



**UNIVERSIDADE FEDERAL DE RORAIMA**  
**CENTRO DE CIÊNCIA E TECNOLOGIA**  
**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**



**DCC301– ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES– 2025**

**PROF. DR. HEBERT OLIVEIRA ROCHA**

**HELIAN VINCIUS FILINTO DA SILVA**

**SALVADOR DE JESUS MALAVÉ**

**LUCAS GUILHERME PEREIRA SANTIAGO**

**LABORATÓRIO DE CIRCUITOS**

**BOA VISTA,**

**RR 2025**

**HELIAN VINCIUS FILINTO DA SILVA**  
**SALVADOR DE JESUS MALAVÉ**  
**LUCAS GUILHERME PEREIRA SANTIAGO**

**LABORATÓRIO DE CIRCUITOS**

Trabalho da disciplina de Arquitetura e Organização de Computadores do ano de 2025 apresentado à Universidade Federal de Roraima do curso de Bacharelado em ciência da computação.

Docente: Prof. Dr. Hebert O. Rocha

## SUMÁRIO

1.	INTRODUÇÃO .....	5
2.	COMPONENTES.....	6
2.1.	Registrador Flip-Flop do tipo JK e do tipo D .....	6
2.1.1	Flip-Flop JK Mestre Escravo .....	6
2.1.2	Registrador Flip-Flop Tipo D.....	7
2.2.	3 Multiplexador de 4 entradas.....	7
2.3.	Porta lógica XOR .....	8
2.4.	Somador de 8 bits mais 4 .....	8
2.5.	Memória ROM de 8 bits .....	9
2.6.	Memória RAM de 8 bits .....	10
2.7.	Banco de Registradores de 8 bits .....	11
2.8.	Somador de 8 bits.....	12
2.9.	Detector de Sequência Binária 101 .....	12
2.10.	Unidade Lógica Aritmética de 8 Bits .....	13
2.11.	Extensor de sinal de 4 bits para 8 bits .....	15
2.12.	Máquina de Estados .....	15
2.13.	Contador Síncrono.....	16
2.14.	Detector de Paridade Ímpar.....	17
2.15.	Problema Otimização Logica.....	18
2.16.	Decodificador de 7 Segmentos.....	19
2.17.	Detector de Número Primo de 4 entradas com Mapa de Karnaugh.....	20
3.	REFERÊNCIAS .....	21

## LISTA DE FIGURAS

Flip-Flop JK Mestre Escravo .....	6
Figura 2 - Flip-Flop D.....	7
Figura 3 - Flip-Flop JK.....	7
Figura 4 - Multiplexador de 4 entradas .....	8
Figura 5 - Porta Lógica XOR .....	8
Figura 6 - Somador Completo .....	9
Figura 7 - Componente do Somador .....	9
Figura 8 - Memória ROM.....	10
Figura 9 - Memória RAM .....	11
Figura 10 - Banco de Registradores.....	12
Figura 11 - Detector de Sequência Binária.....	13
Figura 12 - Tabela de Comandos.....	13
Figura 13 - Unidade Lógica Aritmética .....	14
Figura 14 - Extensor de sinal de 4 bits para 8 bits.....	15
Figura 15 - Máquina de Estados .....	15
Figura 16 - Contador Síncrono .....	16
Figura 17 - Detector de Paridade Ímpa .....	17
Figura 18 - Circuito Antes de ser Simplificado .....	18
Figura 19 - Circuito Após Ser Simplificado .....	18
Figura 20 - Decodificador de 7 Segmentos .....	19
Figura 21 - Componente do Decodificador de 7 Segmentos.....	19
Figura 22 - Detector de Número Primo.....	20

## **1.INTRODUÇÃO**

Este relatório técnico descreve o desenvolvimento e a implementação de diversos circuitos digitais, totalizando 17 componentes, no software Logisim. Cada componente foi meticulosamente construído a partir de portas lógicas, constantes, distribuidores e pinos de entrada e saída.

O objetivo deste trabalho é demonstrar a aplicação prática de conhecimentos em eletrônica digital, documentando a lógica de design, o processo de construção e os testes realizados para validar a funcionalidade de cada circuito. Todos os componentes foram criados e simulados no Logisim versão 2.7.1, uma ferramenta padrão tanto em ambientes acadêmicos quanto profissionais.

## 2.COMPONENTES

### 2.1. Registrador Flip-Flop do tipo JK e do tipo D

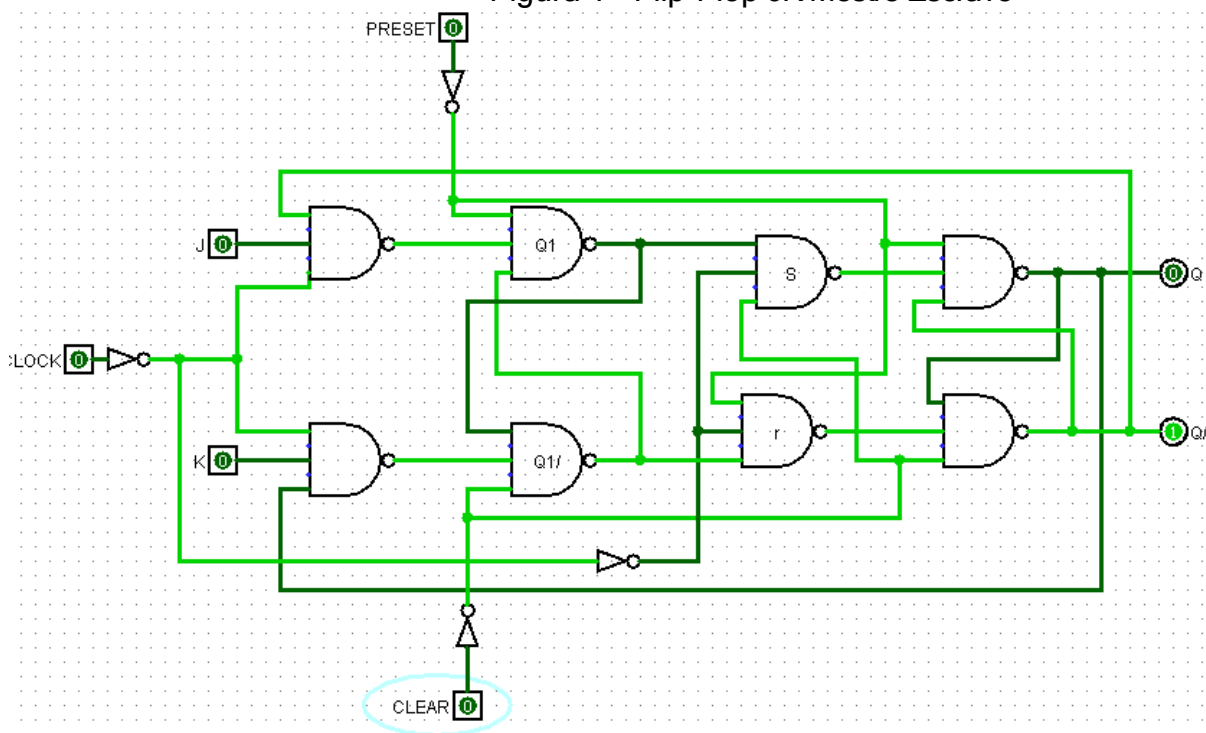
Um flip-flop é um circuito biestável capaz de armazenar um único bit de informação. Ele possui duas saídas estáveis, que representam os estados lógicos 0 e 1. A transição entre esses estados é controlada por sinais de entrada, como o clock e os dados, permitindo que o flip-flop mantenha um valor até ser instruído a alterá-lo.

#### 2.1.1 Flip-Flop JK Mestre Escravo

No circuito, o flip-flop de trás funciona como o mestre, com entradas J e K, enquanto o flip-flop da frente atua como o escravo, com entradas S e R. Quando o Clock está em nível alto (1), seu sinal é invertido ao passar pelas portas NAND, resultando em 0 na saída, o que bloqueia as entradas S e R do flip-flop escravo, permitindo que o mestre processe os sinais de J e K.

Quando o Clock retorna a 0, as saídas do mestre ficam bloqueadas e as entradas S e R do escravo são liberadas, permitindo que ele atualize seu estado. Dessa forma, o circuito só altera seu estado na transição de descida do Clock, caracterizando o funcionamento mestre-escravo usando portas NAND.

Figura 1 - Flip-Flop JK Mestre Escravo

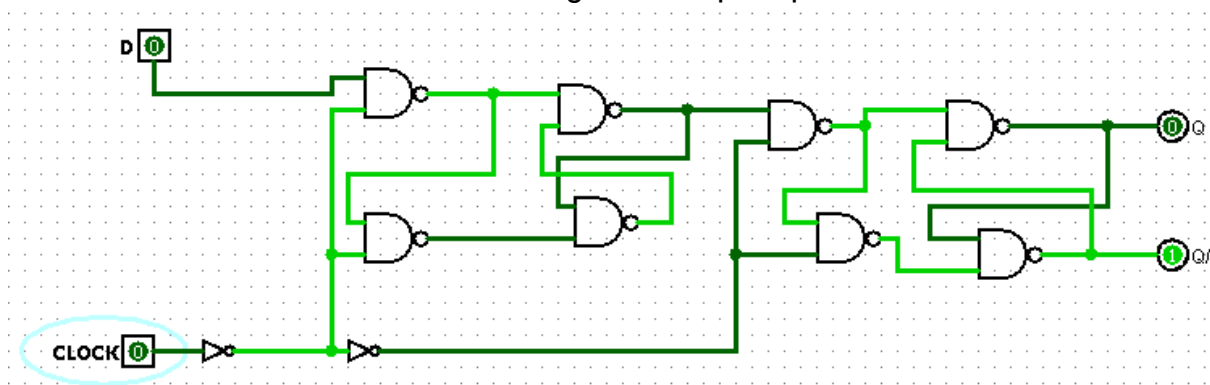


### 2.1.2 Registrador Flip-Flop Tipo D

O flip-flop D é uma versão derivada do flip-flop JK mestre-escravo, em que a entrada K é sempre invertida em relação à entrada J, garantindo que o circuito só possa assumir os estados 0 ou 1, eliminando combinações indesejadas. Ele possui uma única entrada de dados, chamada D, e os valores aplicados nela são transferidos para a saída Q somente quando o Clock é ativado, tornando o armazenamento de informação síncrono com o sinal de relógio.

Dessa forma, o flip-flop D atua como um elemento de memória controlado pelo Clock, armazenando o valor presente na entrada no momento da transição de ativação e mantendo esse valor até a próxima atualização, garantindo estabilidade e previsibilidade no circuito digital.

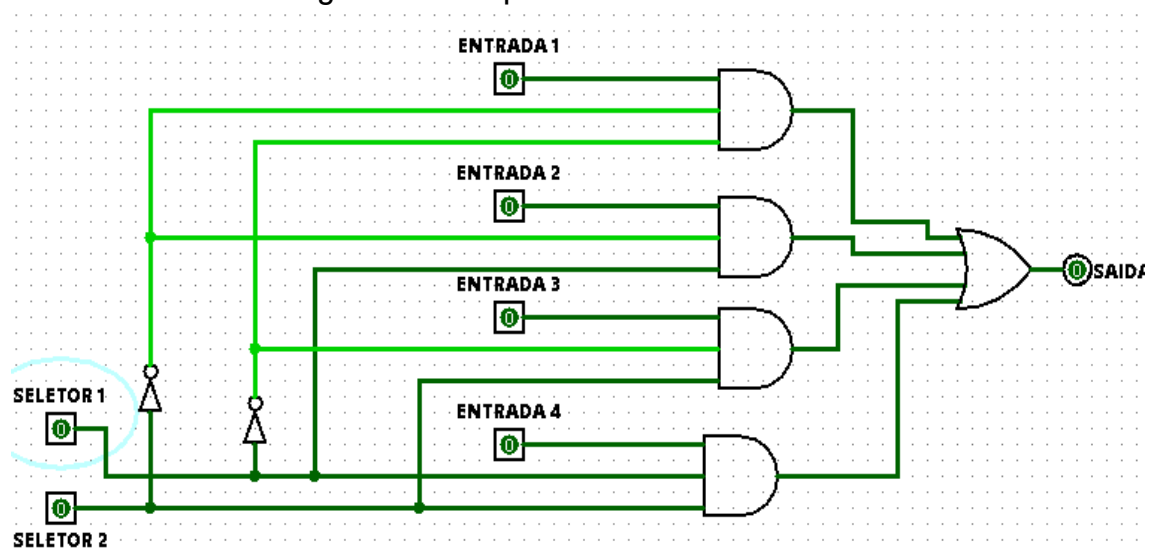
Figura 2 - Flip-Flop D



### 2.2. 3 Multiplexador de 4 entradas

Um multiplexador, ou MUX, é um circuito lógico combinacional que permite escolher uma entre várias entradas de dados e direcioná-la para uma única saída, funcionando como um seletor eletrônico. Ele utiliza entradas de seleção para determinar qual linha de entrada será encaminhada à saída em cada instante. Por exemplo, o MUX 4:1 pedido possui quatro entradas de dados, duas linhas de seleção que definem qual entrada será passada adiante e uma saída.

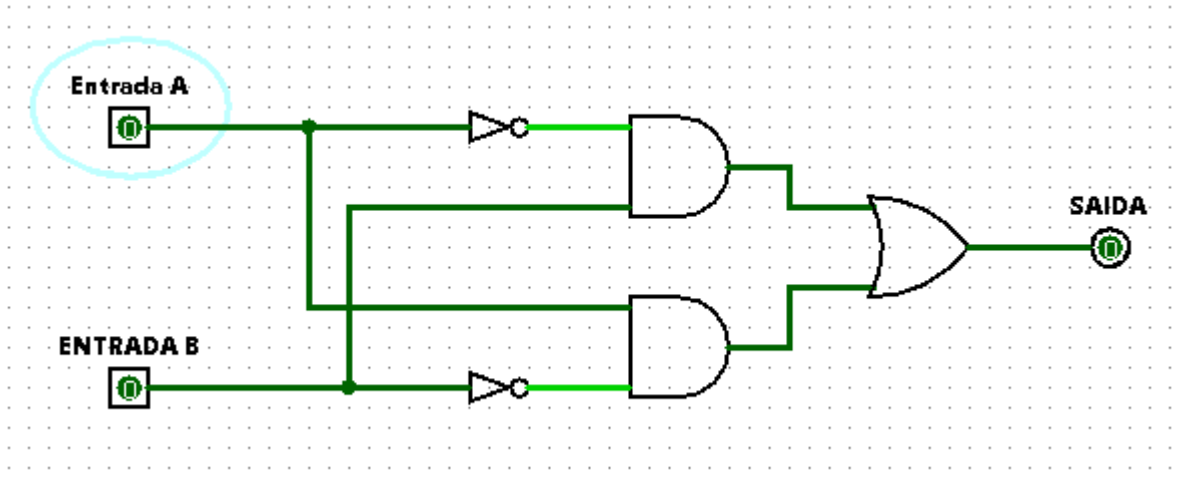
Figura 3 - Multiplexador de 4 entradas



### 2.3. Porta lógica XOR

A porta XOR (do inglês Exclusive OR, ou “OU exclusivo”) é uma porta lógica que só retorna 1 quando as entradas são diferentes. Ou seja, se as duas entradas forem iguais a saída é 0.

Figura 4 - Porta lógica XOR



### 2.4. Somador de 8 bits mais 4

O somador é composto por duas partes principais: o somador completo e o somador de 8 bits. O somador completo recebe três entradas: A e B, que são os bits que serão somados, e o Carry In, que representa o valor de vai-um vindo do bit anterior. A partir dessas entradas, ele calcula a soma e produz duas saídas: S, que é o resultado da soma do bit atual, e o Carry Out, que será levado para a soma do próximo bit.

O somador de 8 bits utiliza oito somadores completos em sequência, recebe números de 8 bits como entrada e gera uma saída de 9 bits, sendo o último bit o overflow, indicando que ocorreu um transbordo na soma.

Figura 5 - Somador Completo

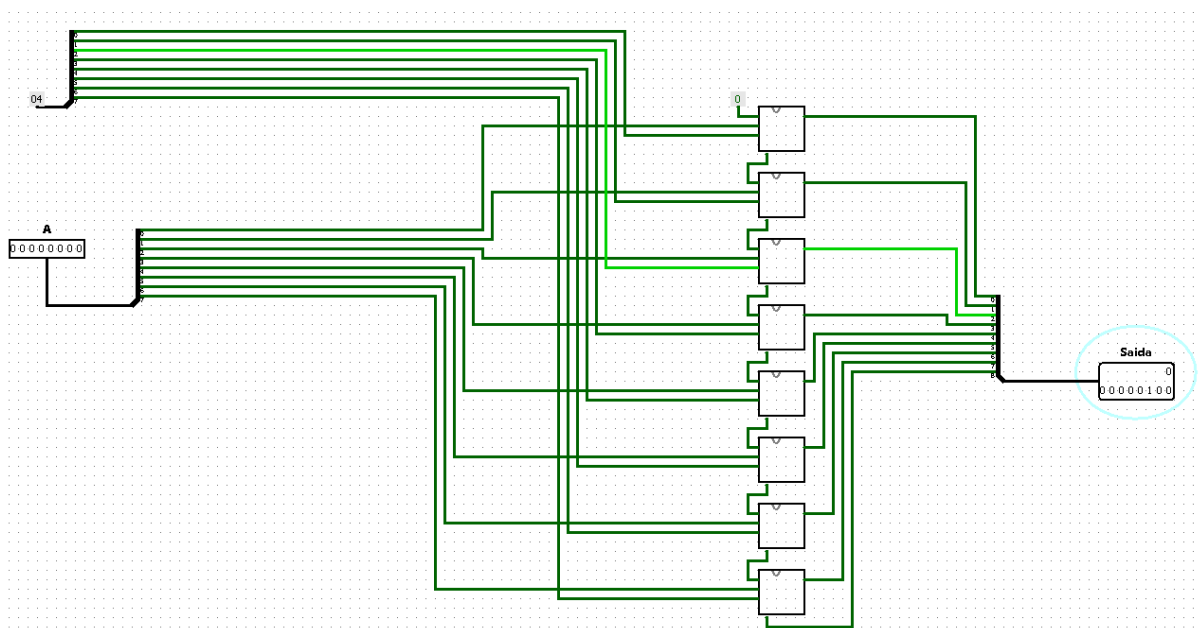
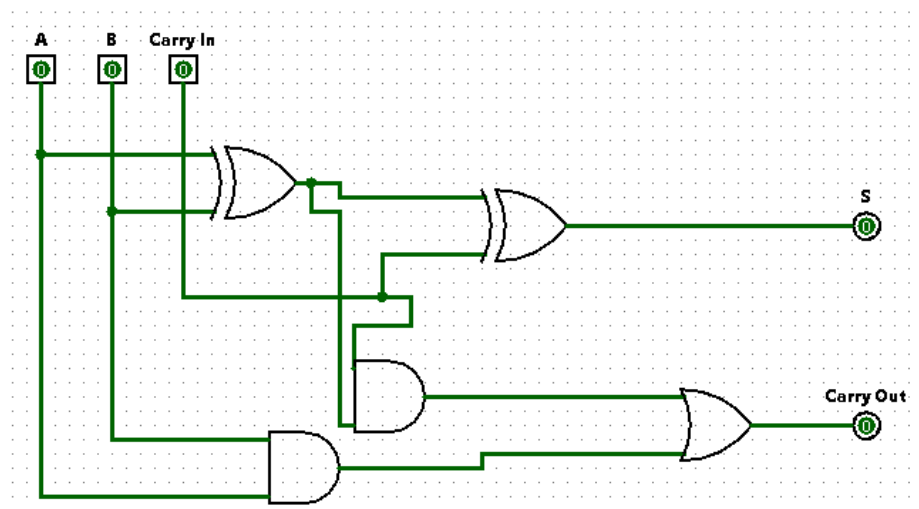




Figura 6 - Componente do Somador



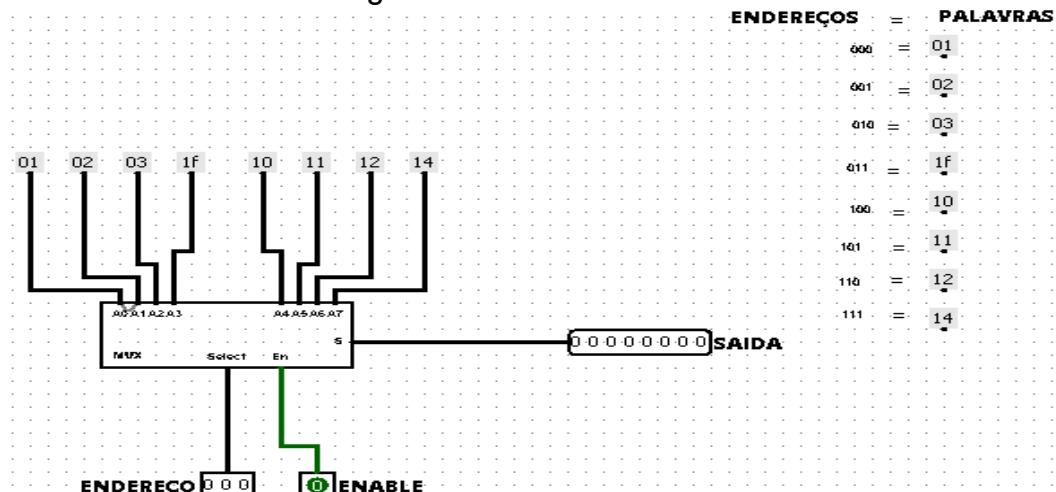
## 2.5. Memória ROM de 8 bits

A memória ROM (Read-Only Memory), como o próprio nome sugere, é um tipo de memória projetado para a leitura de dados. Sua principal característica é a não-volatilidade, ou seja, as informações armazenadas permanecem intactas mesmo após o desligamento do sistema.

Nessa implementação no Logisim, 8 constantes de 8 bits representam os dados permanentes. O acesso a essas "palavras" de dados é controlado por um multiplexador 8x1 de 8 bits, que seleciona uma delas por vez. A escolha da palavra é feita por um endereço, que neste circuito é uma entrada de 3 bits.

A leitura dos dados é controlada por uma entrada Enable. Se o valor dessa entrada for 0, a leitura é desativada, e nenhuma informação é exibida. Por outro lado, se o valor for 1, a leitura ocorre normalmente, e a palavra selecionada pelo endereço é exibida na saída de 8 bits do multiplexador.

Figura 7 - Memória ROM



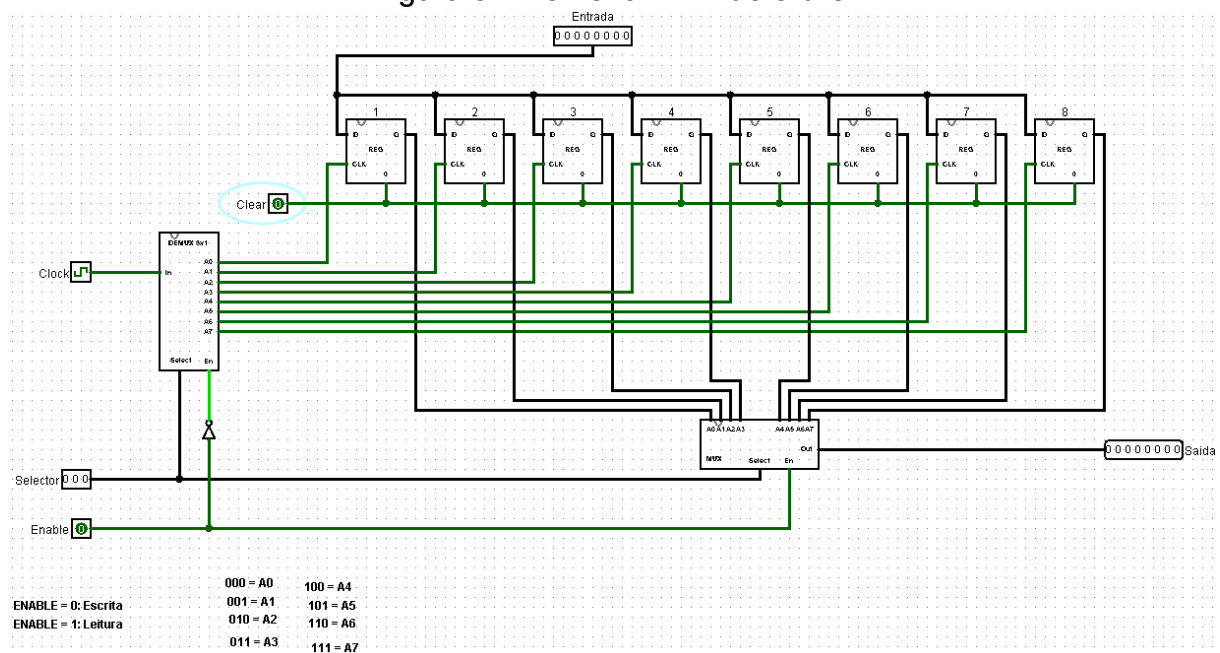
## 2.6. Memória RAM de 8 bits

A memória RAM (Random Access Memory) funciona como o principal espaço de trabalho temporário de um computador, permitindo que o processador acesse dados rapidamente, independentemente de sua localização. Sua estrutura é baseada em registradores paralelos, que são construídos a partir de flip-flops do tipo D, componentes essenciais para o armazenamento de bits.

A gravação de dados na RAM é controlada por um demultiplexador. Ele age como um seletor, direcionando o sinal de clock para o registrador específico que deve receber a informação, com base no endereço fornecido pela entrada Selector. Ao mesmo tempo, a leitura é feita por um multiplexador que escolhe qual registrador será lido para a saída, também guiado pela entrada Selector.

É crucial notar que as operações de leitura e escrita não podem ocorrer simultaneamente. Essa funcionalidade é gerenciada pela entrada Enable. Quando o Enable está em nível lógico 1, a leitura é ativada e a escrita é desativada. Se o Enable estiver em 0, a escrita é ativada e a leitura é desativada, garantindo que as operações não entrem em conflito.

Figura 8 - Memória RAM de 8 bits



## 2.7. Banco de Registradores de 8 bits

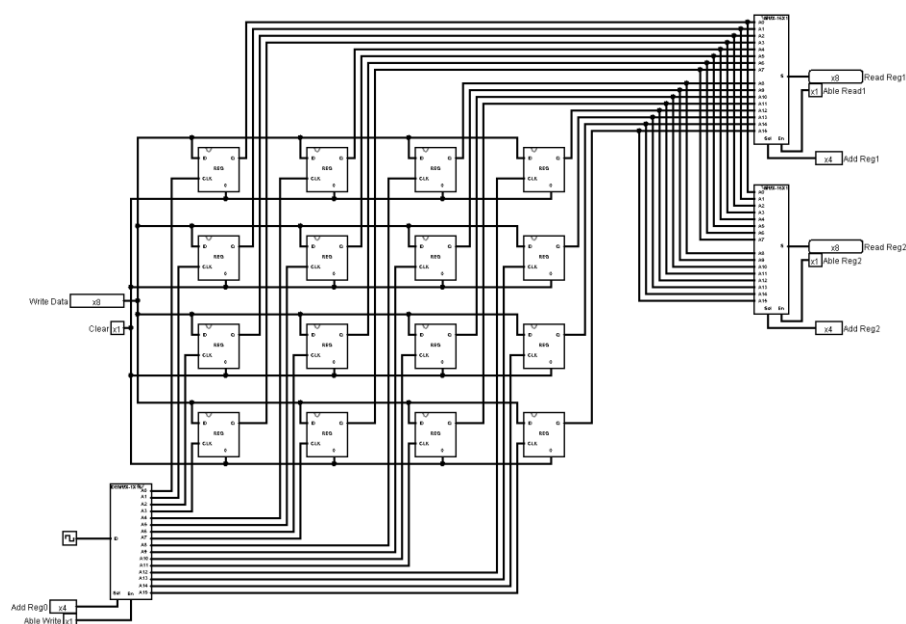
O banco de registradores é um componente central em processadores, atuando como um armazenamento temporário de alta velocidade para os dados em uso. Sua arquitetura é otimizada para permitir acesso e manipulação de informações de forma ágil, facilitando operações de leitura e escrita.

Este banco é composto por 16 registradores, cada um capaz de armazenar 8 bits de informação. Para gerenciar essas operações, o circuito utiliza um demultiplexador 16x1, que direciona o pulso de clock exclusivamente para o registrador que receberá a escrita. Além disso, dois multiplexadores são responsáveis por selecionar e ler os dados armazenados. O banco possui oito entradas no total, sendo elas:

- Uma entrada de 8 bits para o dado a ser gravado.
- Três seletores de 4 bits, que definem o endereço do registrador para leitura ou escrita.
- Três sinais de habilitação de 1 bit, que ativam ou desativam as operações.
- Um sinal de limpeza de 1 bit, que apaga o conteúdo dos registradores.

É importante notar que, sem os sinais de habilitação ativados, nenhuma operação pode ser executada. A inatividade das entradas de habilitação impede tanto a leitura quanto a escrita de dados, mesmo com o pulso de clock. Uma vez que os sinais de habilitação estão ativos, o circuito permite que as operações de escrita e leitura ocorram normalmente. Vale ressaltar que os sinais de habilitação operam de forma independente, o que significa que é possível realizar apenas a leitura ou apenas a escrita de dados, sem que uma operação interfira na outra.

Figura 9 - Banco de Registradores

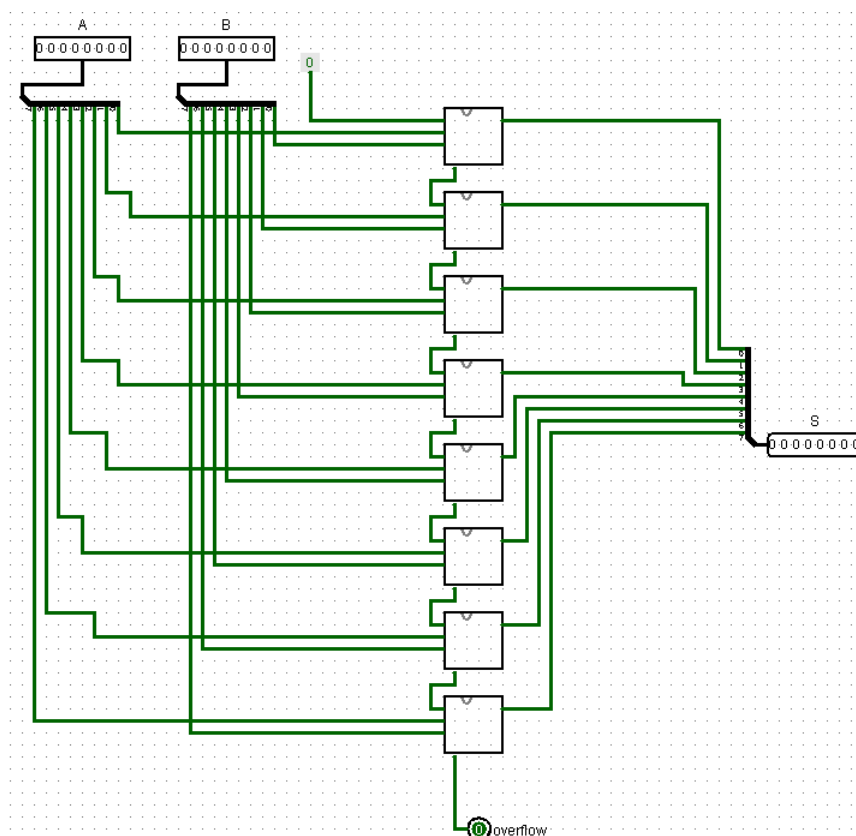


## 2.8. Somador de 8 bits

O somador de 8 bits funciona de forma similar ao somador de 8 bits com 4, mas com a capacidade de somar dois números completos de 8 bits. Para isso, ele utiliza oito somadores de 1 bit em paralelo, onde cada um lida com um par de bits correspondentes de cada um dos dois números de entrada.

O circuito gera duas saídas: uma de 8 bits, que mostra o resultado da soma, e uma de 1 bit adicional que sinaliza a ocorrência de overflow, como pode ser visto na imagem a abaixo.

Figura 10 - Somador de 8 bits



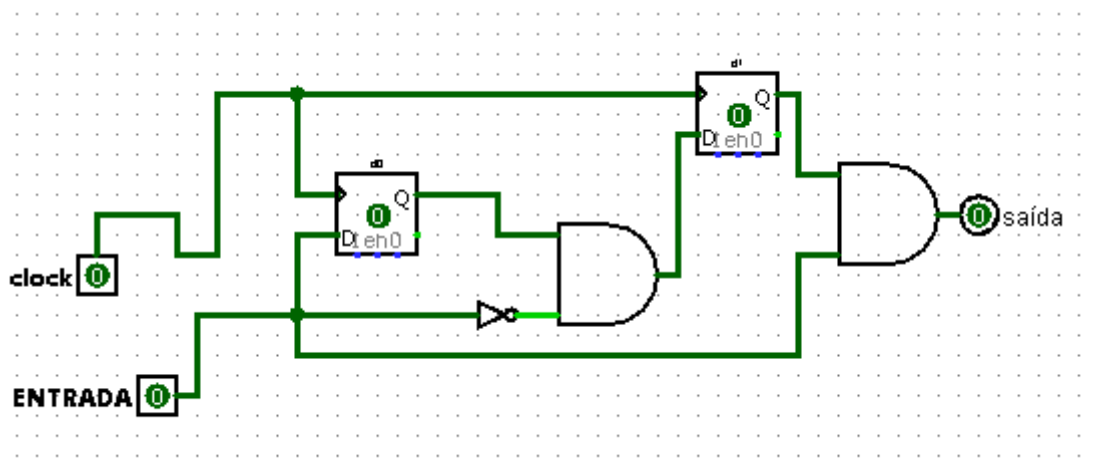
## 2.9. Detector de Sequência Binária 101

Para criar um circuito capaz de reconhecer a sequência binária "101", o processo começou com a projeção de uma máquina de estados finitos. Essa máquina opera com três estados distintos: A (codificado como 00), B (01) e C (10). A transição entre esses estados é controlada por dois flip-flops do tipo D (d1 e d0), que juntos indicam o estado atual do circuito.

Em seguida, foi elaborada uma tabela verdade para mapear o comportamento da máquina. As entradas foram definidas como o estado atual (d1, d0) e o bit de entrada (x). Já as saídas são o próximo estado (d1', d0') e a saída final (S).

Com base nessa tabela, foram criadas expressões lógicas que descrevem a dinâmica do detector. O resultado é um circuito que emite um sinal de saída 1 somente quando a sequência "101" é completada na entrada. Essa abordagem sistemática garante que o circuito se comporte de forma previsível e precisa.

Figura 11 - Detetector de Sequência Binária



## 2.10. Unidade Lógica Aritmética de 8 Bits

A Unidade Lógica e Aritmética (ULA) é um componente essencial em sistemas digitais, encarregada de executar operações lógicas e aritméticas. Nesse sentido, nossa ULA recebe dois operandos de 8 bits (A e B) e possui uma entrada de controle de 4 bits para selecionar uma de suas 10 operações. O resultado da operação escolhida é exibido em uma saída de 8 bits.

A ULA foi construída com componentes padrão do Logisim, incluindo portas lógicas, deslocadores de bits, um somador e um subtrator de 8 bits. O resultado de cada operação é direcionado para um multiplexador 16x1 de 8 bits, que, com base no código de 4 bits, seleciona e exibe a saída correta.

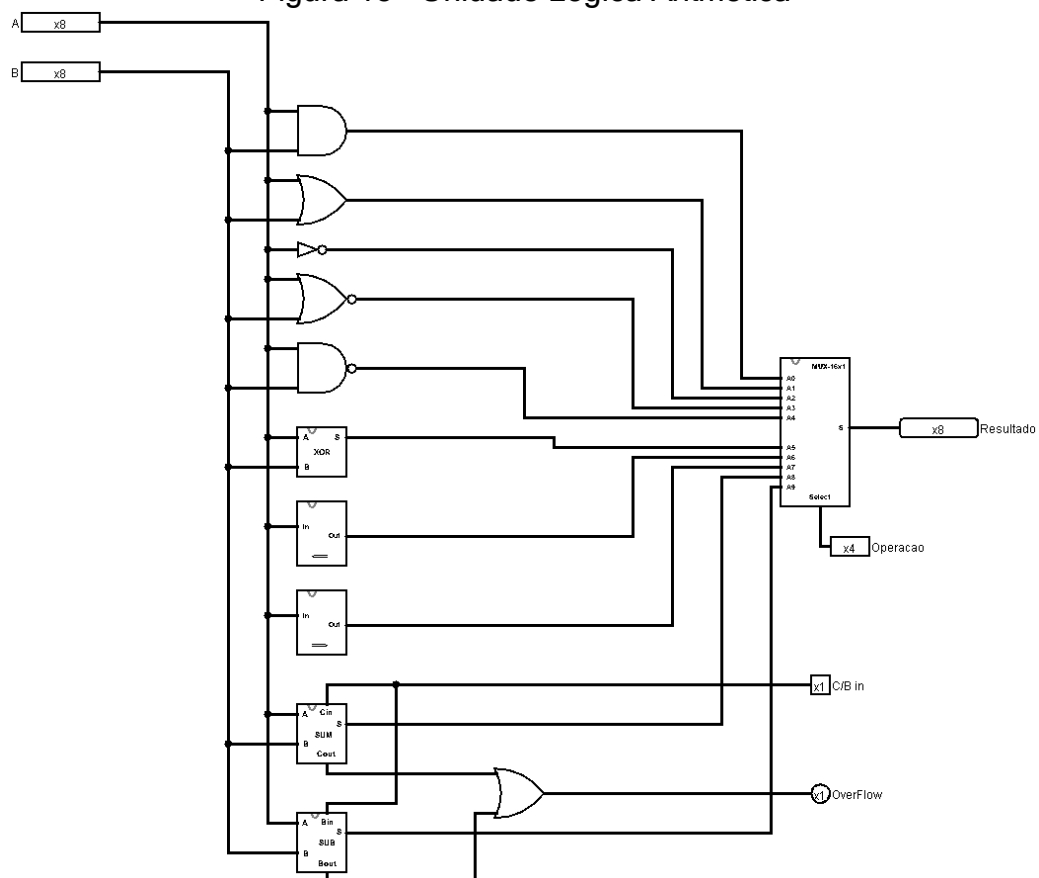
Ademais, a ULA foi aprimorada com o uso de deslocadores de bit e um subtrator. Para realizar os deslocamentos de 2 bits, dois circuitos de 1 bit foram conectados em série, com a direção controlada por uma constante. Já o subtrator de 1 bit foi desenvolvido a partir da lógica de um somador, com a expressão de Carry-Out ajustada para efetuar a operação de subtração.

A tabela abaixo detalha as operações da ULA, acessadas por códigos binários:

Figura 12 - Tabela de Comandos

Código	Operação	Descrição
0000	AND bit a bit	A saída é 1 apenas se ambos os bits de A e B forem 1.
0001	OR bit a bit	A saída é 1 se ao menos um dos bits de A ou B for 1.
0010	NOT bit a bit	Inverte cada bit do operando A.
0011	NOR bit a bit	A saída é 1 apenas se ambos os bits de A e B forem 0.
0100	NAND bit a bit	A saída é 0 apenas se ambos os bits de A e B forem 1.
0101	XOR bit a bit	A saída é 1 se os bits de A e B forem diferentes.
0110	SHIFT à Esquerda	Desloca os bits de A duas posições para a esquerda (equivalente a uma multiplicação por 4).
0111	SHIFT à Direita	Desloca os bits de A duas posições para a direita (equivalente a uma divisão por 4).
1000	Soma	Soma os operandos A e B, considerando o vai-um.
1001	Subtração	Subtrai B de A.

Figura 13 - Unidade Lógica Aritmética

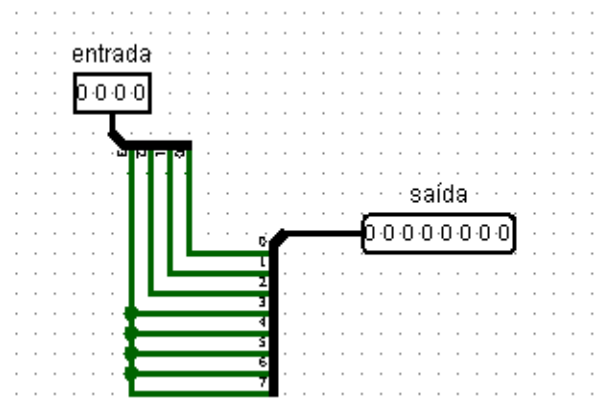


### 2.11. Extensor de sinal de 4 bits para 8 bits

Um extensor de sinal é um circuito digital projetado para aumentar o número de bits de um valor binário, sem alterar seu valor numérico. Ele é particularmente útil quando se trabalha com números em complemento de dois, pois garante que o sinal (positivo ou negativo) do número original seja preservado na nova representação.

Um extensor de 4 bits para 8 bits funciona da seguinte maneira: os 4 bits da entrada são copiados diretamente para as quatro posições menos significativas da saída. Para manter o sinal do número, o bit mais significativo (o bit de sinal) da entrada é replicado e preenche as quatro posições mais significativas da saída. Esse processo assegura que o número continue com o mesmo valor, mesmo com a ampliação para 8 bits.

Figura 14 - Extensor de sinal de 4 bits para 8 bits

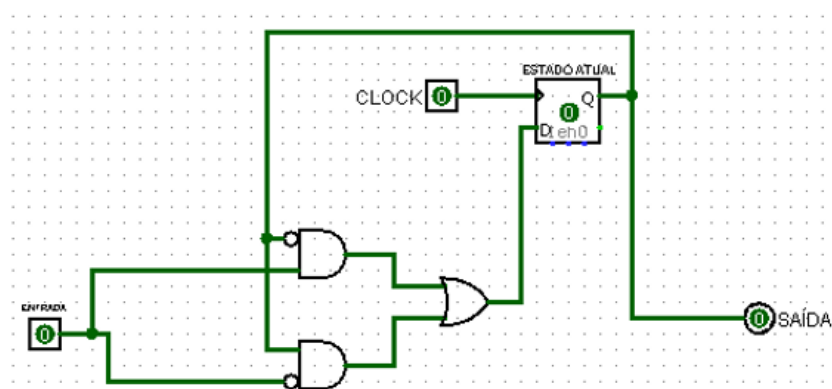


### 2.12. Máquina de Estados

Uma máquina de estados é um circuito digital que opera em uma sequência de estados predefinidos, com suas transições controladas por sinais de entrada e um sinal de clock. Frequentemente utilizada em sistemas de controle, ela altera seu comportamento com base em condições específicas.

A aplicação desse conceito é vista no circuito, que simula um semáforo de pedestres, que opera com apenas dois estados: verde e vermelho. A única entrada é um botão que, quando pressionado (estado 1), aciona a mudança de estado da máquina. Se o botão não for acionado (estado 0), o semáforo permanece no seu estado atual.

Figura 15 - Máquina de Estados







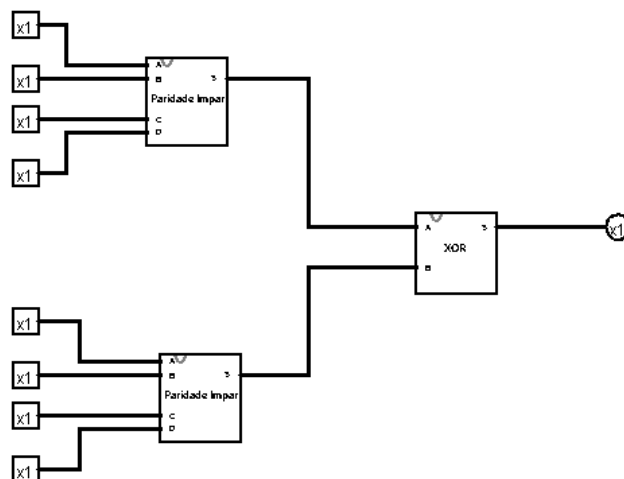
## 2.14. Detector de Paridade Ímpar

Um detector de paridade ímpar é um circuito digital fundamental, utilizado para verificar se a quantidade de bits com valor 1 em um conjunto de dados binários é ímpar. A lógica por trás disso se baseia na operação XOR (ou exclusivo), uma vez que essa porta lógica retorna um valor 1 apenas quando a quantidade de entradas com valor 1 for ímpar.

A implementação de um detector de 4 entradas, por exemplo, é feita com o uso de portas lógicas XOR, que processam as entradas em pares. Nesse sentido, a lógica do circuito é baseada na verificação das entradas uma a uma, e acumula o resultado para, no final, indicar a paridade. Essa abordagem é expressa como  $(a \oplus b) \oplus (c \oplus d)$ , que é a forma de realizar a operação XOR em 4 bits.

Adicionalmente, a lógica pode ser facilmente expandida para acomodar um número maior de entradas, bastando conectar as saídas de dois detectores de 4 entradas a uma porta XOR adicional. Dessa forma, o circuito mantém a sua funcionalidade para conjuntos de bits maiores, provando a eficiência e a escalabilidade desse método de detecção de paridade.

Figura 17 - Detector de Paridade Ímpar



## 2.15. Problema Otimização Lógica

A otimização de um circuito decodificador para displays de 7 segmentos, como o que acende o LED "C", é essencial para a sua eficiência. Para isso, usamos ferramentas de simplificação lógica, como os mapas de Karnaugh. Essa técnica nos permite analisar todas as combinações de entrada e determinar a expressão lógica mais simples para a saída desejada.

No caso do decodificador do LED "C", ele deve acender em todas as situações, exceto quando a entrada binária for o número 2. A aplicação de um mapa de Karnaugh para este cenário nos permite identificar as combinações que ativam o LED (saída 1) e aquelas que o mantêm apagado (saída 0).

A grande vantagem desse processo é que a expressão lógica final se torna muito mais simples. Isso não apenas reduz a quantidade de portas lógicas necessárias, mas também diminui a complexidade do circuito e, consequentemente, seu consumo de energia. O resultado é um circuito mais limpo, eficiente e que cumpre perfeitamente sua função.

Figura 18 - Circuito Antes de ser Simplificado

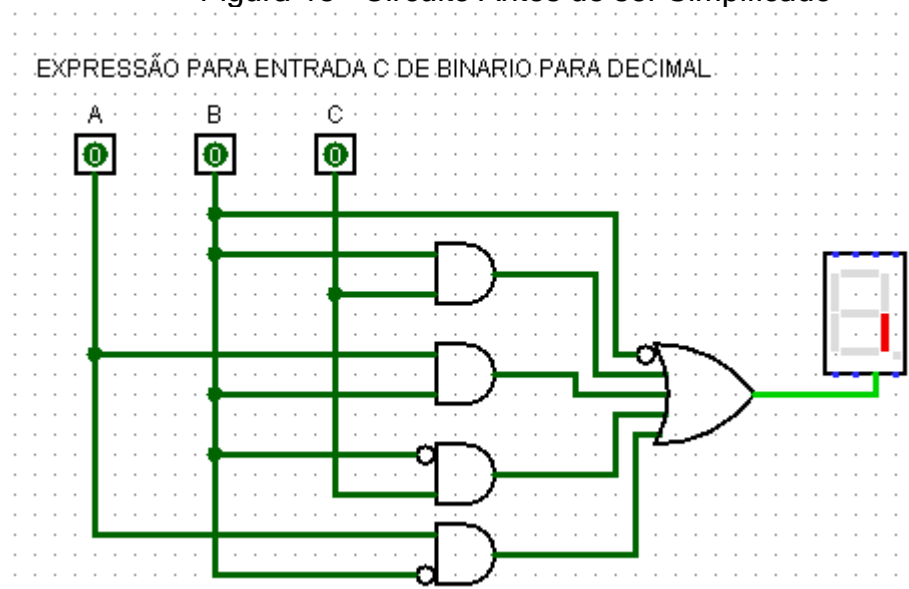
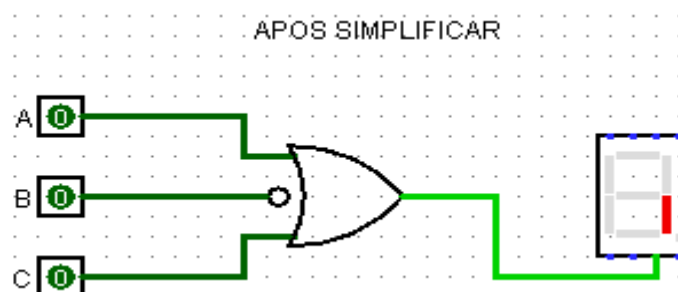


Figura 19 - Circuito Após Ser Simplificado



## 2.16. Decodificador de 7 Segmentos

O decodificador de 7 segmentos, projetado para converter um número binário de 4 bits em uma representação hexadecimal, é um componente essencial na visualização de dados. A sua função é traduzir a entrada binária em sinais que acionam os sete LEDs do display, que são identificados pelas letras de A a G.

Para desenvolver este circuito, foi preciso criar uma tabela verdade para cada um dos segmentos do display (A, B, C, D, E, F e G). As entradas da tabela correspondiam aos 4 bits binários, e a saída determinava o estado (ligado ou desligado) de cada LED. Dessa forma, as expressões lógicas necessárias para acionar o display foram obtidas, permitindo que os números de 0 a 9 e as letras de A a F fossem exibidos corretamente.

Figura 20 - Decodificador de 7 Segmentos

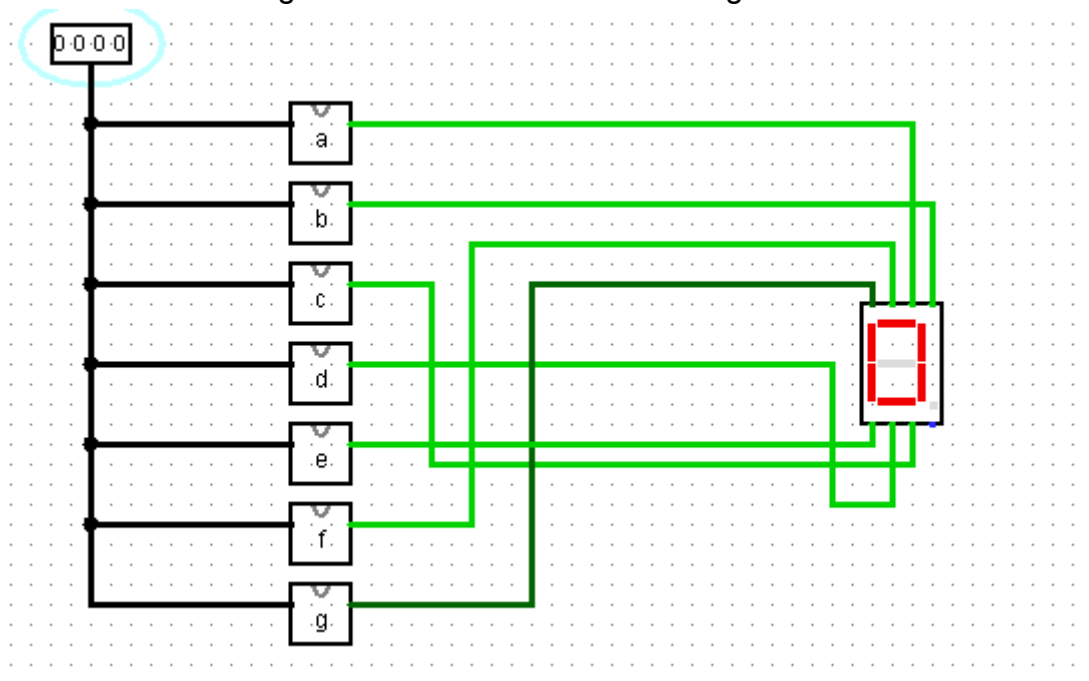
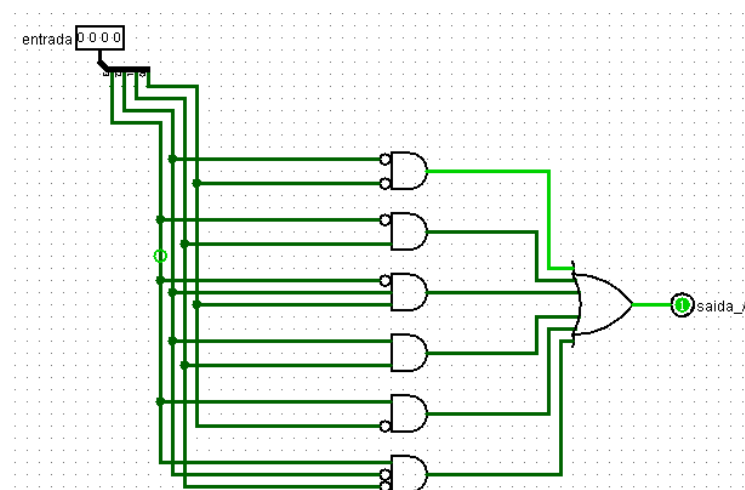


Figura 21 - Componente do Decodificador de 7 Segmentos

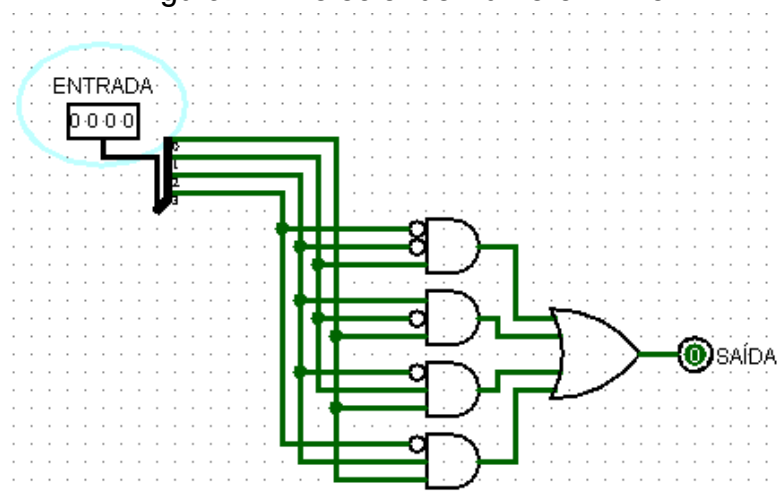


### 2.17. Detector de Número Primo de 4 entradas com Mapa de Karnaugh

O projeto de um detector de número primo de 4 bits é um desafio de lógica que exige uma abordagem sistemática. O circuito, que utiliza portas lógicas, recebe uma entrada de 4 bits (representada por A, B, C e D) e tem uma única saída (S), que será 1 somente se o número de entrada for primo. Nesse contexto, os números primos são 2, 3, 5, 7, 11 e 13.

Para otimizar o projeto, foram utilizadas duas ferramentas essenciais: a tabela verdade e o mapa de Karnaugh. Primeiro, a tabela verdade mapeou a saída desejada ( $S=1$ ) para cada uma das entradas binárias correspondentes aos números primos. Em seguida, o mapa de Karnaugh foi usado para agrupar as saídas lógicas, simplificando a expressão booleana e resultando em um circuito mais eficiente, com menos portas lógicas e menor complexidade.

Figura 22- Detector de Número Primo



### 3. REFERÊNCIAS

Silva, Helian; SANTIAGO, Lucas, MALAVÉ, Salvador. **Repositório AOC**. Disponível em: [https://github.com/LucasGilherme/AOC\\_HelianViniciusLucasGuilhermeSalvadorMalave\\_UFRR\\_LabCircuitos\\_2025](https://github.com/LucasGilherme/AOC_HelianViniciusLucasGuilhermeSalvadorMalave_UFRR_LabCircuitos_2025). Acesso em: 29 out. 2025.

PATTERSON, David A. **Organização e projeto de computadores**. 3. ed. Rio de Janeiro: Elsevier, 2005.