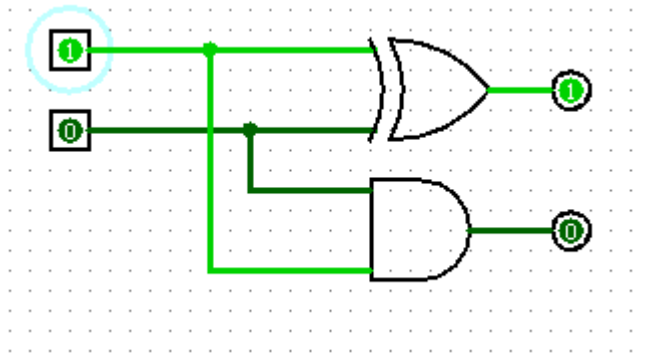
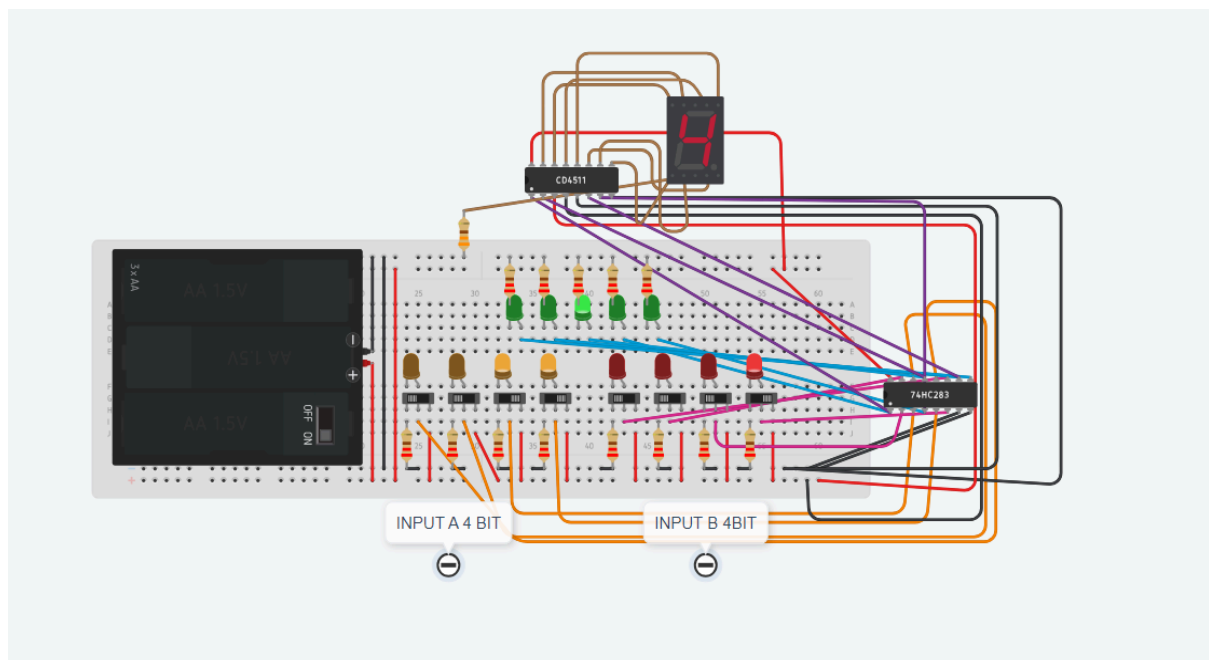


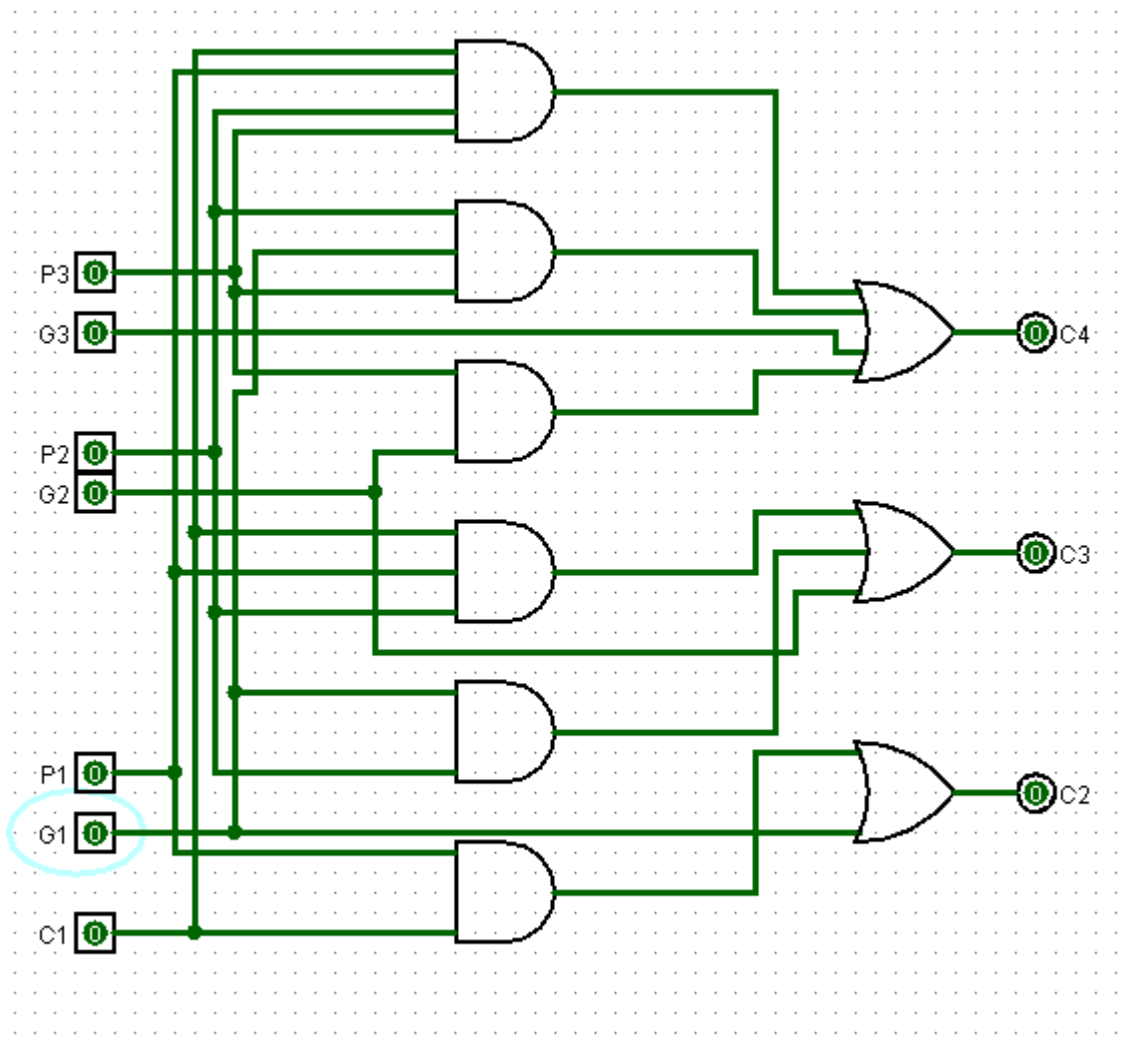
- 1- Se um dos terminais de entrada de uma porta lógica não estiver conectado a 0 ou 1, ele estará "flutuando", essa condição pode causar um comportamento imprevisível, pois a entrada flutuante pode ser lida como 0 ou 1.
- 2- O problema de tempo associado a este tipo de somador é o "propagation delay" do carry. Como o carry deve ser propagado através de todos os somadores, cada estágio depende do cálculo do estágio anterior, o que introduz um atraso cumulativo.
- 3- O tempo estimado é de 160ns para o carry out do último bit pois o somador de 4 bits é composto por quatro somadores completos de 1 bit, cada um com 40ns de delay.
- 4- Para construir um somador 32 bit utilizando o mesmo princípio seria necessário encadear 32 somadores de 1 bit.

### Meio somador feito no logisim



### Circuito somador 4 bit realizando soma de 3 com 1





## Conversa com o chat GPT :

**Pergunta:** Como criar multiplicador no logisim?

**Resposta:** Para criar um multiplicador no Logisim, você pode seguir os passos abaixo. Este exemplo usa um multiplicador de 2 bits, mas o conceito pode ser expandido para números de mais bits.

### Passos para Criar um Multiplicador de 2 Bits

1. **Abra o Logisim:** Inicie o Logisim e abra um novo projeto.
2. **Criando as Entradas:**
  - Adicione quatro entradas (Input) para representar os dois números binários de 2 bits. Cada número terá duas entradas.
  - Nomeie as entradas como **A0**, **A1** (para o primeiro número) e **B0**, **B1** (para o segundo número).
3. **Criando as Saídas:**
  - Adicione quatro saídas (Output) para o produto, que será um número binário de 4 bits. Nomeie as saídas como **P0**, **P1**, **P2**, **P3**.
4. **Implementação do Produto:**

- **Passo 1:** Realize as operações AND entre os bits correspondentes. No Logisim, use o componente AND para isso.
    - **AND1:** A0 AND B0 (conecte a saída à P0).
    - **AND2:** A1 AND B0.
    - **AND3:** A0 AND B1.
    - **AND4:** A1 AND B1.
  - **Passo 2:** Agora, some os resultados das operações AND usando portas XOR e AND para gerar os bits intermediários e as somas.
    - **XOR1:** XOR do resultado do **AND2** com o resultado do **AND3**. A saída deste XOR será P1.
    - **AND5:** AND do resultado do **AND2** com o resultado do **AND3** para obter o carry.
  - **Passo 3:** Some o resultado de **AND5** com **AND4** usando outra porta XOR e AND.
    - **XOR2:** XOR de **AND5** com **AND4**. Conecte a saída a P2.
    - **AND6:** AND de **AND5** com **AND4** para gerar o carry final, que será P3.
5. **Conectar as Saídas:**
- Conecte as saídas de acordo com as operações realizadas para formar o número final de 4 bits que representa o produto.

## Considerações Finais

Este é um exemplo básico para um multiplicador de 2 bits. Para multiplicadores de mais bits, você precisará expandir o circuito, aumentando o número de portas AND, somadores (XOR e AND) e ajustando as conexões de acordo com o algoritmo de multiplicação binária.

Se precisar de mais ajuda ou de um circuito específico, posso criar um exemplo mais detalhado ou te guiar para um multiplicador de mais bits!

