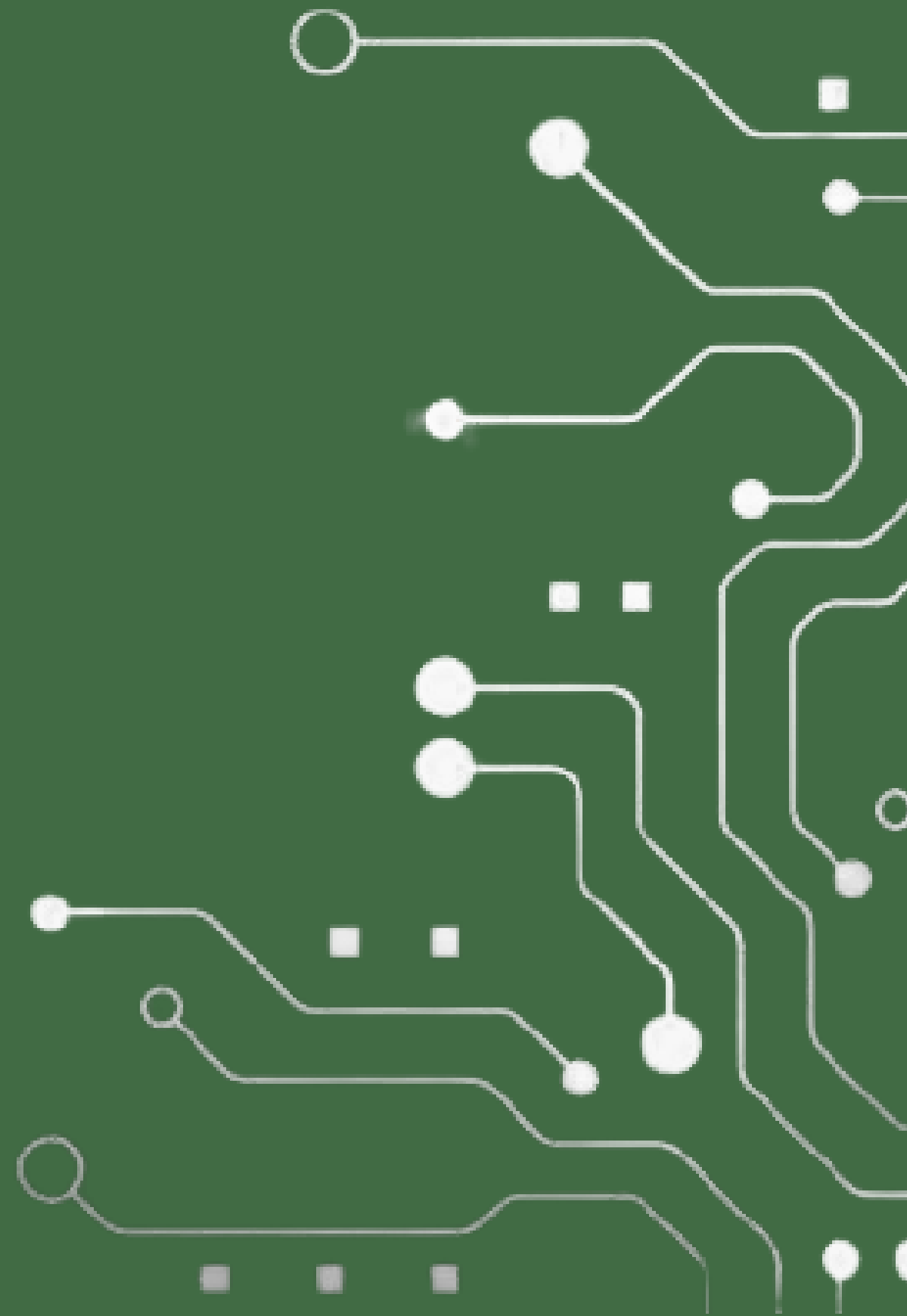




ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

LABORATÓRIO DE CIRCUITOS – CODIFICAÇÃO E SIMULAÇÕES

Amanda de Brito Barbosa, Fábio Aurélio Barros Alexandre e José Carvalho Neto





COMPONENTE 01 - FLIP-FLOPS



Flip-Flop Tipo D:

- É o mais direto; funciona como uma memória de "copia e cola".
- O que está na entrada D (Dados) é copiado para a saída Q quando o CLOCK pulsa.
- Se o clock não pulsar, ele ignora a entrada e guarda o último valor.

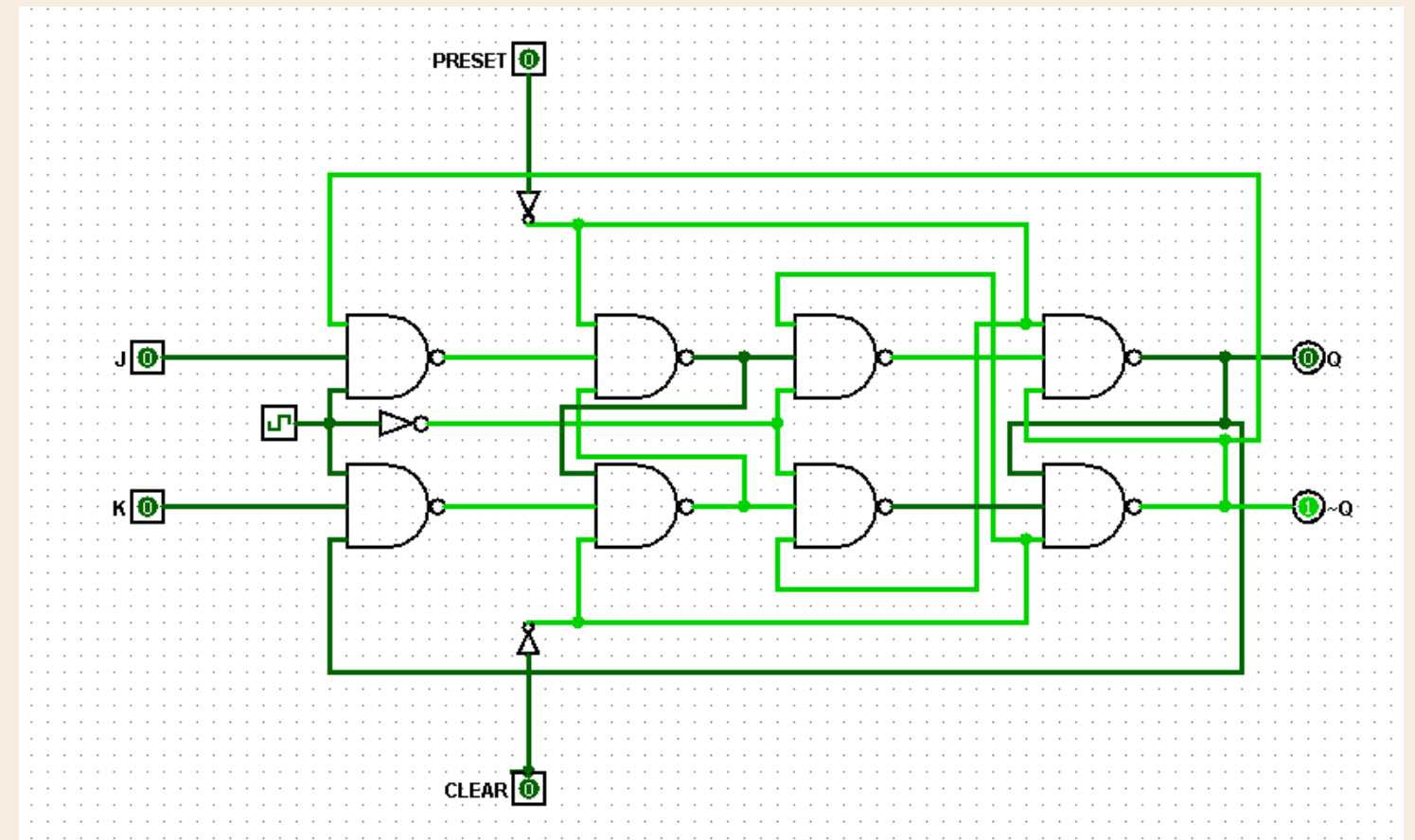
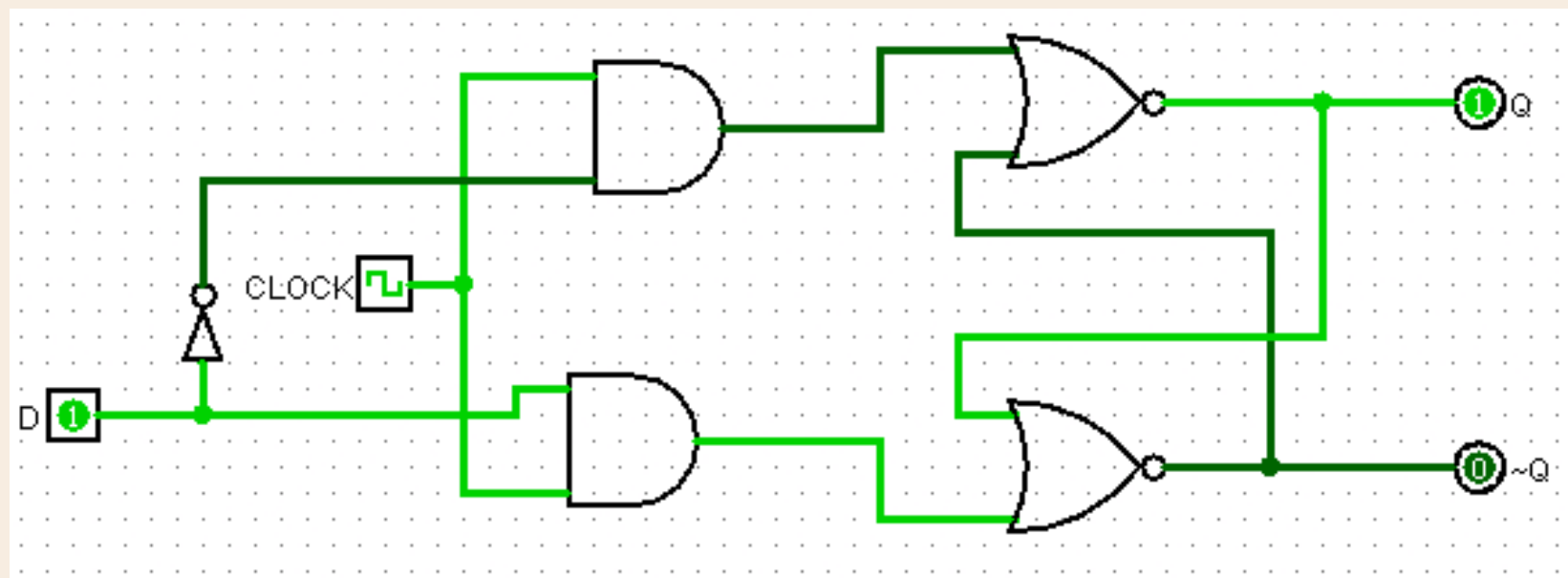


Flip-Flop Tipo JK:

- Possui duas entradas (J e K) e pode ser programado para quatro funções:
- J=0, K=0: Manter o estado.
- J=0, K=1: Resetar (saída vai para 0).
- J=1, K=0: Setar (saída vai para 1).
- J=1, K=1: Bascular (inverte o valor atual).

COMPONENTE 01 - FLIP-FLOPS

Flip-Flop Tipo D



Flip-Flop Tipo JK



PARA QUE SERVE?



- Armazenamento de 1 Bit: São os blocos de construção fundamentais da memória.



- Memória RAM: Células de memória são feitas com Flip-Flops D para armazenar cada bit de dados.



- Contadores: Contadores síncronos usam flip-flops (como o Tipo T, derivado do JK) para avançar a contagem a cada pulso do clock.



- Máquinas de Estado: O Flip-Flop D é o elemento de memória que armazena o "estado atual" da máquina.

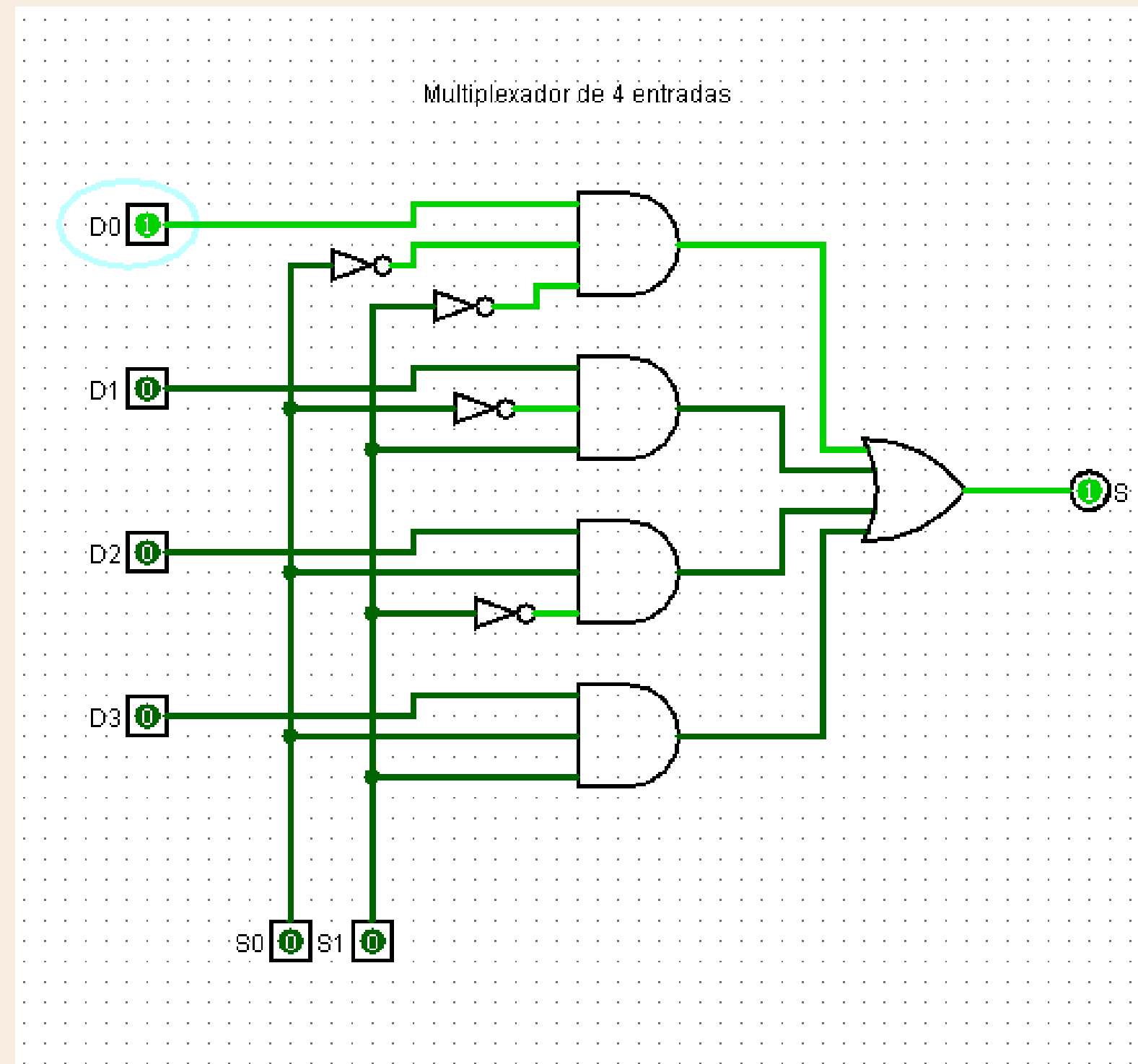


COMPONENTE 02 - MULTIPLEXADOR



- Um multiplexador (MUX) é um circuito lógico que funciona como um seletor.
- Ele permite que apenas uma de suas várias entradas seja enviada para uma única saída.
- A seleção de qual entrada passará é determinada pelas "chaves seletoras".
- Este circuito específico tem 4 entradas de dados (D0-D3), 2 chaves seletoras (S0, S1) e 1 saída.

COMPONENTE 02 - MULTIPLEXADOR





PARA QUE SERVE?



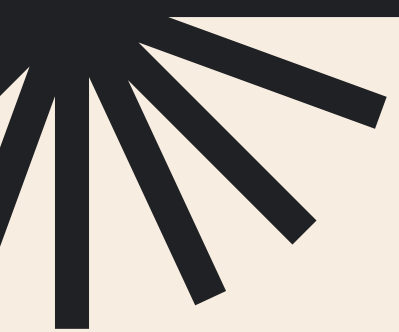
- Roteamento de Dados: A principal função é selecionar qual dado deve seguir por um caminho.



- Seleção de Leitura na RAM: Em uma memória RAM, um multiplexador é usado para selecionar qual célula de memória (qual registrador) será lida e enviada para a saída.



- Saída da ULA (ALU): Em uma ULA, várias operações (soma, AND, OR, etc.) são calculadas ao mesmo tempo. Um multiplexador na saída seleciona qual desses resultados deve ser a saída final da ULA, com base no "opcode" (sinal de controle).

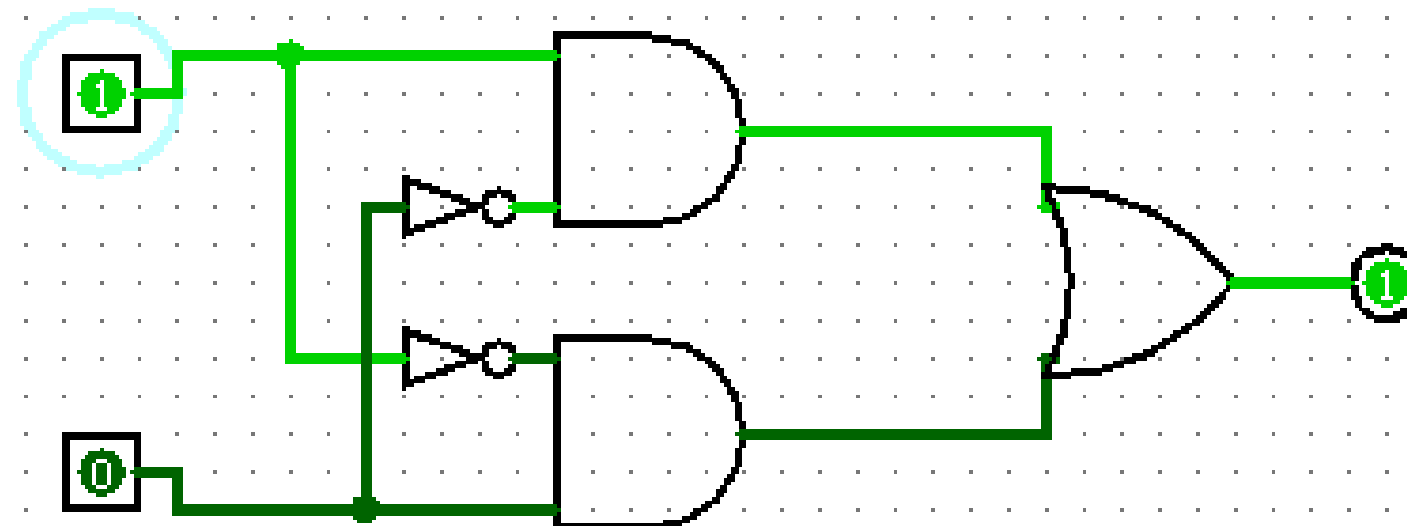


COMPONENTE 03 - PORTA LÓGICA XOR



- XOR (Ou Exclusivo) é a porta lógica que resulta em 1 (Verdadeiro) somente quando suas entradas são diferentes.
- Neste projeto, ela foi implementada usando apenas os componentes básicos: AND, OR e NOT.

COMPONENTE 03 - PORTA LÓGICA XOR





PARA QUE SERVE?



- Aritmética (Soma): É um componente essencial no "Somador Completo". A lógica da soma de dois bits $A \oplus B$ é uma operação XOR.



- Detector de Paridade: É a forma mais simples de verificar paridade. Um detector de paridade ímpar pode ser construído como uma representação em módulos da porta XOR.



- Comparação: Pode ser usado para verificar se dois bits ou dois números (bitwise) são diferentes.

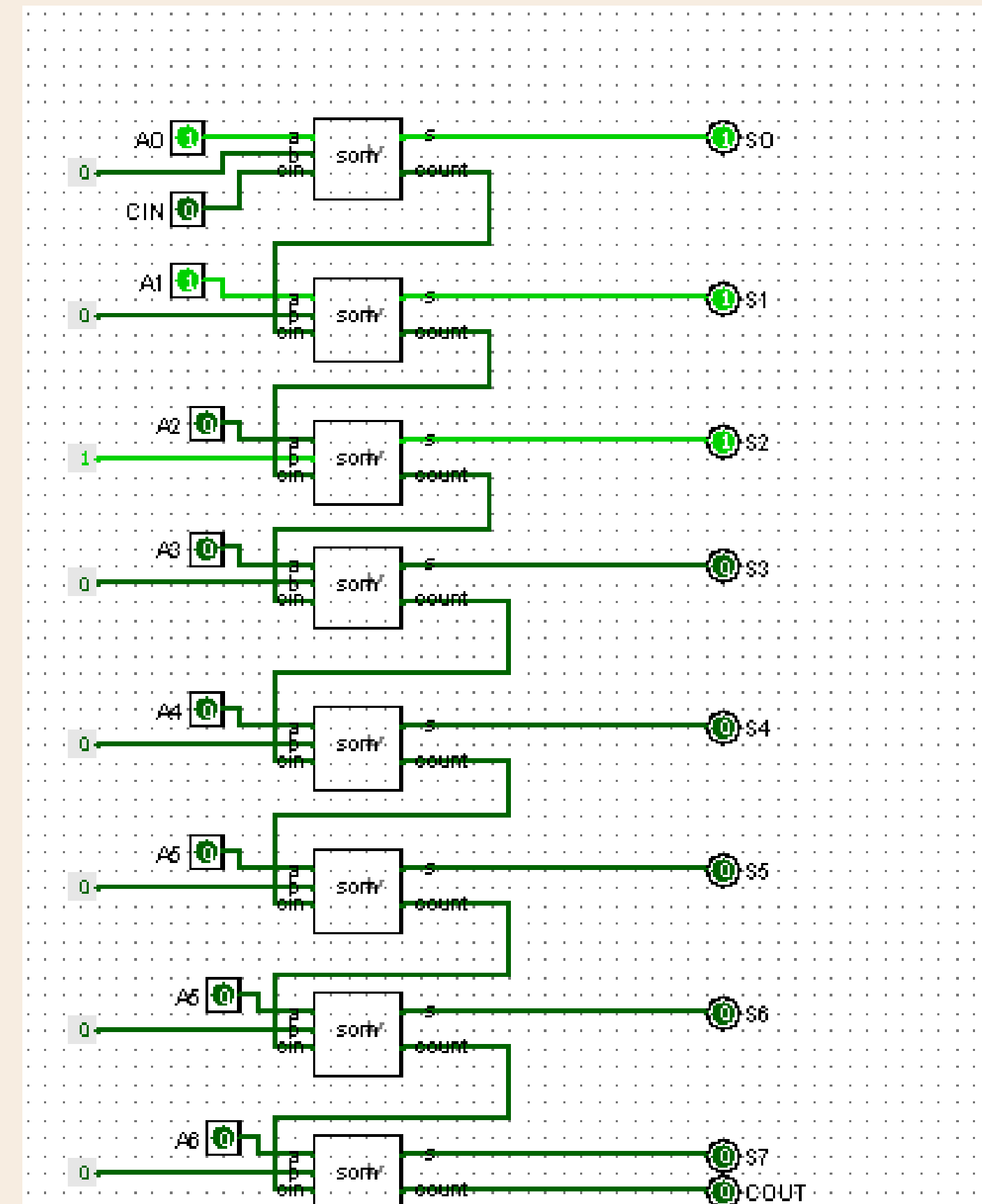
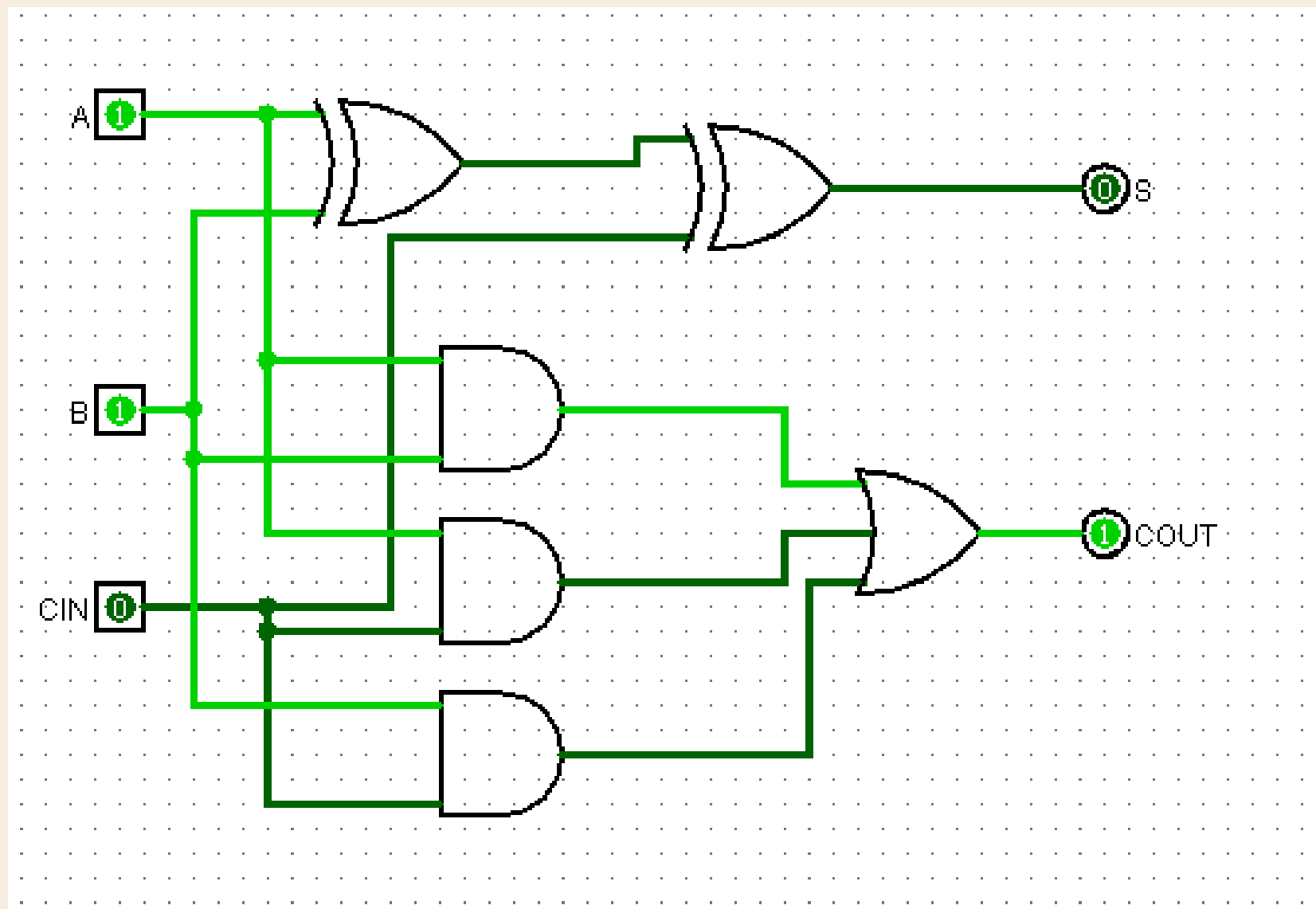


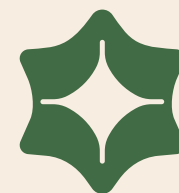
COMPONENTE 04 - SOMADOR (VALOR + 4)



- Um circuito somador de 8 bits especializado.
- Diferente de um somador comum, ele não recebe duas entradas variáveis.
- Ele tem uma única missão: receber um número de 8 bits (A) e somar o valor fixo 4 a ele.

COMPONENTE 04 - SOMADOR (VALOR + 4)





PARA QUE SERVE?

- Este é um exemplo de lógica "hardwired" (fixa).
- Serve para implementar operações aritméticas muito específicas e constantes, onde um dos valores da operação nunca muda.
- Em um processador, isso pode ser usado para instruções que incrementam um valor por uma constante (como ADDI - Add Immediate).
- É mais eficiente para esta tarefa específica do que usar uma ULA completa.

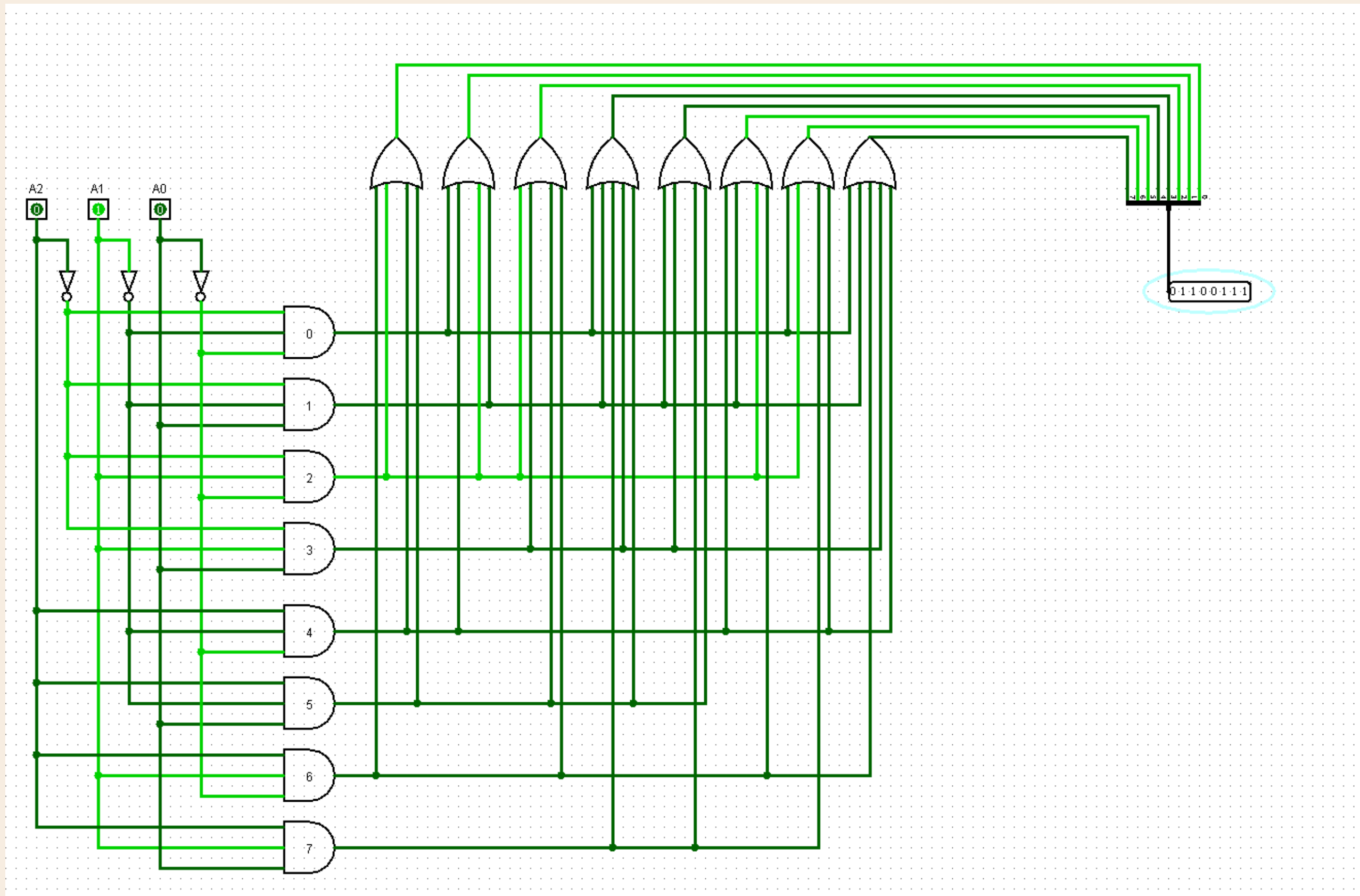


COMPONENTE 05 - MEMÓRIA ROM



- ROM significa "Read-Only Memory" (Memória Apenas de Leitura).
- É um tipo de memória não volátil, o que significa que ela guarda os dados mesmo sem energia.
- É usada para armazenar dados e instruções essenciais e permanentes.
- Este circuito implementa uma ROM de 8 palavras por 8 bits (8x8), usando 3 pinos de endereço (A0, A1, A2) para selecionar qual das 8 palavras será lida.

COMPONENTE 05 - MEMÓRIA ROM





PARA QUE SERVE?



- BIOS de Computadores: Armazena o código de inicialização do hardware.



- Sistemas Embarcados: Guarda o "firmware" de dispositivos (ex: micro-ondas, controle remoto).



- Cartuchos de Videogame: Antigamente, os jogos eram armazenados em chips de ROM.



- Em geral, é usada para qualquer dado que precisa estar disponível assim que o dispositivo é ligado.

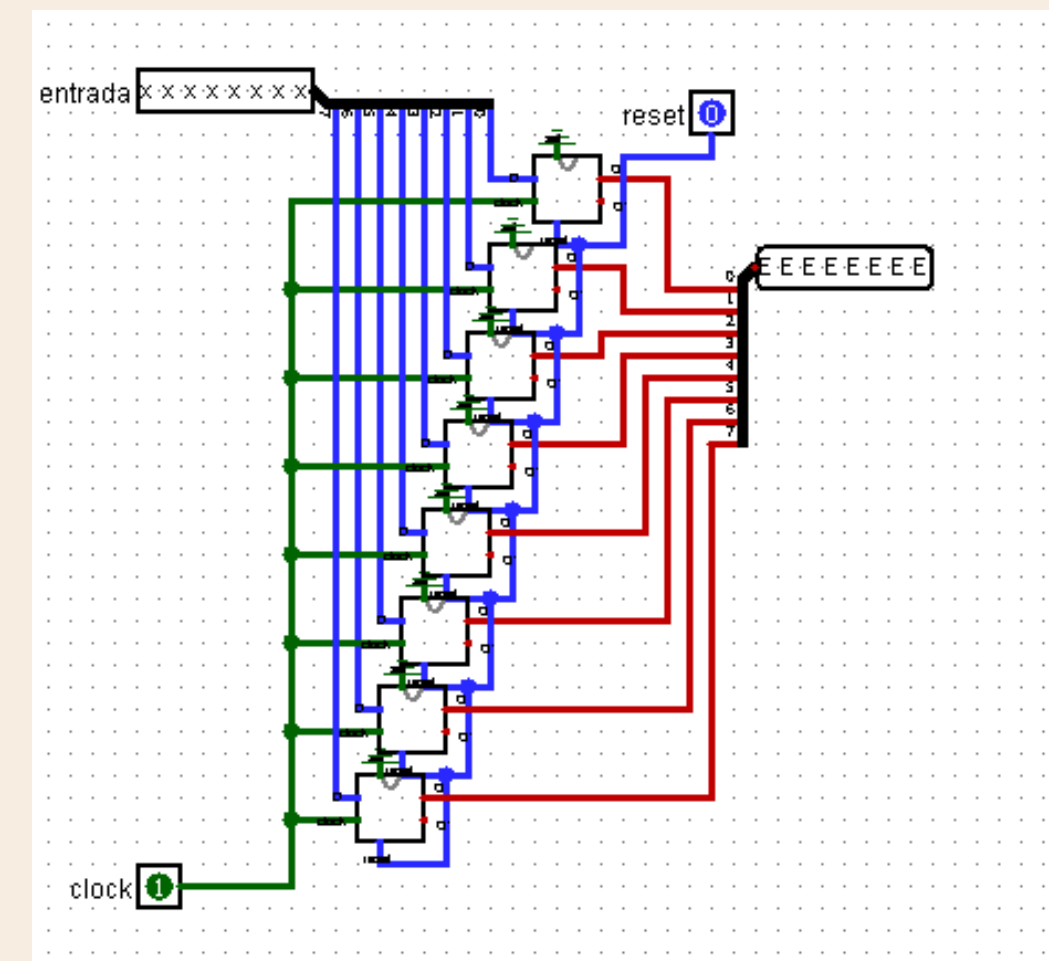
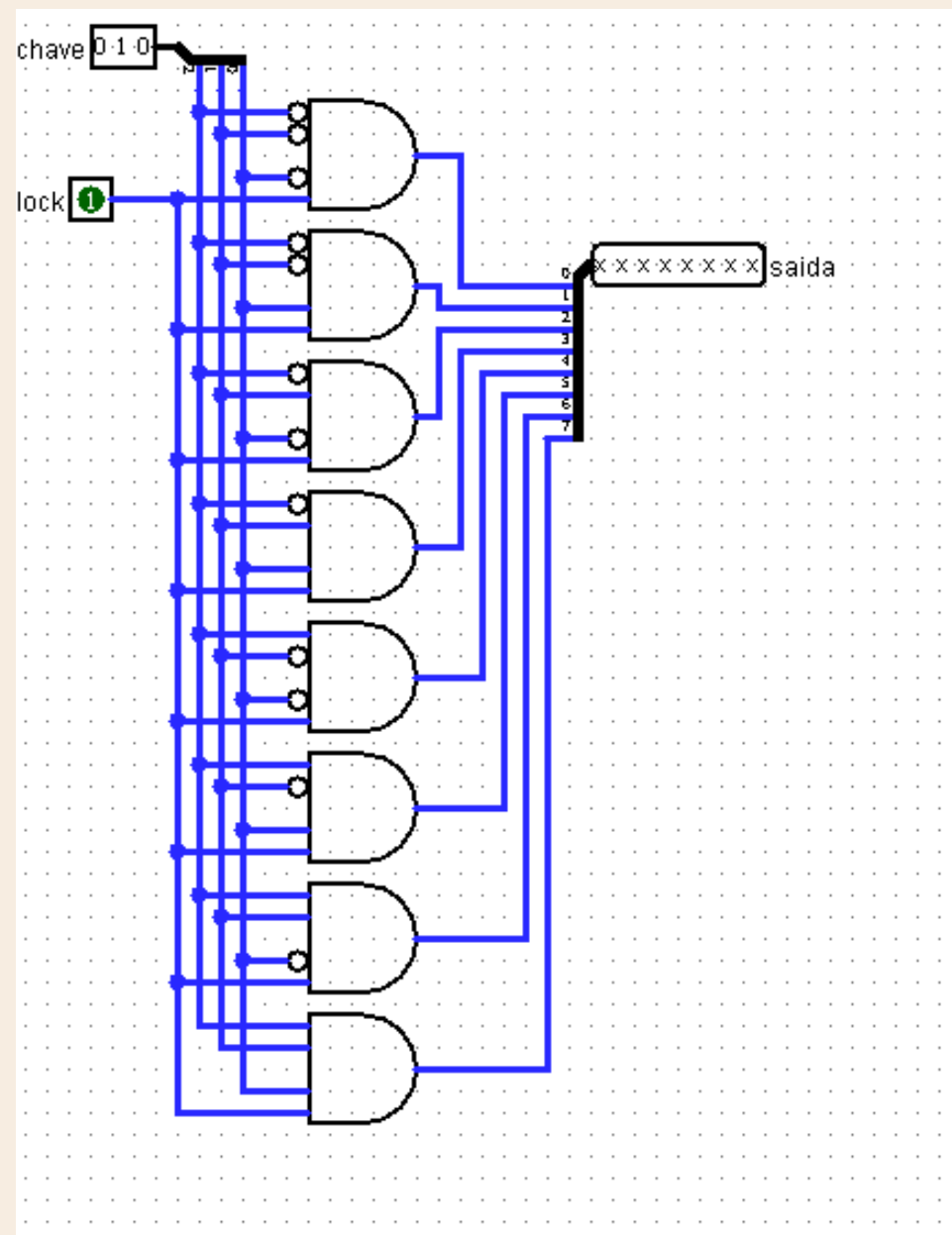
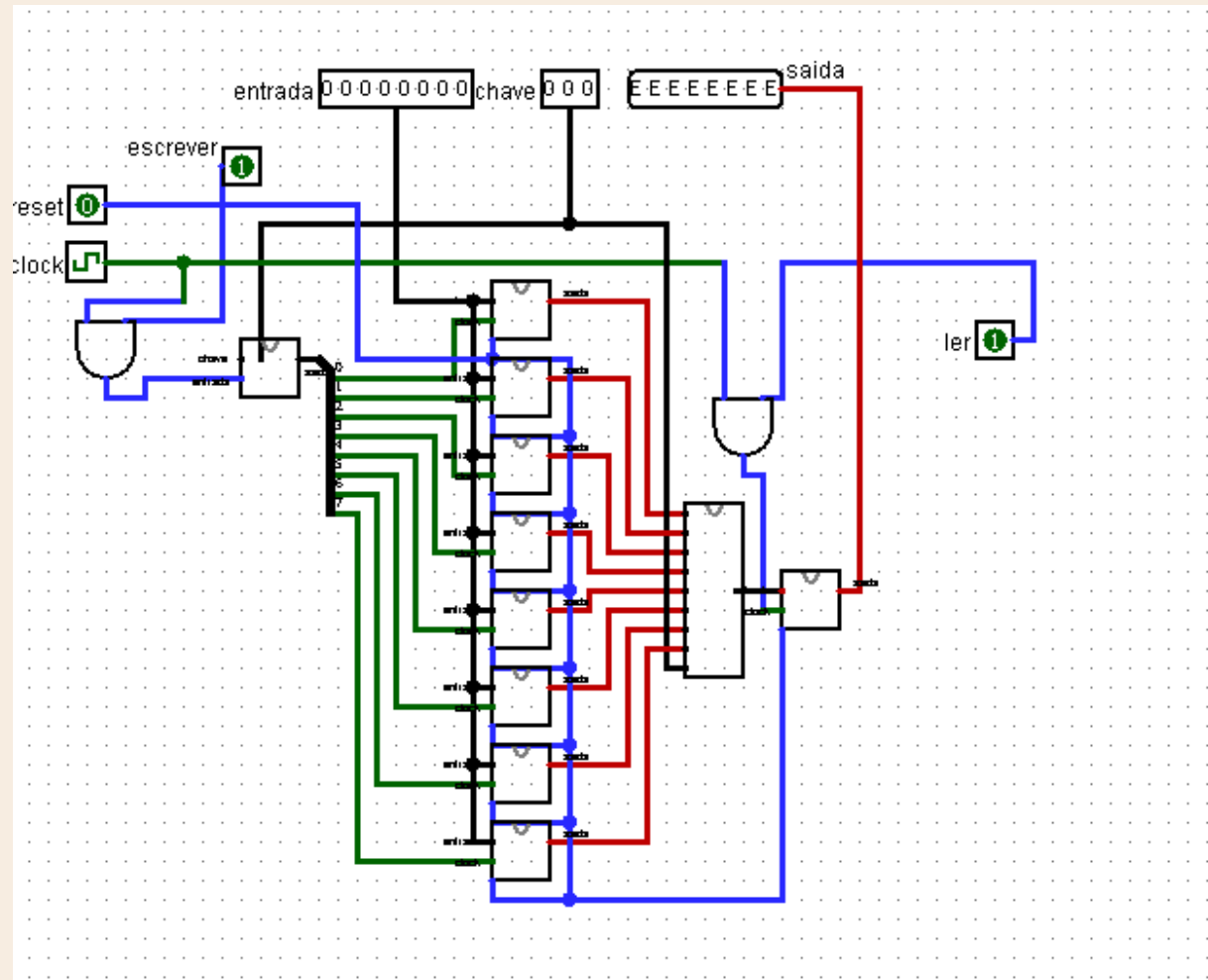


COMPONENTE 06 - MEMÓRIA RAM



- RAM significa "Random Access Memory" (Memória de Acesso Aleatório).
- É a memória "de trabalho" do computador, usada para leitura e escrita de dados.
- Este circuito usa Flip-Flops D como células de memória para armazenar cada bit.
- Utiliza um Demultiplexador para direcionar o sinal de "escrever" para a célula correta.
- Utiliza um Multiplexador para selecionar qual célula será lida e enviada para a saída.

COMPONENTE 06 - MEMÓRIA RAM





PARA QUE SERVE?



- Memória Volátil: É a principal aplicação. Armazena programas e dados que estão em uso ativo (ex: o sistema operacional, o seu navegador).



- Sistemas Embarcados: Usada como pequenas RAMs para dispositivos de baixo custo.



- Diferente da ROM, a RAM é volátil (perde os dados sem energia) mas permite leitura e escrita em tempo real.

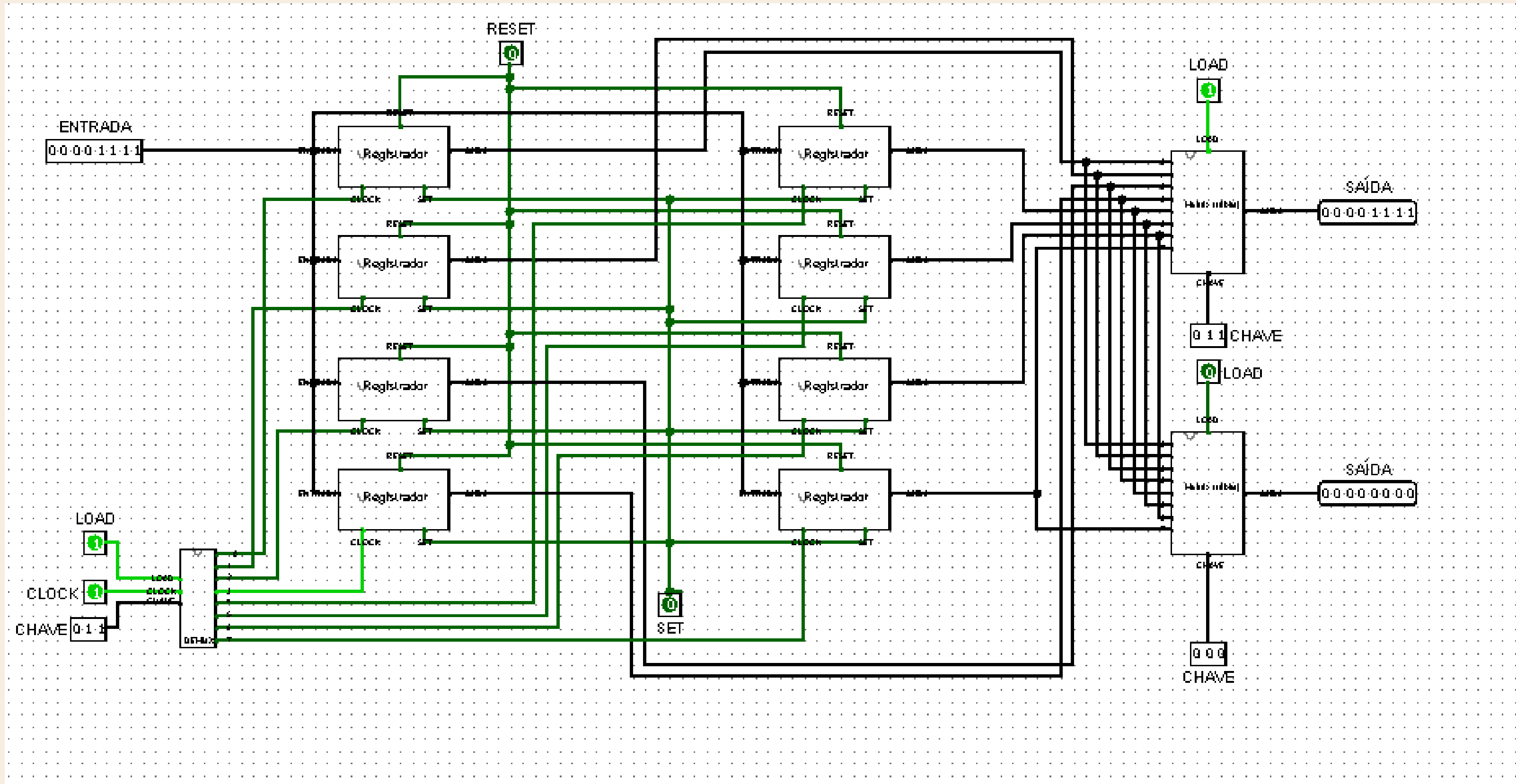


COMPONENTE 07 - BANCO DE REGISTRADORES



- É a "mesa de trabalho" do processador; um pequeno "gaveteiro" super rápido.
- Não guarda arquivos grandes, mas sim os poucos números que o processador está usando exatamente agora para um cálculo.
- Este projeto tem 8 "gavetas" (registradores), cada uma guardando 8 bits.
- Sua grande vantagem: permite escrever em um registrador e, ao mesmo tempo, ler de outros dois.

COMPONENTE 07 - BANCO DE REGISTRADORES





PARA QUE SERVE?



- Agilizar Operações: É essencial para organizar e acelerar as operações dentro da CPU.



- Interface com a ULA: Ele "segura" os operandos (ex: os números 5 e 10) que serão enviados para a ULA para uma soma.



- Armazena temporariamente o resultado do cálculo da ULA antes que ele seja enviado para a memória RAM.

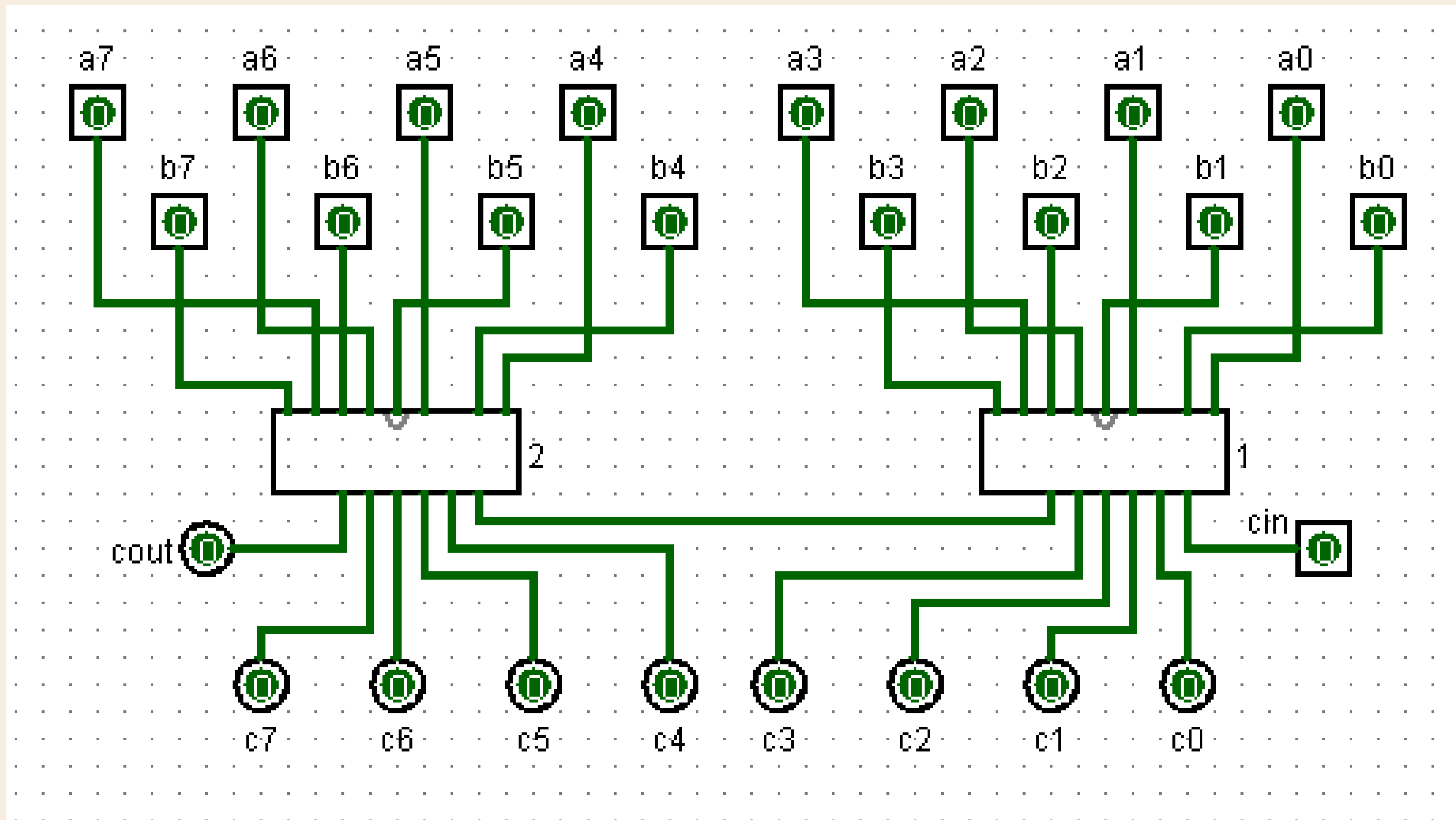


COMPONENTE 08 - SOMADOR DE 8 BITS



- Um dos circuitos mais fundamentais para realizar operações aritméticas em computadores.
- Ele é construído de forma modular:
- Começa com um somador de 1 bit.
- Quatro somadores de 1 bit formam um somador de 4 bits.
- Dois somadores de 4 bits formam o somador de 8 bits.
- O "Carry-Out" (vai-um) de um somador é conectado ao "Carry-In" do próximo

COMPONENTE 08 - SOMADOR DE 8 BITS





PARA QUE SERVE?



- Núcleo da ULA: É o componente central da Unidade Lógica e Aritmética (ULA) para operações de SOMA.



- Base para Subtração: Usando a lógica de "complemento de dois", este mesmo circuito pode ser usado para realizar subtrações.



- Qualquer cálculo aritmético complexo em um computador é decomposto em operações básicas como esta.

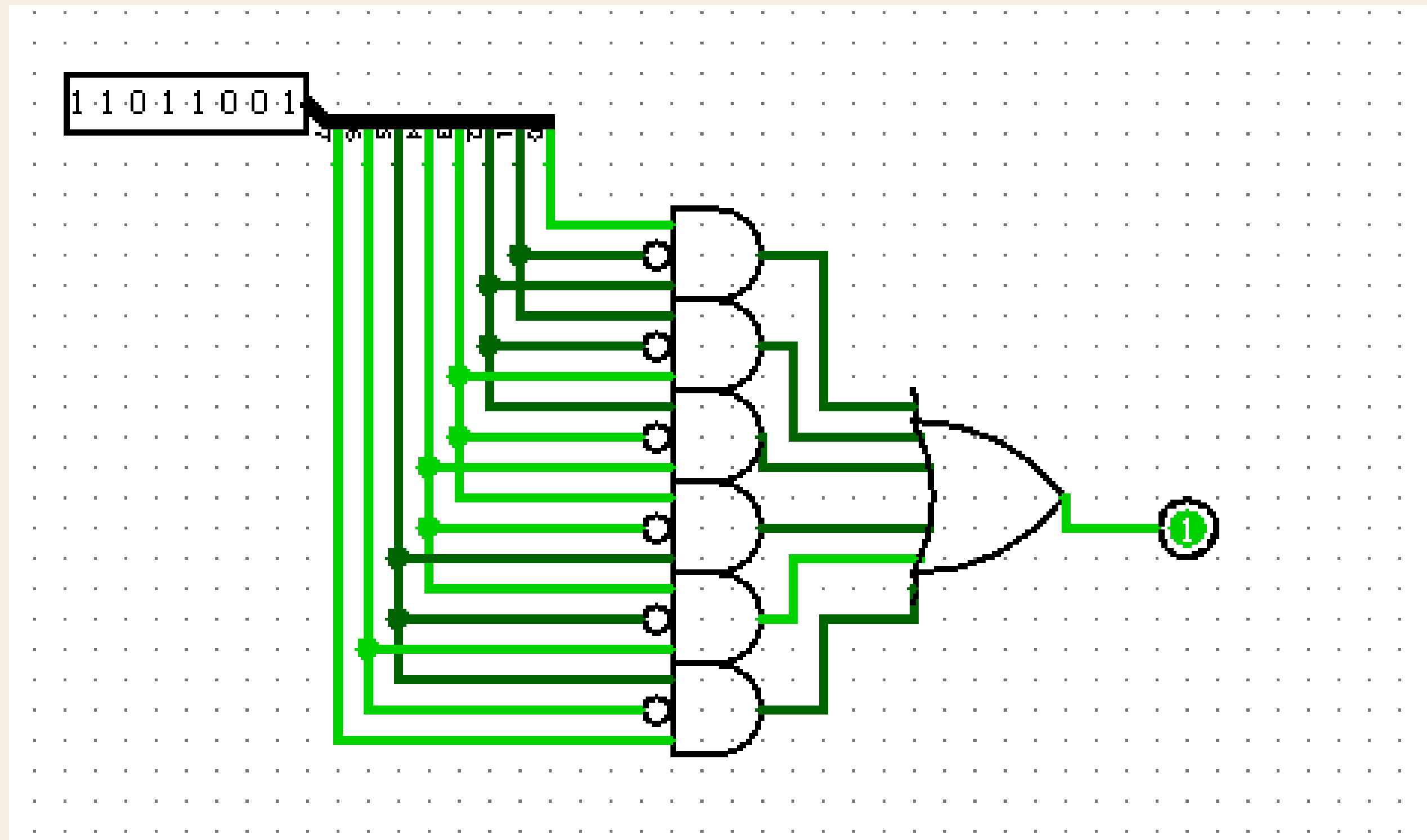


COMPONENTE 09 - DETECTOR DE SEQUÊNCIA 101



- Um circuito lógico projetado para detectar um padrão binário específico (neste caso, "101") em um fluxo de bits de entrada.
- Ele utiliza portas lógicas básicas (AND, OR, NOT) para validar as condições .
- A saída do circuito é um único bit, que se torna '1' (ativo) apenas quando a sequência "101" é identificada na entrada.

COMPONENTE 09 - DETECTOR DE SEQUÊNCIA 101





PARA QUE SERVE?



- Comunicação Digital: Usado para identificar padrões específicos, como cabeçalhos de pacotes ou sinais de controle em fluxos de dados.



- Máquinas de Estado: Pode ser usado para monitorar eventos em sistemas digitais.



- Filtros de Sinal: Permite extrair informações relevantes de uma sequência binária longa.

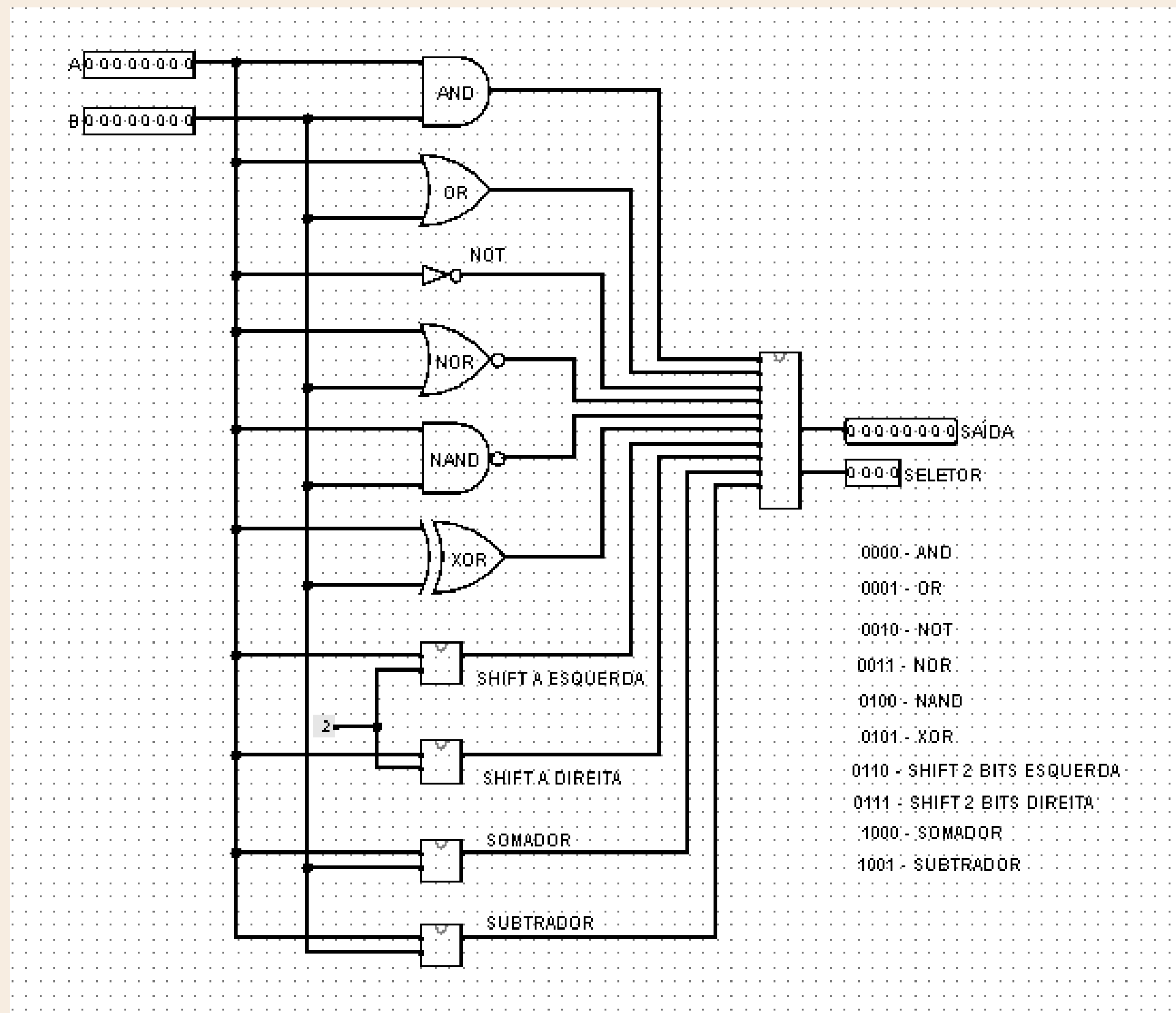


COMPONENTE 10 - ULA (UNIDADE LÓGICA E ARITMÉTICA)



- A ULA (ou ALU, em inglês) é o "núcleo computacional" ou a "calculadora" de uma CPU.
- Sua função é executar todas as operações aritméticas (Soma, Subtração) e lógicas (AND, OR, XOR, etc.).
- Esta ULA de 8 bits pode realizar 10 operações diferentes.
- Ela calcula todos os resultados em paralelo, e um Multiplexador na saída usa um "Sinal de Seleção" (opcode) para escolher qual resultado será a saída final.

COMPONENTE 10 - ULA (UNIDADE LÓGICA E ARITMÉTICA)





PARA QUE SERVE?



- Execução de instruções: É a parte do processador que "faz as contas".



- Quando o processador lê uma instrução como ADD ou AND, ele configura o sinal de seleção da ULA e envia os dados (dos registradores) para ela calcular o resultado.



- É fundamental para a execução de qualquer programa de computador.

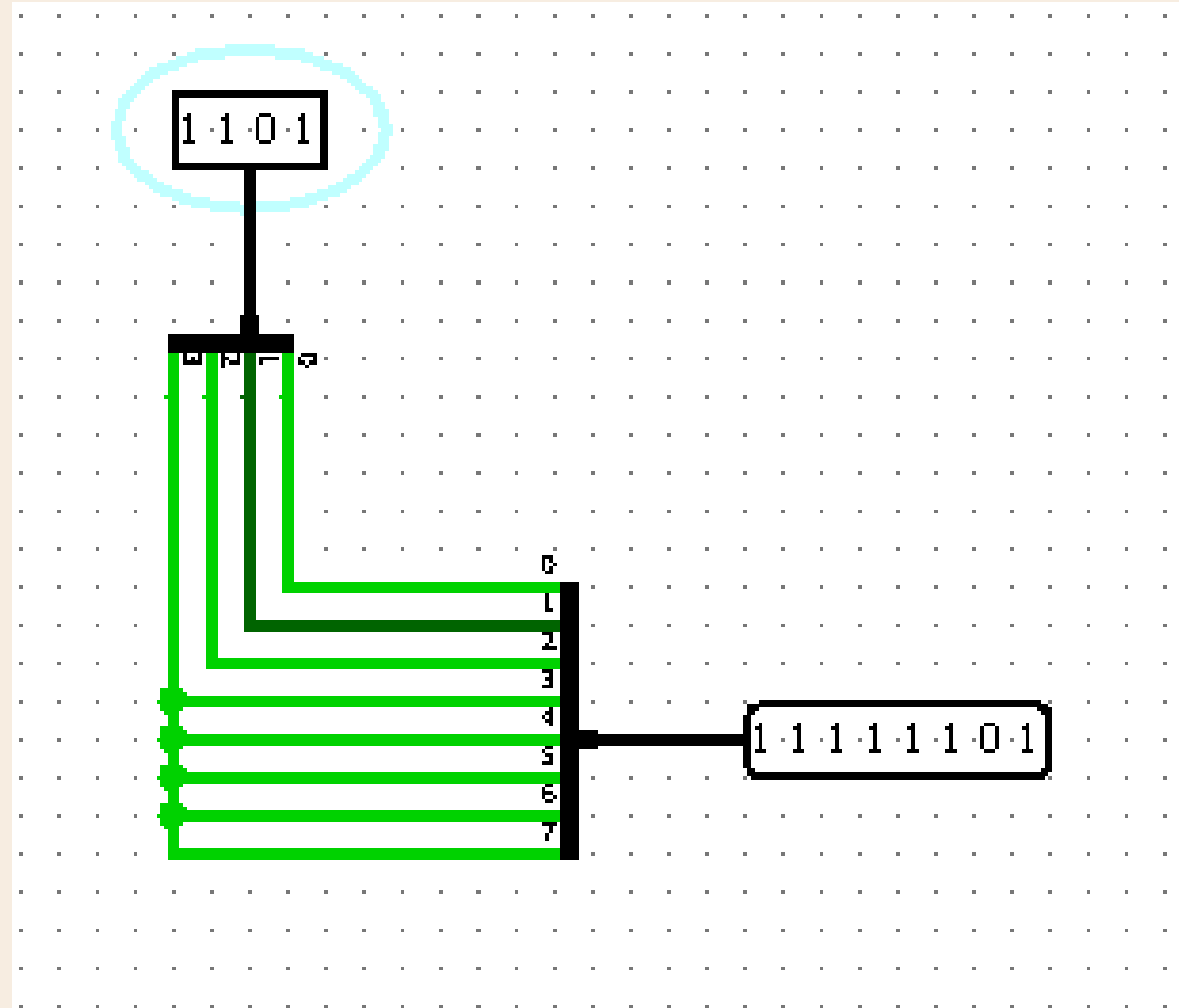


COMPONENTE 11 - EXTENSOR DE SINAL



- Um circuito que converte um número com sinal de 4 bits para um número de 8 bits, sem alterar seu valor.
- Isso é crucial para números negativos (em complemento de dois).
- Ele pega o bit mais significativo do número de 4 bits.
- Em seguida, ele replica (copia) esse bit de sinal para todos os 4 novos bits adicionados na saída de 8 bits.

COMPONENTE 11 - EXTENSOR DE SINAL





PARA QUE SERVE?



- Permite que operações aritméticas sejam feitas entre números de tamanhos diferentes (ex: somar um número de 4 bits com um de 8 bits).



- Garante que o valor de um número negativo seja preservado corretamente quando ele é "promovido" para um tipo de dado maior.

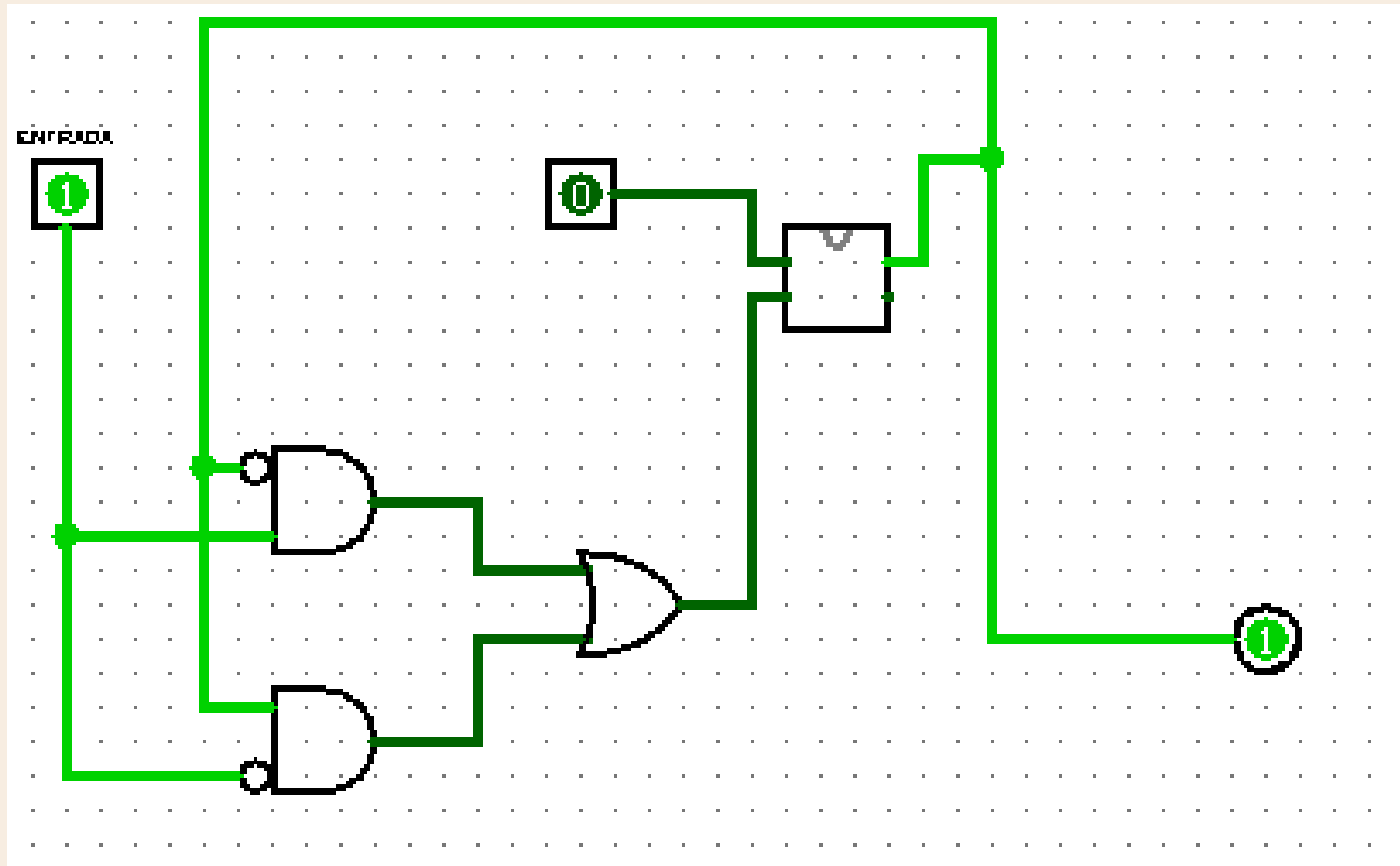


COMPONENTE 12 - MÁQUINA DE ESTADOS



- Um modelo matemático usado para controlar sistemas que têm um número finito de "estados".
- As transições entre os estados acontecem com base em entradas e pulsos de clock.
- É composta por dois blocos principais:
- Memória (Flip-Flop D): Armazena o "Estado Atual" da máquina.
- Lógica Combinacional (Portas): Calcula o "Próximo Estado" com base no estado atual e na entrada.

COMPONENTE 12 - MÁQUINA DE ESTADOS





PARA QUE SERVE?



- Controle de Sistemas: É a base para qualquer lógica sequencial.



- Exemplos:

1. Controle de um semáforo (estados: verde, amarelo, vermelho).
2. Contadores sequenciais.
3. Automação industrial e controle de robôs.
4. A "Unidade de Controle" de um processador é uma grande máquina de estados.

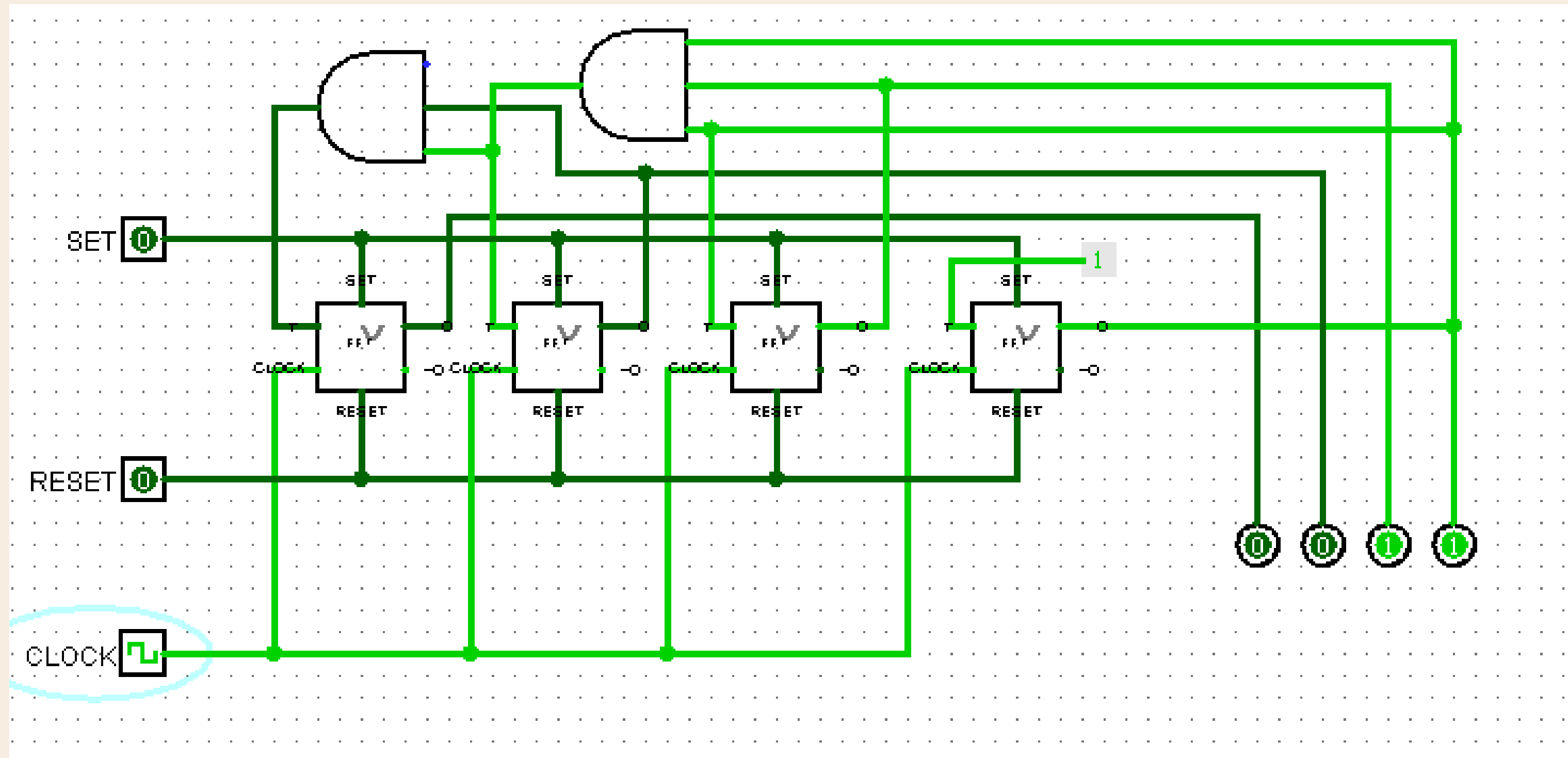


COMPONENTE 13 - CONTADOR SÍNCRONO



- Um circuito que, como o nome diz, "conta".
- Ele usa Flip-Flops para avançar a contagem em binário (001, 010, 011...) a cada pulso do CLOCK.
- O termo "Síncrono" significa que todos os flip-flops recebem o sinal de CLOCK ao mesmo tempo e mudam de estado em perfeita sincronia.

COMPONENTE 13 - CONTADOR SÍNCRONO





PARA QUE SERVE?



- Contagem de eventos: Usado em qualquer lugar que precise contar pulsos de clock ou eventos.



- Divisor de frequência: Pode ser usado para criar um sinal de clock mais lento a partir de um mais rápido.



- Contador de programa: Em um processador, o PC é um contador que aponta para o endereço da próxima instrução a ser executada.



COMPONENTE 14

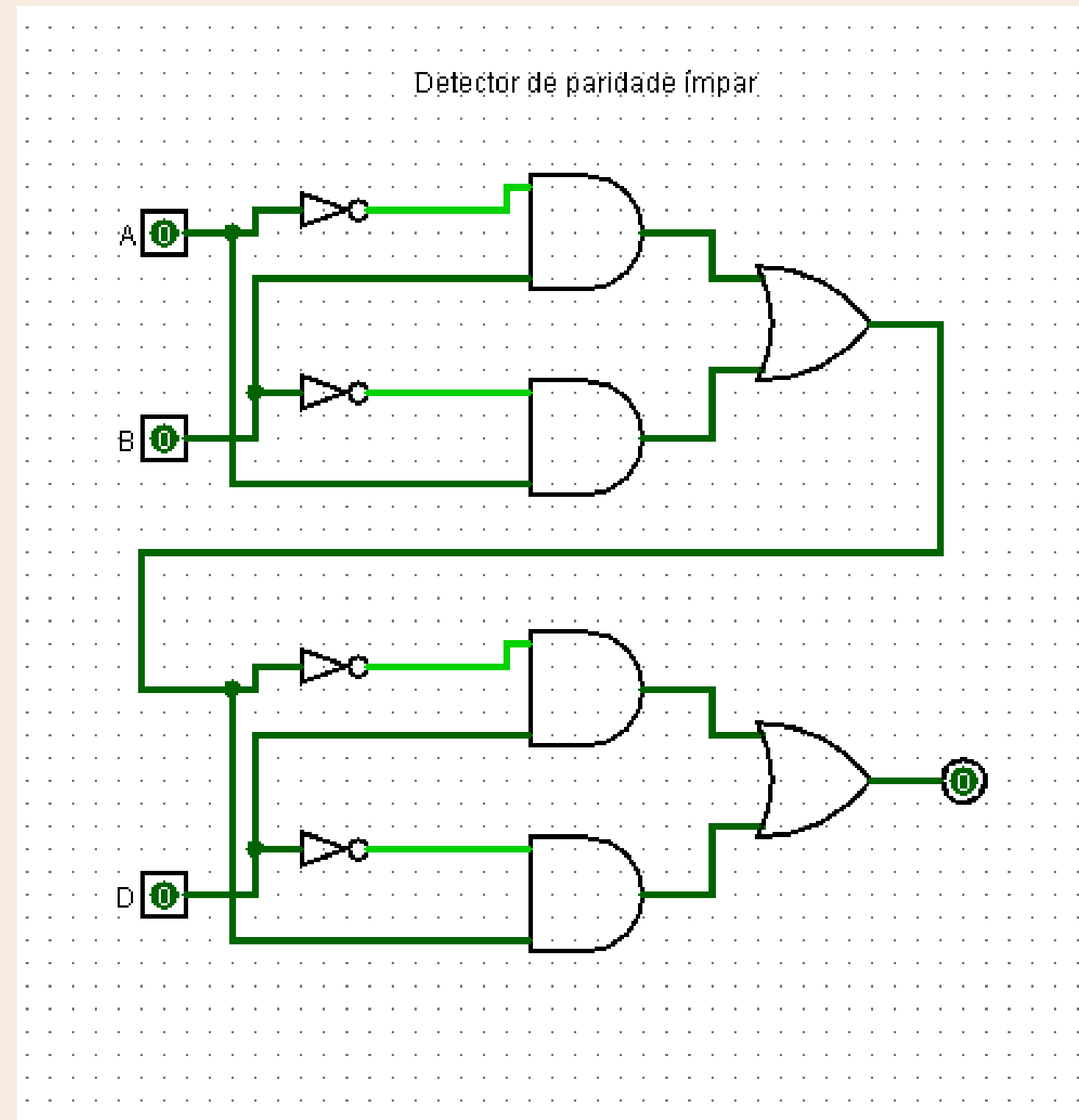
DETECTOR DE PARIDADE ÍMPAR



- Um circuito que verifica se o número de bits '1' em uma entrada é par ou ímpar.
- Se a quantidade de bits '1' for ímpar, a saída é 1.
- Se a quantidade de bits '1' for par, a saída é 0.
- Este circuito (e sua tabela verdade) é funcionalmente equivalente a uma cascata de portas XOR.

COMPONENTE 14

DETECTOR DE PARIDADE ÍMPAR





PARA QUE SERVE?



- Detecção de Erros: É a forma mais simples de verificação de erros.



- Um "bit de paridade" é adicionado a um dado (ex: um byte na RAM) antes de ser armazenado.



- Quando o dado é lido de volta, a paridade é verificada novamente. Se ela mudou, o sistema sabe que ocorreu um erro (um bit "virou").

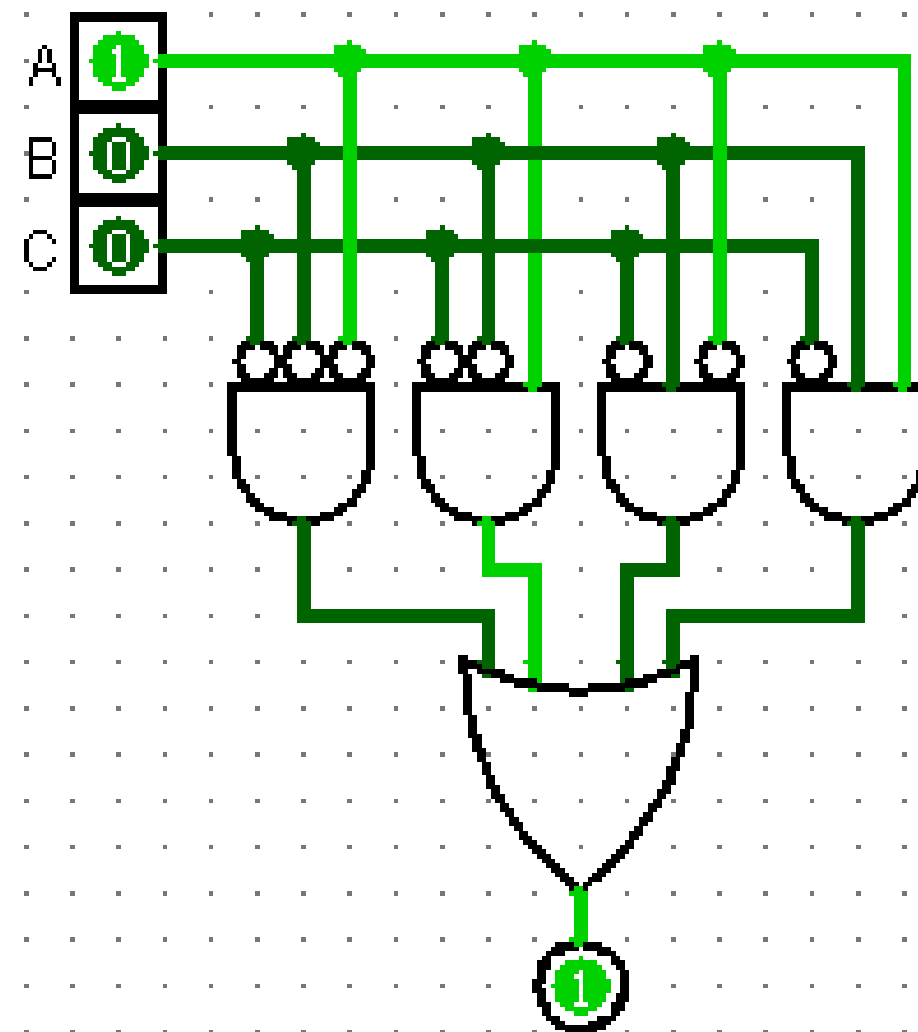


COMPONENTE 15 - OTIMIZAÇÃO COM MAPA DE KARNAUGH



- Não é um componente, mas um método de otimização.
- O Mapa de Karnaugh é uma ferramenta gráfica para simplificar expressões booleanas (lógica de portas).
- O objetivo é pegar um circuito "complexo" e encontrar um circuito "simplificado" que faça exatamente a mesma coisa⁷⁴.
- A expressão é a $S = A'B'C + ABC' + A'BC' + ABC$ versão otimizada da expressão original do circuito.

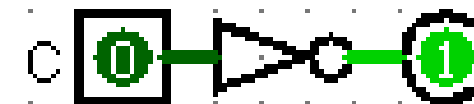
COMPONENTE 15 - OTIMIZAÇÃO COM MAPA DE KARNAUGH

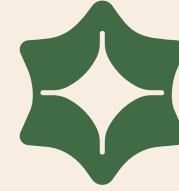


CIRCUITO SIMPLES

$$S = A'B'C' + AB'C' + A'BC' + ABC'$$

CIRCUITO SIMPLIFICADO





PARA QUE SERVE?



- Eficiência: A otimização "reduz o número de componentes necessários".



- Redução de custo: Menos portas lógicas significam um chip menor e mais barato.



- Velocidade e energia: Circuitos mais simples são geralmente mais rápidos e consomem menos energia.



- Manutenção: Um circuito otimizado é mais fácil de entender e manter.

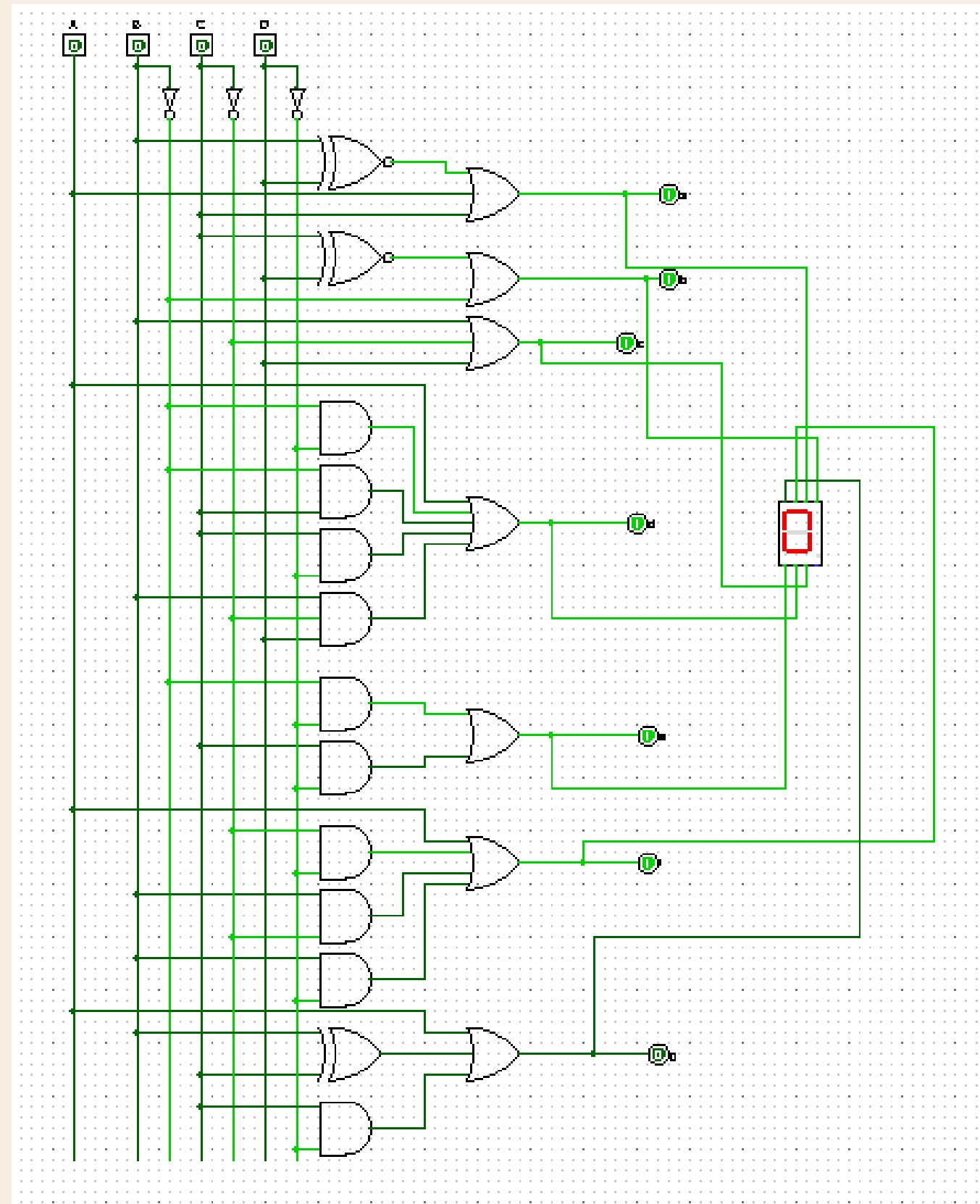


COMPONENTE 16 - DECODIFICADOR DE 7 SEGMENTOS



- É um circuito "tradutor".
- Computadores entendem binário (ex: 0011 para o número 3).
- Nós, humanos, lemos números em displays.
- Este circuito recebe um número binário de 4 bits e "traduz", ativando as saídas corretas (a, b, c, d, g) para acender os segmentos e formar o dígito legível no display.

COMPONENTE 16 - DECODIFICADOR DE 7 SEGMENTOS





PARA QUE SERVE?



- Qualquer dispositivo eletrônico que precise mostrar números para um usuário.



- Exemplos:
 1. Calculadoras
 2. Relógios digitais
 3. Placares esportivos
 4. Displays em fornos de micro-ondas ou painéis de carro.

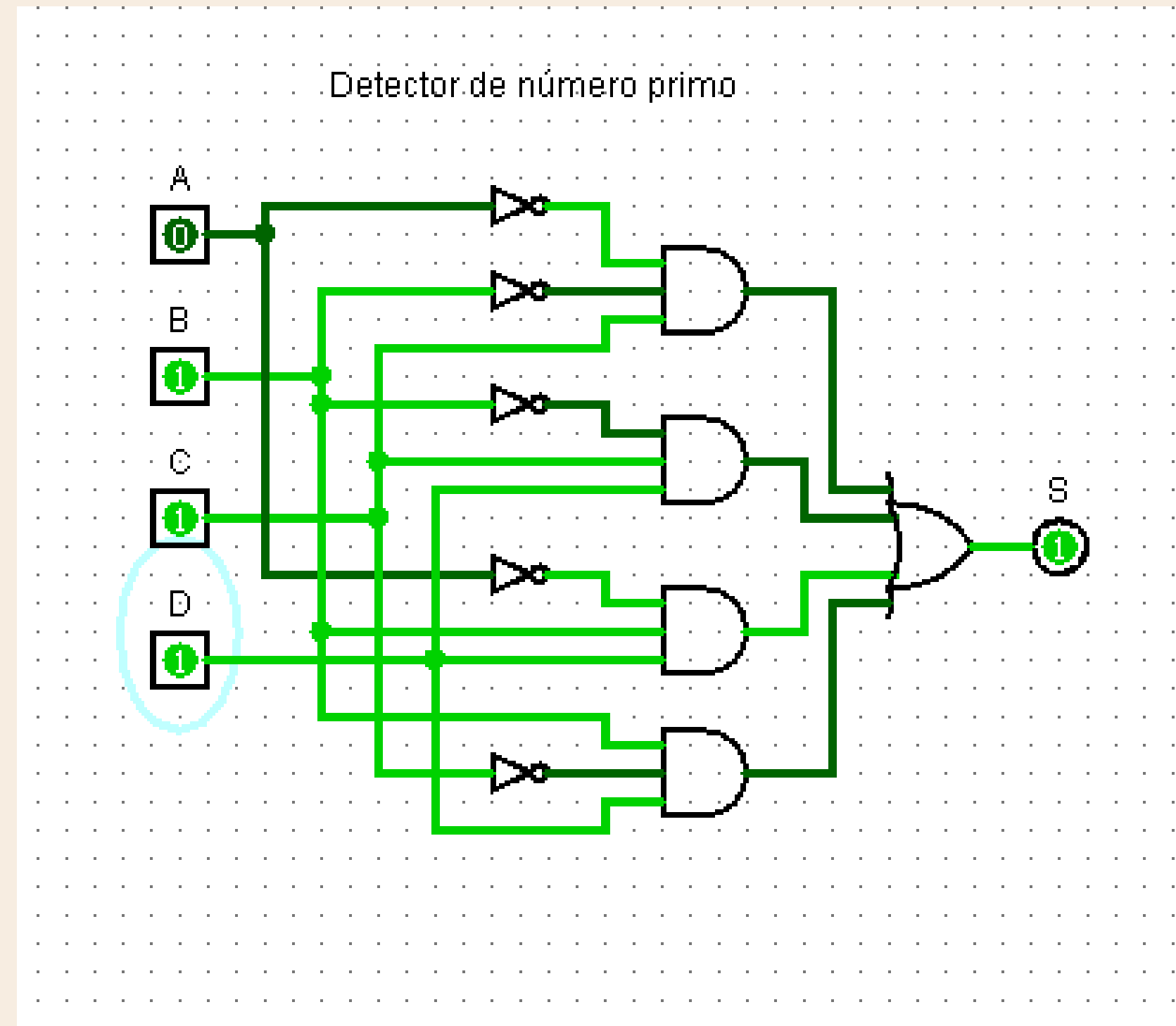


COMPONENTE 17 - DETECTOR DE NÚMERO PRIMO



- Um circuito lógico customizado, construído para uma tarefa muito específica: identificar se um número de 4 bits é primo.
- Ele tem 4 entradas (para o número de 0 a 15) e 1 saída.
- A saída será '1' (Verdadeiro) se o número de entrada for primo.
- A saída será '0' (Falso) para todos os outros números que não forem primos.

COMPONENTE 17 - DETECTOR DE NÚMERO PRIMO





PARA QUE SERVE?



- Este é um exemplo de lógica combinacional personalizada.



- Ele mostra como qualquer "problema" que possa ser definido por uma tabela verdade (como "é primo?") pode ser transformado diretamente em um circuito de hardware.



- Esse tipo de lógica é usado em "ASICs" (Application-Specific Integrated Circuits), que são chips projetados para fazer uma única tarefa de forma muito rápida (ex: minerar criptomoedas, processar vídeo).



OBRIGADO PELA ATENÇÃO!

Hora das perguntas.

Amanda de Brito Barbosa, Fábio Aurélio Barros Alexandre e José Carvalho Neto