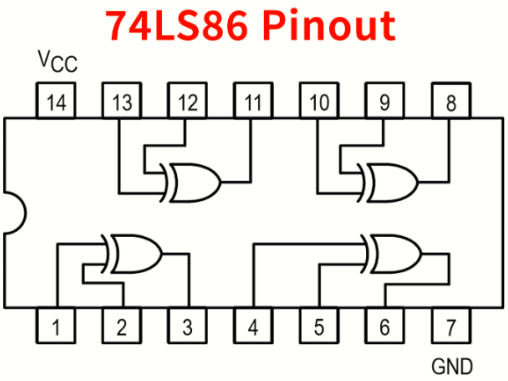
[](https://www.circuits-diy.com/74ls86-quad-2-input-exclusive-or-xor-gate-ic-datasheet/)

Figura 4 - Porta AND - 74LS08

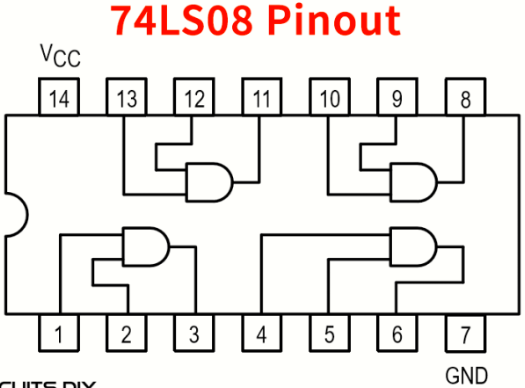
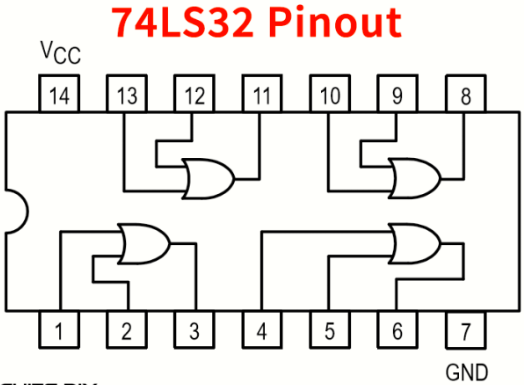
[](https://www.circuits-diy.com/74ls08-quadruple-two-input-and-gate-datasheet/)

Figura 5 - Porta OR - 74LS32

[](https://www.circuits-diy.com/74ls32-quad-2-input-or-logic-gate-ic-datasheet/)

1. **Pinos de alimentação (VCC e GND) e os pinos de entrada e saída de cada porta lógica:**

Figura 6 – Pinos Porta XOR

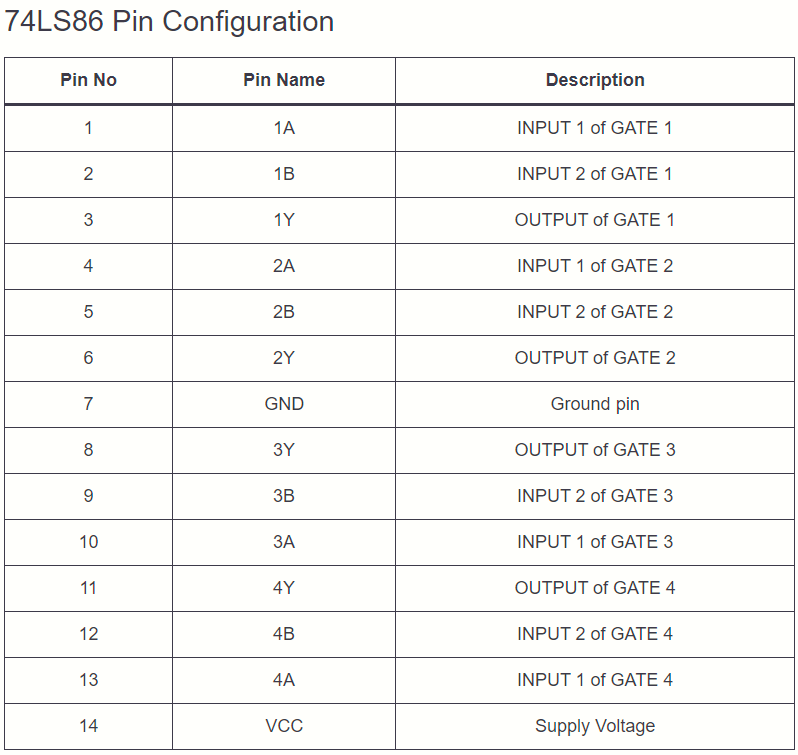
[](https://www.circuits-diy.com/74ls86-quad-2-input-exclusive-or-xor-gate-ic-datasheet/)

Figura 7 - Pinos Porta AND

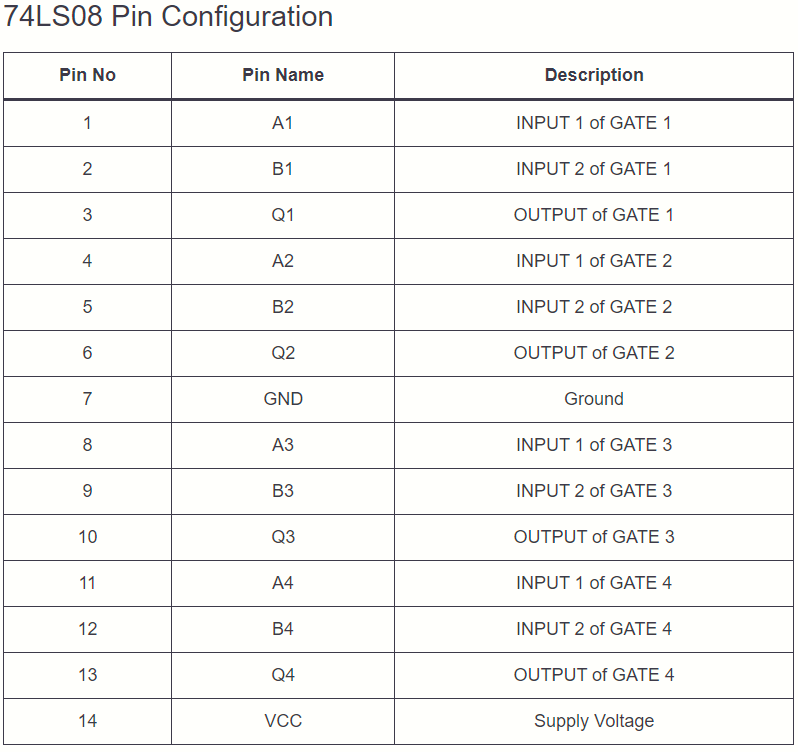
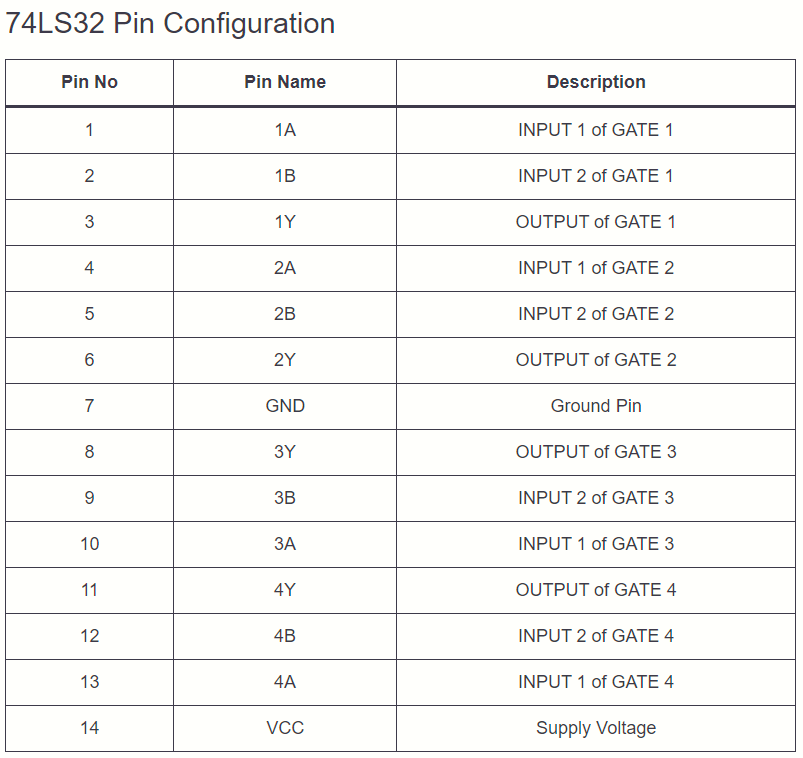
[](https://www.circuits-diy.com/74ls08-quadruple-two-input-and-gate-datasheet/)

Figura 8 - Pinos Porta OR

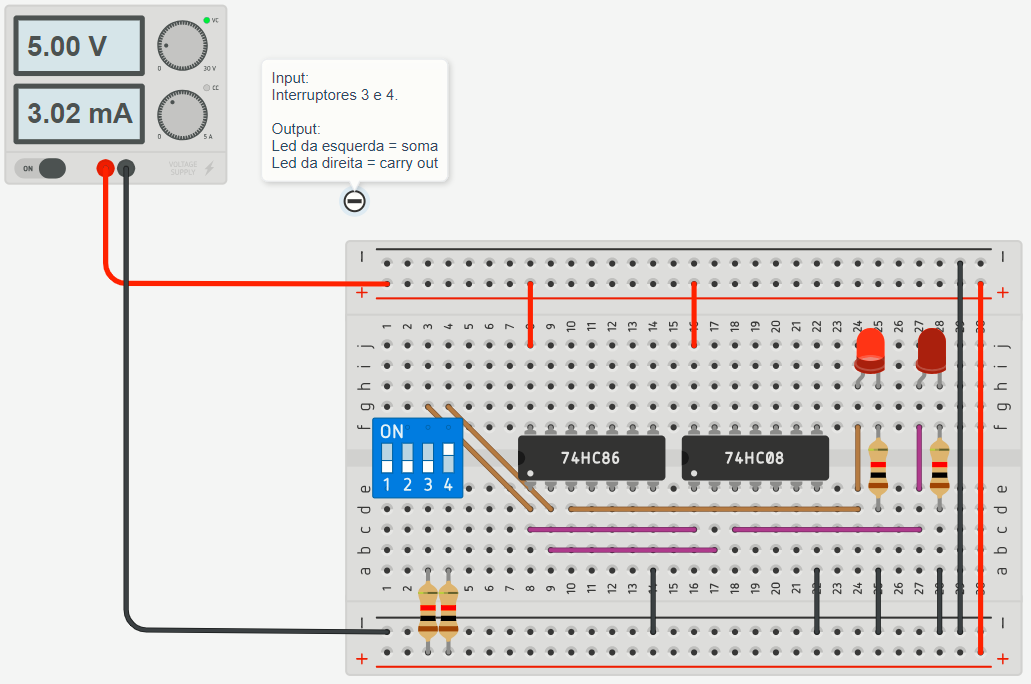
[](https://www.circuits-diy.com/74ls32-quad-2-input-or-logic-gate-ic-datasheet/)

**Pergunta 1:** O que acontece se um dos terminais de entrada de uma porta lógica não estiver conectado em 0 ou 1? (eletricamente ele deverá estar flutuando, ou seja, não  
conectado a nenhum nível lógico).

**R.:** Se uma das entradas de uma porta lógica estiver flutuando (não conectada a um nível lógico definido de 0 ou 1), isso pode resultar em comportamento imprevisível e instabilidade no circuito. Isso inclui saídas indeterminadas, interferência de ruídos externos, aumento do consumo de energia devido à fuga de corrente, e a possibilidade de falso *triggering*, onde a porta é ativada de forma não intencional.

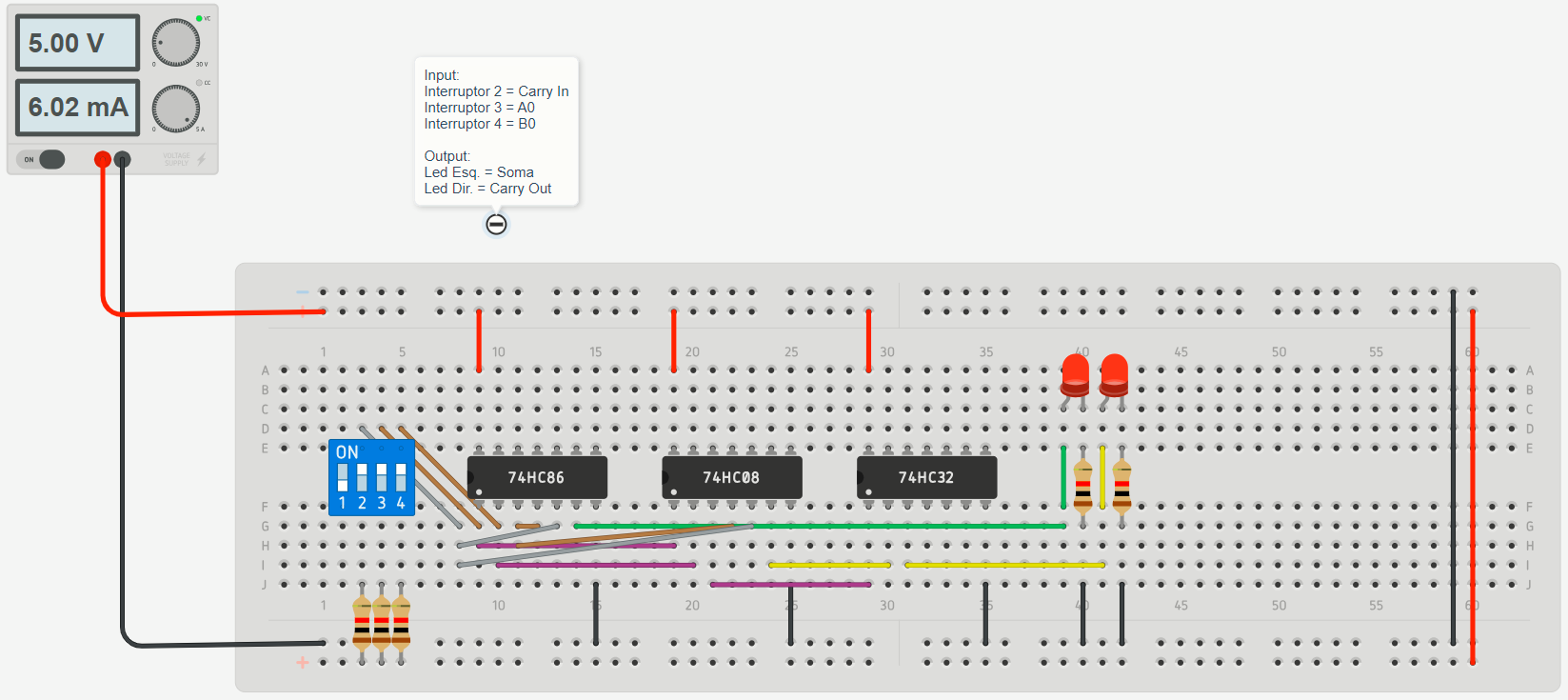
1. **½ Somador no *Tinkercad*:**

Figura 9 - Meio Somador Tinkercad

[](https://www.tinkercad.com/things/0opW8pel2bV-half-adder)

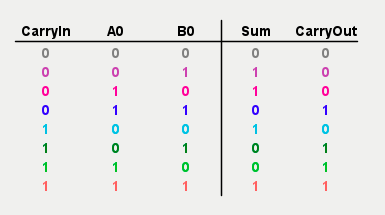
1. **Circuito somador completo de 1 bit no *Tinkercad*:**

Figura 10 - Somador Completo de 1 bit utilizando 2 meio-somadores

[](https://www.tinkercad.com/things/66UNbryKw90-full-adder-1-bit)

1. **Tabela Verdade do Somador completo de 1 bit:**

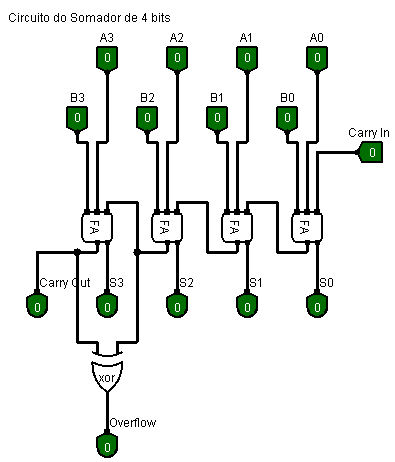
Figura 11 - Tabela Verdade do Somador Completo de 1 bit



1. **Explicação do funcionamento de um somador de 4 bits:**

Um somador de 4 bits é um circuito digital que realiza a adição de dois números binários de 4 bits. Ele é composto por quatro somadores completos (*full-adders*) conectados em série, onde cada somador adiciona um par de bits correspondente dos números de entrada junto com um bit de transporte (*carry*) proveniente do somador anterior. O primeiro somador utiliza um *carry-in* de zero, e cada somador subsequente utiliza o *carry-out* do somador anterior como seu *carry-in*. O resultado final é um número binário de 4 bits, junto com um bit de *carry-out* que representa um eventual excesso de capacidade, caso a adição resulte em um número que excede 4 bits.

Figura 12 - Somador de 4 bits



**Pergunta 2:** Qual o problema de tempo associado a esse tipo de somador (pense no carry)? Considere o atraso médio de cada porta lógica de 10 ns.

**R.:** O principal problema de tempo associado a um somador de 4 bits é a propagação do carry. Em um somador com vários bits, cada somador completo deve esperar o resultado do carry do somador anterior antes de poder completar a sua própria operação de adição. Esse atraso é cumulativo, o que significa que o tempo total necessário para completar a adição de 4 bits depende do número de bits e do tempo que cada carry leva para se propagar através dos somadores.

**Pergunta 3:** Qual o tempo necessário para a computação de uma soma e do vai um em um somador de 4 bits?

**R.:** Considerando um atraso médio de 10 nanosegundos (ns) por porta lógica, o tempo total necessário para a operação é de 90 ns. No início (tempo 0), o primeiro somador completo leva 30 ns para calcular a soma e o carry (vai-um). Os somadores subsequentes, já tendo calculado suas somas, apenas aguardam a propagação do carry anterior, resultando em um atraso adicional de 20 ns por somador. Portanto, o tempo médio necessário, Tmed, é calculado a partir da quantidade total de bits, *n*, e dado pela fórmula:

**Tmed = 30 ns + 20 ns \* (n-1)**

**Pergunta 4:** O que seria necessário para um somador de 32 bits?

**R.:** Para um somador de 32 bits são necessários 32 somadores de 1 bit.

**Pergunta 5:** Considerando esses tempos acima, calcule a frequência de operação de um somador de 32 bits.

**R.:** Dada a fórmula abstraída na questão 3, tem-se:

Tmed = 30 ns + 20 ns \* 32-1 **=>** Tmed = 30 ns + 620 ns **=>** Tmed = 650 ns

**Pergunta 6:** Você consegue propor alguma forma de tornar essa soma mais veloz?

**R.:** Uma forma de tornar o somador mais rápido é eliminar a dependência da propagação do *carry* entre os estágios. No somador *Carry-Lookahead Adder (CLA)*, cada estágio calcula diretamente seu próprio *carry* de entrada com base nas entradas iniciais (*bits e carry-in*), sem esperar o *carry* propagado dos estágios anteriores. Dessa forma, o somador consegue determinar rapidamente os *carries* para todos os bits simultaneamente, o que reduz significativamente o tempo de propagação e acelera o processo de adição.

**Calculadora de 4 bits (*Logisim*):**

Figura 13 – Circuito de uma calculadora de 4 bits com display hexadecimal

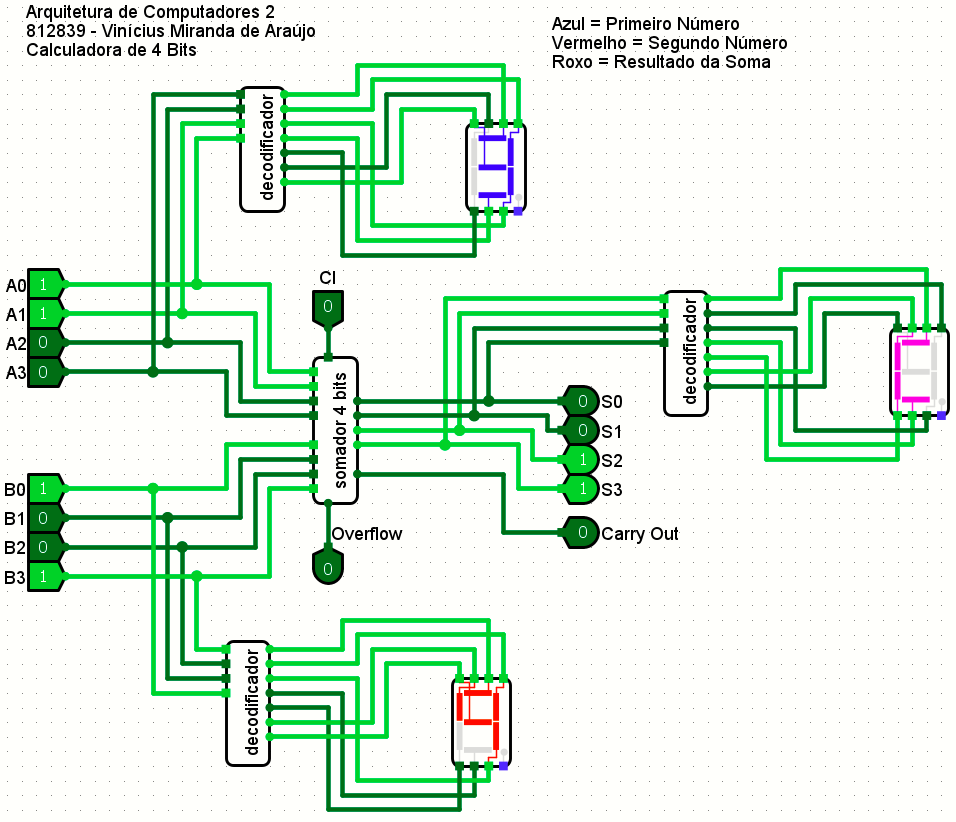


Figura 14 - Circuito de Meia Soma

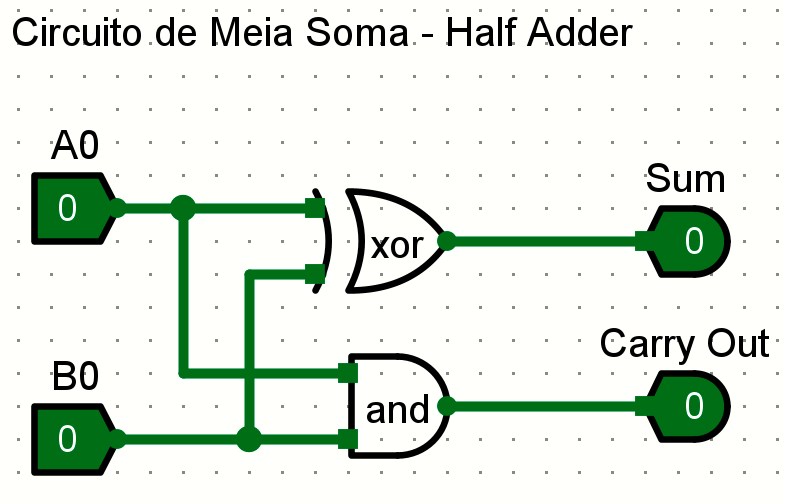


Figura 15 - Circuito de Soma Completa

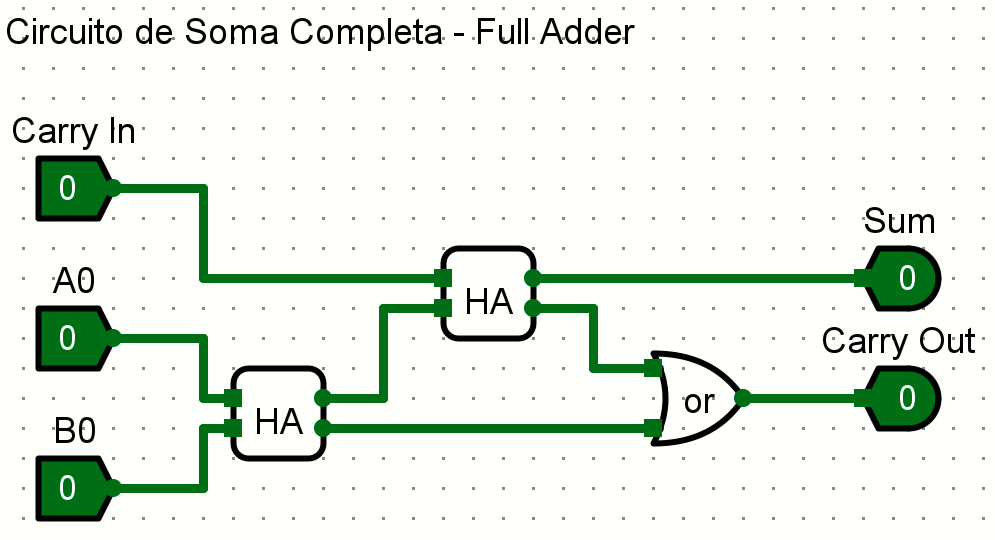


Figura 16 - Circuito de Somador de 4 bits

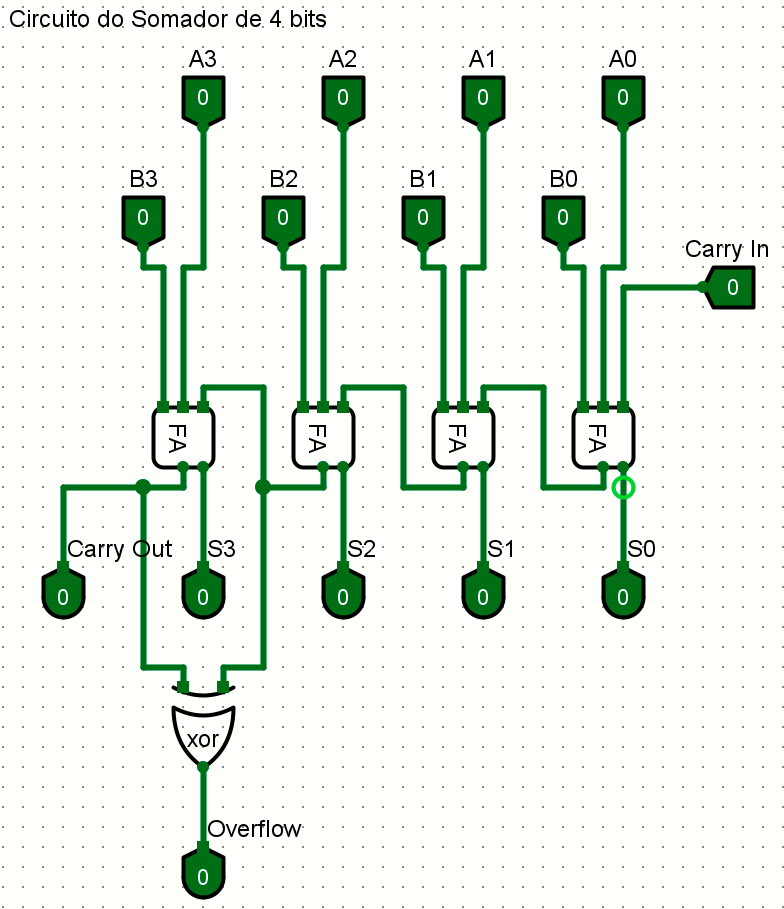
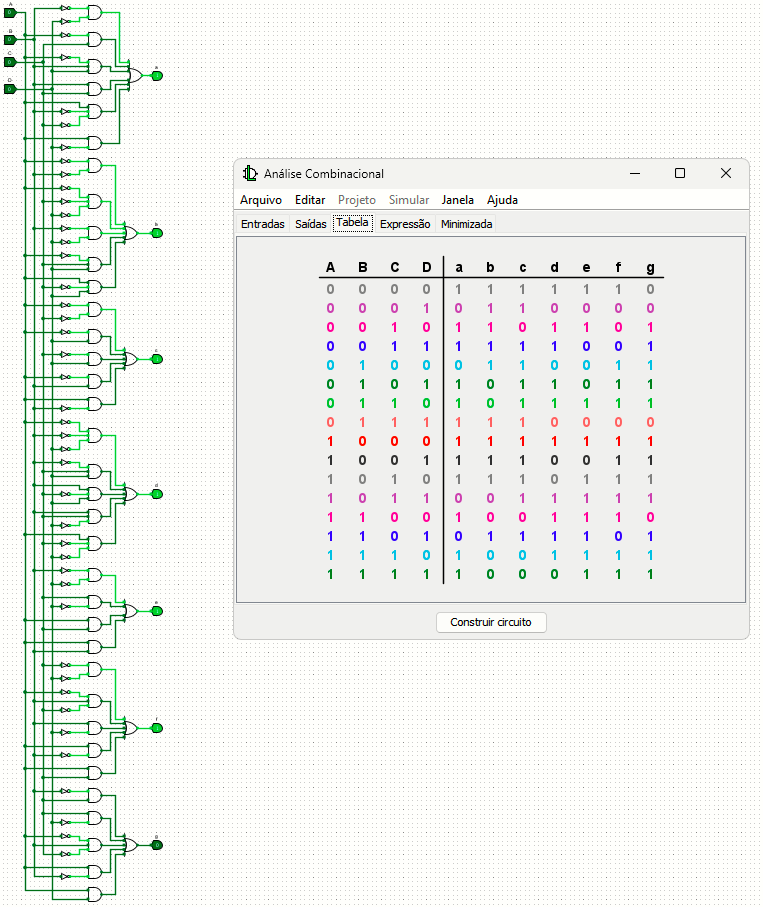


Figura 17 - Circuito de Decodificação

**FIM**