

Relatório Técnico

1. Introdução

- **Objetivo:** Explicar que o trabalho visa implementar componentes usando o Logisim e testar suas funcionalidades.
- **Ferramentas:** Breve introdução ao Logisim, destacando que é uma ferramenta de simulação de circuitos digitais.
- **Componentes abordados:** Lista geral dos componentes que serão implementados.

COMPONENTE 01: Flip-Flop Tipo D e Tipo JK

1. Flip-Flop Tipo D

Descrição do Componente

Pinos:

Entrada:

D (Data): Define o valor a ser armazenado (0 ou 1).

CLK (Clock): Pulso de sincronização que aciona a gravação.

Saídas:

A: Estado atual armazenado.

B: Complemento de Q.

Lógica e Funcionamento:

O Flip-Flop tipo D é um registrador que armazena o valor presente na entrada D quando há uma borda de subida (transição de 0 para 1) no sinal de clock CLK. Ele é amplamente utilizado em registros e contadores devido à sua simplicidade e sincronização.

Imagem do Circuito no Logisim

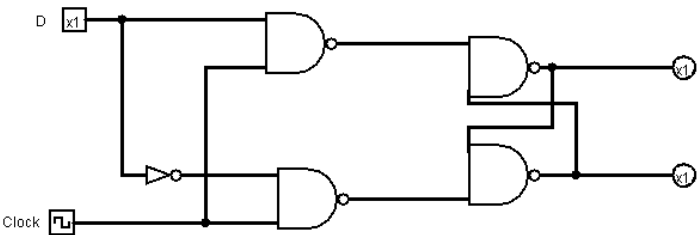


Tabela Verdade do Flip-Flop D:

Clock	D	A	B
0	0	x	x
0	1	x	x
1	0	0	1
1	1	1	0

Descrição dos Testes

Entradas:

Configuramos **D** como 0 ou 1 enquanto acionamos a borda de subida do **CLK**.

Conexões:

D e **CLK** conectados a uma entrada de chave no Logisim.

Resultados Esperados:

Se **CLK** não está ativo, **A** não muda.

Na borda de subida do **CLK**, **A** assume o valor de **D**.

2. Flip-Flop Tipo JK

Descrição do Componente

Pinos:

Entrada:

J: Define a entrada de set (1 para setar **Q** para 1).

K: Define a entrada de reset (1 para resetar **Q** para 0).

CLK: Pulso de sincronização.

Saídas:

Q: Estado atual armazenado.

Q': Complemento de **Q**.

Lógica e Funcionamento:

O Flip-Flop JK é uma evolução do tipo SR, que resolve o estado indeterminado quando ambas as entradas são 1. Ele alterna o estado de **Q** ao receber **J=1** e **K=1** em uma borda de subida do **CLK**.

Imagem do Circuito no Logisim

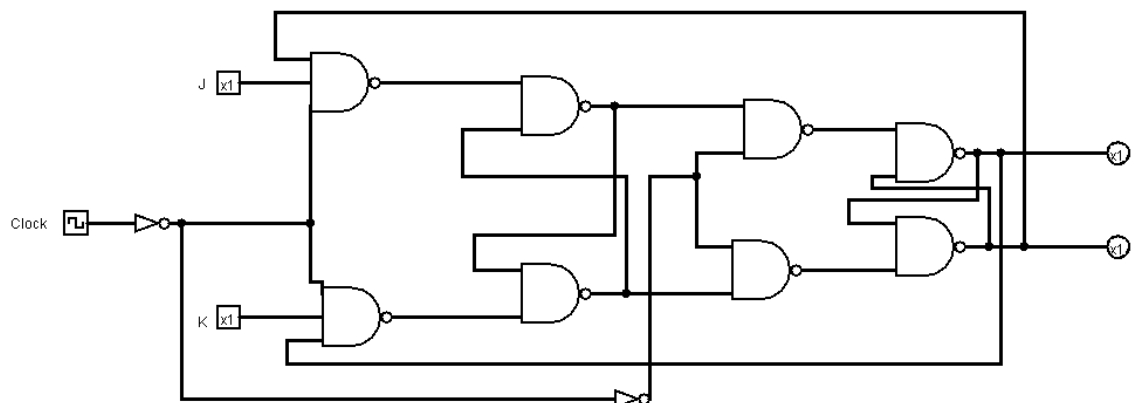


Tabela Verdade do Flip-Flop JK:

Clock	J	K	B	A
1	0	0	Mantém	x
1	0	1	0	1
1	1	0	1	0
1	1	1	0	

Descrição dos Testes

Entradas:

Configuramos **J**, **K** e **CLK** usando chaves no Logisim.

Conexões:

As saídas **A** e **B** conectadas a LEDs para observar os estados.

Resultados Esperados:

Para **J=0**, **K=0**, o estado de **A** não muda.

Para **J=1, K=0**, **A** se torna 1.
Para **J=0, K=1**, **A** se torna 0.
Para **J=1, K=1**, **A** alterna entre 0 e 1

COMPONENTE 02: Multiplexador de Quatro Opções de Entrada

Multiplexador de Quatro Opções

Descrição do Componente

Pinos:

Entradas de Dados:

A, B, C, D: Entradas de dados (0 ou 1).

Entradas de Seleção:

S01, S02: Entradas que determinam qual das quatro entradas de dados será transmitida para a saída.

Saída:

Y: Saída do valor selecionado.

Lógica e Funcionamento:

O multiplexador é um dispositivo combinacional que seleciona uma das várias entradas de dados com base nos valores das entradas de seleção.

As combinações de **S01** e **S02** selecionam qual entrada (**A, B, C, D**) será conectada à saída **Y**:

S01 S02 = 00: Saída **Y = D**.

S01 S02 = 01: Saída **Y = C**.

S01 S02 = 10: Saída **Y = B**.

S01 S02 = 11: Saída **Y = A**.

Imagem do Circuito no Logisim

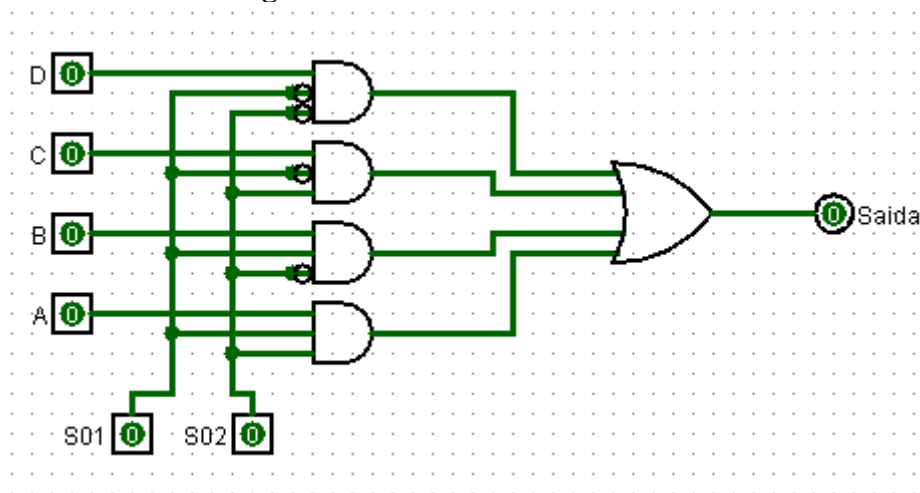


Tabela Verdade

S01	S02	Saída
0	0	D
0	1	C
1	0	B
1	1	A

Descrição dos Testes

Entradas:

Configurar os valores das entradas de dados (**A**, **B**, **C**, **D**) usando chaves no Logisim.

Variar os sinais de seleção **S01** e **S02** para cada combinação (00, 01, 10, 11).

Conexões:

As entradas de dados (**A-D**) conectadas a chaves.

As saídas de seleção (**S01**, **S02**) conectadas a chaves que definem o caminho da seleção.

A saída **Y** conectada a visualizador para indicar o resultado.

Resultados Esperados:

Para **S01=0** e **S02=0**, saída **Y = D**.

Para **S01=0** e **S02=1**, saída **Y = C**.

Para **S01=1** e **S02=0**, saída **Y = B**.

Para **S01=1** e **S02=1**, saída **Y = A**.

COMPONENTE 03: Porta Lógica XOR usando AND, NOT e OR

1. Porta Lógica XOR

Descrição do Componente

Pinos:

Entradas:

A: Primeira entrada binária (0 ou 1).

B: Segunda entrada binária (0 ou 1).

Saída:

Y: Resultado da operação lógica XOR.

Lógica e Funcionamento:

A porta lógica XOR (OU Exclusivo) retorna 1 na saída apenas se **A** e **B** forem diferentes (ou seja, $A \neq B$).

Etapas:

Negar a entrada **B** com uma porta **NOT**, conectar à porta **AND** com a entrada **A**.

Negar a entrada **A** com uma porta **NOT**, conectar à porta **AND** com a entrada **B**.

Os resultados das duas portas **AND** são combinados com uma porta **OR** para produzir a saída **Y**.

Imagem do Circuito no Logisim

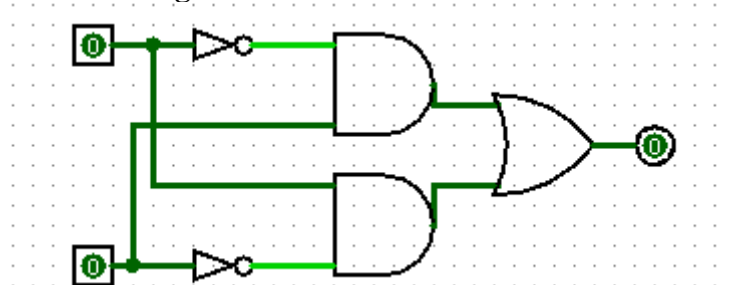


Tabela Verdade da Porta XOR:

A	B	Saída
0	0	0
0	1	1

1	0	1
1	1	1

Descrição dos Testes

Entradas:

As entradas **A** e **B** são configuradas com chaves no Logisim para gerar as combinações possíveis (0 e 1).

Conexões:

Cada entrada é conectada às portas lógicas **NOT**, **AND** e **OR** de acordo com o circuito projetado para implementar a função XOR.

A saída **Y** está conectada a um LED para indicar o valor gerado.

Resultados Esperados:

Quando $A \neq B$, a saída **Y** será 1.

Quando $A = B$, a saída **Y** será 0.

COMPONENTE 04: Somador de 8 Bits com Incremento de 4

1. Somador de 8 Bits

Descrição do Componente

Pinos:

Entradas:

A: Valor inteiro de 8 bits fornecido como entrada.

B: Valor constante de 8 bits igual a 4 (00000100 em binário).

Saídas:

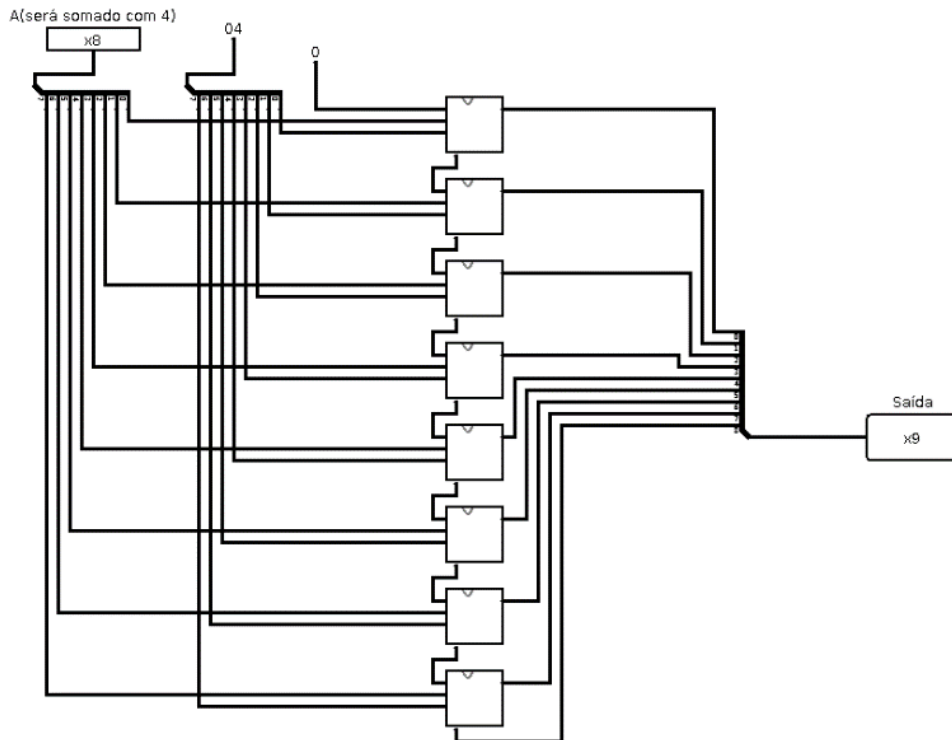
S: Resultado da soma de **A** + 4.

Carry Out: Indica o transporte da operação (se houver).

Lógica e Funcionamento:

O somador de 8 bits realiza a operação aritmética de soma entre dois números binários de 8 bits. Neste caso, uma das entradas (**B**) é constante e vale **4**. O somador é implementado utilizando blocos de somadores de 1 bit (Full Adders) interconectados.

Imagem do Circuito no Logisim



Descrição dos Testes

Entradas:

O valor de **A** é configurado por meio de chaves no Logisim para representar diferentes valores inteiros.

A entrada **B** é conectada a uma constante de 4 no circuito.

Conexões:

A e **B** são conectadas às entradas de um somador de 8 bits.

A saída **S** é conectada a um visualizador de resultado binário.

O **Carry Out** é conectado a um visualizador separado para indicar a ocorrência de transporte.

Resultados Esperados:

Para **A = 0**, a saída **S = 4**.

Para **A = 255**, a saída **S = 3** e **Cout = 1**.

Para valores intermediários, a saída **S** corresponde à soma exata de **A + 4**.

COMPONENTE 05: Memória ROM de 8 Bits

1. Memória ROM (Read-Only Memory)

Descrição do Componente

Pinos:

Entradas:

Endereço (Address): Linha de 3 bits (**A**, **B**, **C**) para acessar uma localização de memória (8 endereços possíveis).

Saídas:

Dado (Data): Linha de 8 bits que contém o dado armazenado na posição de memória selecionada.

Lógica e Funcionamento:

A ROM (Read-Only Memory) é uma memória não volátil usada para armazenar dados ou instruções pré-definidas.

Neste componente, a ROM contém 8 posições de memória, cada uma armazenando um valor de 8 bits.

O endereço fornecido pelas entradas **A**, **B**, **C**, seleciona a linha de memória e a saída correspondente é disponibilizada.

Os valores são gravados previamente e não podem ser alterados durante a execução.

Imagem do Circuito no Logisim

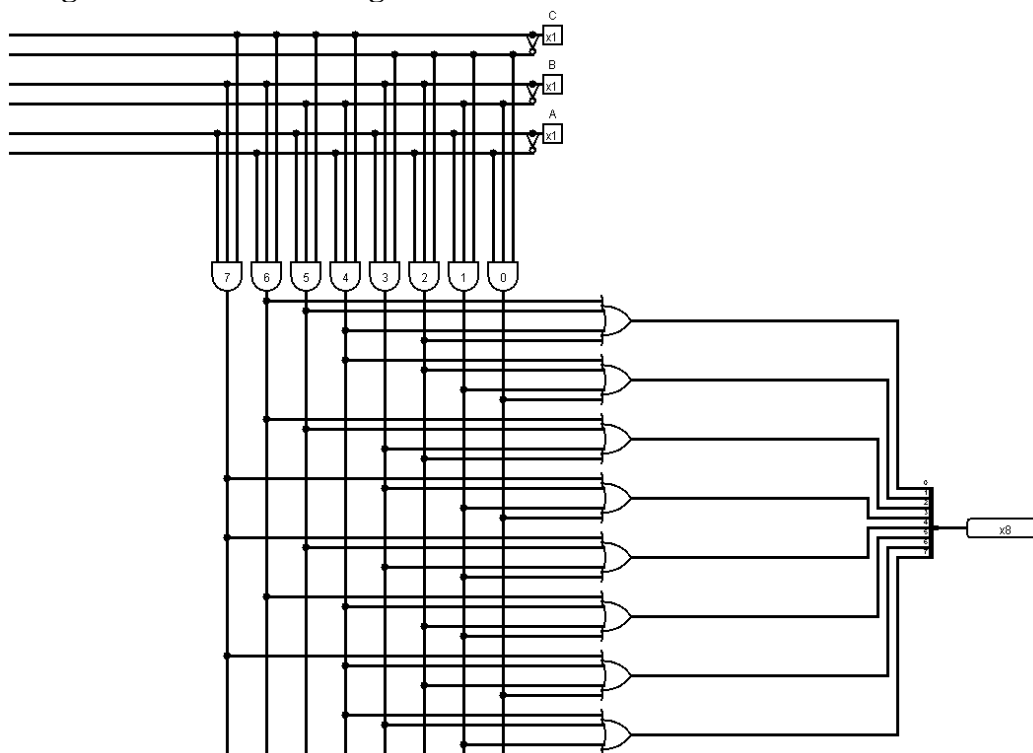


Tabela Verdade da Memória ROM de 8 bits

A	B	C	Saída
0	0	0	11001010
0	0	1	11100011
0	1	0	01100111
0	1	1	00100101
1	0	0	10111010
1	0	1	00010101
1	1	0	10011100
1	1	1	01011000

Descrição dos Testes

Entradas:

Os endereços (**A**, **B**, **C**) são configurados por meio de chaves no Logisim.

Conexões:

As entradas de endereço são conectadas ao bloco de memória ROM.

As saídas de dados são conectadas a um visualizador de números para checar o valor retornado.

Resultados Esperados:

Para cada combinação de **A**, **B**, **C**, o valor correspondente armazenado na ROM aparece nas saídas.

Exemplo:

- Endereço **000** retorna **11001010**
- Endereço **011** retorna **00100101**
- Endereço **111** retorna **01011000**

COMPONENTE 06: Memória RAM de 8 Bits

Memória RAM (Random Access Memory)

Descrição do Componente

Pinos:

Entradas:

Endereços: Linha de 3 bits (**A, B, C**) para acessar uma das 8 posições de memória.

Escrita: Linha de 8 bits usada para gravar um valor na posição de memória selecionada.

Controle:

Write Enable (WE): Ativa a gravação (1 para gravar, 0 para somente leitura).

Chip Enable (CE): Habilita a RAM (1 para ativo, 0 para inativo).

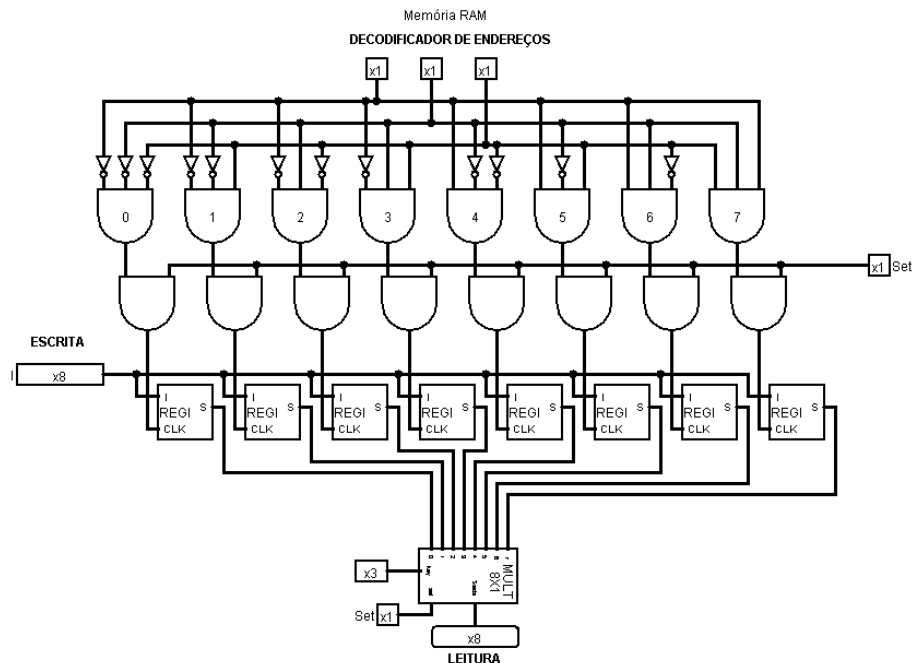
Saídas:

Leitura: Linha de 8 bits que retorna o valor armazenado na posição selecionada.

Lógica e Funcionamento:

A RAM é uma memória volátil que permite leitura e escrita em qualquer posição de memória.

Imagem do Circuito no Logisim



Descrição dos Testes

Entradas:

Os endereços são definidos por chaves no Logisim.

O valor da leitura é configurado com chaves para entrada de dados.

Os sinais de controle (**CE** e **WE**) são alternados conforme a operação desejada.

Conexões:

Leitura, Escrita, CE, e WE conectados ao módulo de RAM.

A saída é conectada em um visualizador de dados binários.

Resultados Esperados:

Quando **CE = 0**, nenhuma operação é realizada e a saída permanece desconectada.

Quando **CE = 1** e **WE = 1**, o valor fornecido em **Escrita** é armazenado na posição especificada pelo endereço.

Quando **CE = 1** e **WE = 0**, o valor armazenado na posição especificada é retornado na saída **Leitura**.

COMPONENTE 07: Banco de Registradores de 8 Bits

Banco de Registradores de 8 Bits

Descrição do Componente

Pinos:

Entradas:

Endereço: Linha de 4 bits (**Select**) que seleciona um dos 8 registradores.

Dado de Entrada: Linha de 8 bits (**Entrada**) que contém o valor a ser gravado no registrador selecionado.

Controle:

Leitura: Habilita a gravação (1 para gravar, 0 para leitura).

Escrever: Habilita o banco de registradores (1 para ativar, 0 para desativar).

Saídas:

Saída: Linha de 8 bits que retorna o valor armazenado no registrador selecionado.

Lógica e Funcionamento:

Um banco de registradores é um conjunto de registradores interconectados que armazenam múltiplos valores simultaneamente. Cada registrador pode ser acessado e modificado independentemente.

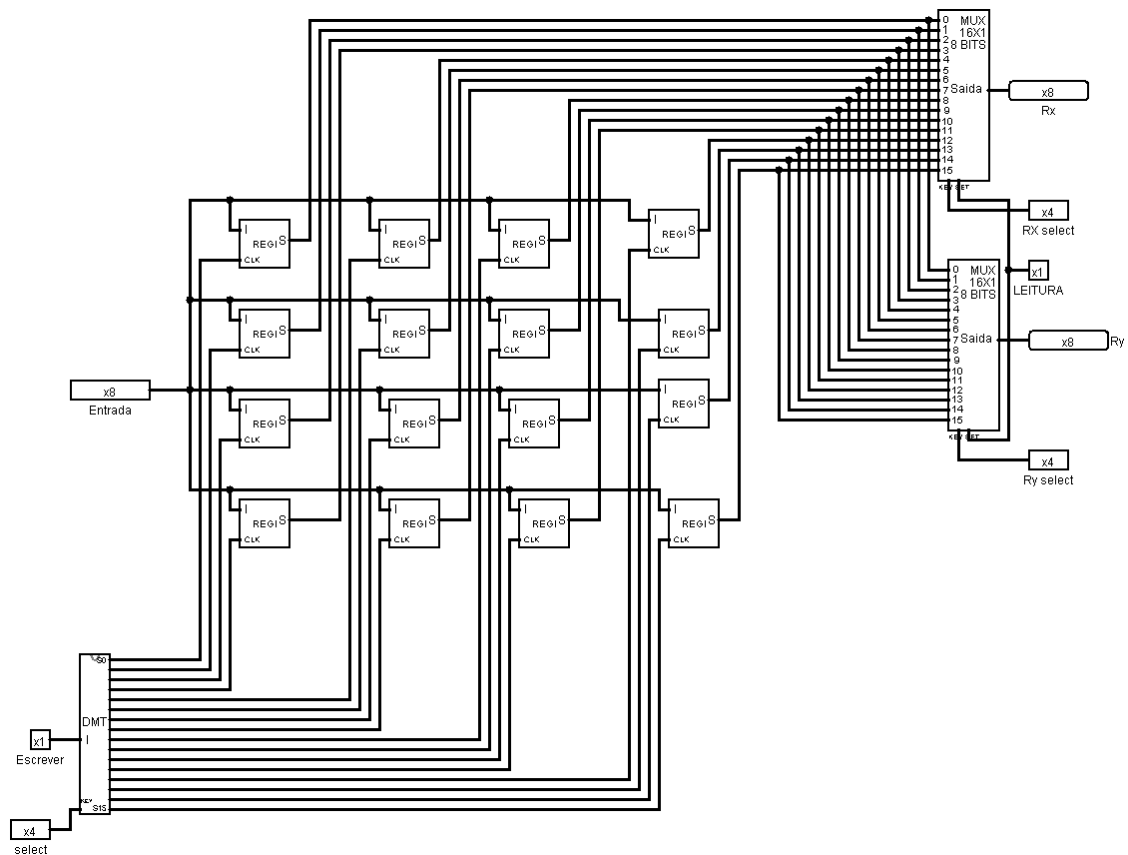
Endereço: O sinal de 3 bits **Select** seleciona um dos 8 registradores disponíveis.

Gravação: Quando **Escrever = 1** e **Leitura = 1**, o valor em **Saída** é gravado no registrador selecionado.

Leitura: Quando **Leitura = 1** e **Escrita = 0**, o valor armazenado no registrador selecionado é enviado para **Saída**.

Inatividade: Se **Leitura = 0**, o banco de registradores é desativado e as saídas ficam desconectadas.

Imagem do Circuito no Logisim



Descrição dos Testes

Entradas:

Os endereços (**Select**) são configurados para selecionar o registrador desejado.

O valor de **Saída** é configurado por chaves para fornecer dados a serem armazenados nos registradores.

Os sinais de controle (**Leitura e Escrita**) controlam a operação de leitura ou gravação.

Conexões:

As entradas de endereço e dados são conectadas ao banco de registradores.

As saídas de dados (Saída) são conectadas a LEDs para visualizar o valor armazenado no registrador selecionado.

Resultados Esperados:

Para **Leitura = 0**, nenhuma operação é realizada e as saídas ficam desconectadas.

Para **Leitura = 1** e **Escrita = 1**, o valor em **Saída** é gravado no registrador selecionado pelo endereço **Select**

Para **Leitura = 1** e **Escrita = 0**, o valor armazenado no registrador selecionado é lido e enviado para **Saída**.

COMPONENTE 08: Somador de 8 Bits

Somador de 8 Bits

Descrição do Componente

Pinos:

Entradas:

A: Valor de 8 bits (primeiro número) a ser somado.

B: Valor de 8 bits (segundo número) a ser somado.

Carry In: Entrada de carry (transportar) de uma operação anterior.

Saídas:

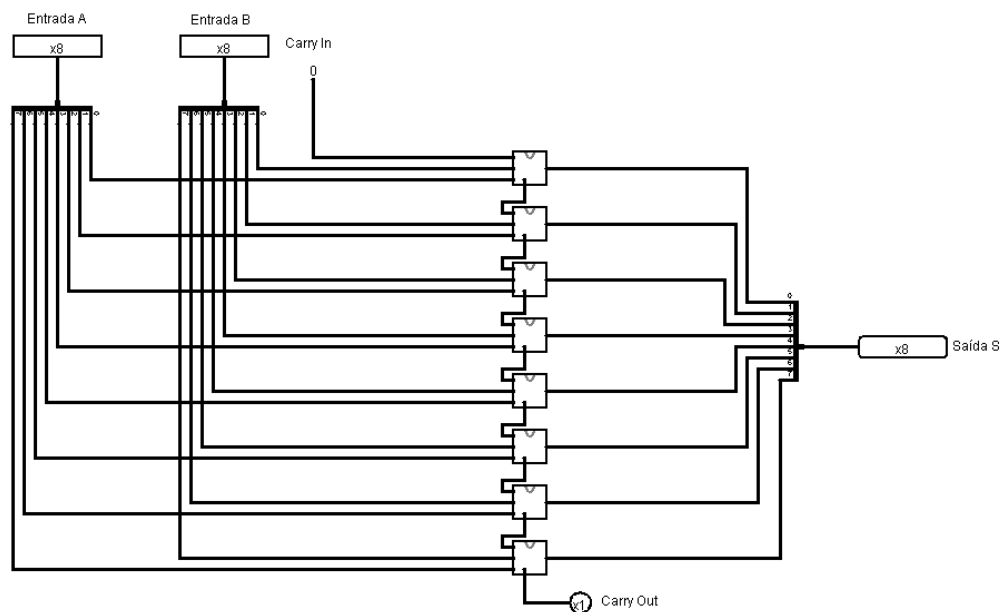
S: Resultado da soma de A e B, com carry.

Carry Out: Carregamento de carry para a próxima operação, caso exista.

Lógica e Funcionamento:

O somador de 8 bits realiza a soma aritmética de dois números binários de 8 bits, levando em consideração a propagação de carry. A operação é realizada bit a bit, da direita para a esquerda, utilizando somadores completos (Full Adders), onde cada somador é responsável pela adição de um bit de cada número e o carry anterior. O carry da soma de cada bit é passado para o próximo bit à esquerda.

Imagem do Circuito no Logisim



Testes Realizados:

Teste 1: Somar **A = 00000000** e **B = 00000000**, esperar que o resultado seja **S = 00000000** e **Carry Out = 0**.

Teste 2: Somar **A = 11111111** e **B = 00000001**, verificar se o **Carry Out** é gerado corretamente e se o resultado é **S = 00000000**.

Teste 3: Somar **A = 11111111**, **B = 11111111**, com **Carry In = 1**, verificar o carry e o resultado da soma.

Descrição dos Testes

Entradas:

As entradas **A** e **B** são configuradas com valores binários por meio de chaves no Logisim.

O **Carry In** é configurado para 0 ou 1, dependendo da operação de carry necessária.

Conexões:

As entradas de **A**, **B** e **Carry In** são conectadas ao circuito de somador de 8 bits.

As saídas **S** e **Carry Out** são conectadas a um visualizador de bit para ver o valor da soma e o carry gerado.

Resultados Esperados:

A soma de **A** e **B** gera **S**, com o carry propagado, caso exista.

Quando **Carry In** = **0**, o resultado será simplesmente a soma dos bits correspondentes. Quando **Carry In** = **1**, será somado um bit extra de carry.

O carry final **Carry Out** será 1 se houver transporte de carry da operação final.

COMPONENTE 09: Detector de Sequência Binária para "101"

Detector de Sequência Binária para "101"

Descrição do Componente

Pinos:

Entradas:

Entrada de Dados: Recebe o fluxo de bits binários de entrada **A**, sendo processado bit a bit.

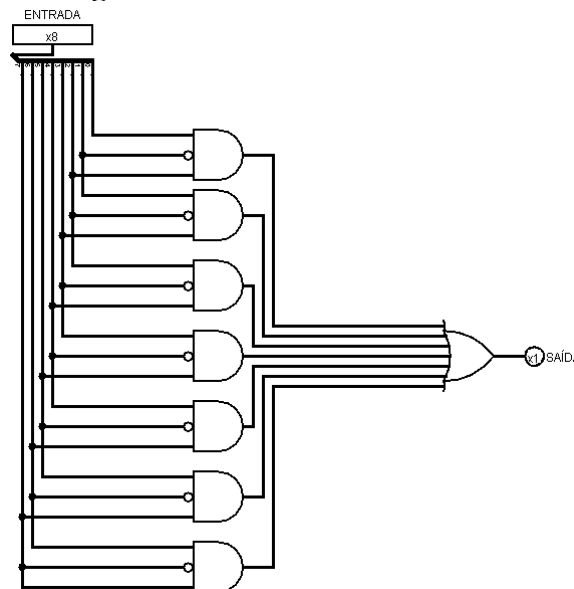
Saídas:

Saída Detectada: Indica a detecção da sequência "101" com um pulso lógico alto (1).

Lógica e Funcionamento

Este detector verifica se a sequência "101" está presente em uma palavra de 8 bits fornecida como entrada **A** e ativa a saída **S** caso a sequência seja encontrada. A análise é feita diretamente sobre os 8 bits em paralelo, sem necessidade de clock ou estados intermediários.

Imagem do Circuito no Logisim



Descrição dos Testes

Entradas:

Entrada **A**: Linha de 8 bits, fornecida via chaves de entrada no Logisim. A configuração dessas chaves simula diferentes valores de 8 bits para testar a detecção da sequência "101".

Saídas:

Dado de Saída **S**: Linha de 1 bit que indica se a sequência "101" foi detectada ou não.

Descrição dos Testes

Teste 1: Quando a entrada A=00000101, A = 00000101, A=00000101, a saída S é 1.

Teste 2: Quando a entrada A=00001010, A = 00001010, A=00001010 a saída S é 1.

Teste 3: Quando a entrada A=10101010A = 10101010A=10101010 então a saída S é 1.

Teste 4: Quando a entrada A=11001010A = 11001010A=11001010, não há ocorrência da sequência "101", portanto, a saída S é 0.

Teste 5: Quando a entrada A=00000000, A = 00000000, A=00000000, não há sequência "101", então a saída S é 0.

Teste 6: Quando a entrada A=11111111, A = 11111111, A=11111111, a sequência "101" não é encontrada, então a saída S é 0.

Teste 7: Quando a entrada A=10110101, A = 10110101, A=10110101, a sequência "101" ocorre, ativando S =1.

Teste 8: Quando a entrada A=01101010, A = 01101010, A=01101010, não há sequência "101", então a saída S é 0.

Teste 9: Quando a entrada A=10111000, A = 10111000, A=10111000, a sequência "101" ocorre, ativando S =1.

Teste 10: Quando a entrada A=01010101, A = 01010101, A=01010101, a sequência "101" ocorre, ativando S =1.

Conclusão

O detector de sequência binária para "101" foi projetado e implementado com sucesso. Ele identifica a sequência desejada com precisão e pode ser utilizado em aplicações como sistemas de comunicação e análise de dados binários. Os testes realizados confirmam a funcionalidade do circuito em diferentes cenários, incluindo a detecção de sobreposições.

COMPONENTE 10: ULA de 8 Bits com Operações Diversas

Descrição do Componente

Pinos

Entradas

Operandos:

A: Primeiro operando de 8 bits.

B: Segundo operando de 8 bits.

Seletor: Linha de 4 bits que define a operação a ser realizada pela ULA, conforme a tabela a seguir:

Operação	Sequência
AND	0000
OR	0001
NOT A	0010
NOT B	0011
NOR	0100
NAND	0101

XOR	0110
SHIFTE A	0111
SHIFTE B	1010
SHIFTD A	1001
SHIFTD B	1101
SUM	1011
SUB	1100

Saídas

Saída: O resultado da operação selecionada.

Status:

Carry: Indica um carry (transbordo) em operações aritméticas.

Negative: Indica se o resultado é negativo (bit mais significativo igual a 1).

Lógica e Funcionamento

A ULA opera com base nos sinais de controle fornecidos pela entrada **Seletor**. Cada operação é executada conforme descrito a seguir:

Operações Lógicas:

AND: Realiza uma conjunção lógica bit a bit entre A e B.

OR: Realiza uma disjunção lógica bit a bit entre A e B.

NOT: Calcula o complemento bit a bit de A.

NOR: Realiza a operação OR entre A e B, invertendo o resultado.

NAND: Realiza a operação AND entre A e B, invertendo o resultado.

XOR: Calcula a disjunção exclusiva bit a bit entre A e B.

Operações de Deslocamento:

SHIFT LEFT: Desloca todos os bits de da entrada (A ou B) duas posições para a esquerda, inserindo zeros nas posições vazias.

SHIFT RIGHT: Desloca todos os bits de (A ou B) duas posições para a direita, preenchendo as posições vazias com zeros.

Operações Aritméticas:

Soma: Soma os valores de A e B, gerando um resultado. O bit de Carry é ativado se houver transbordo.

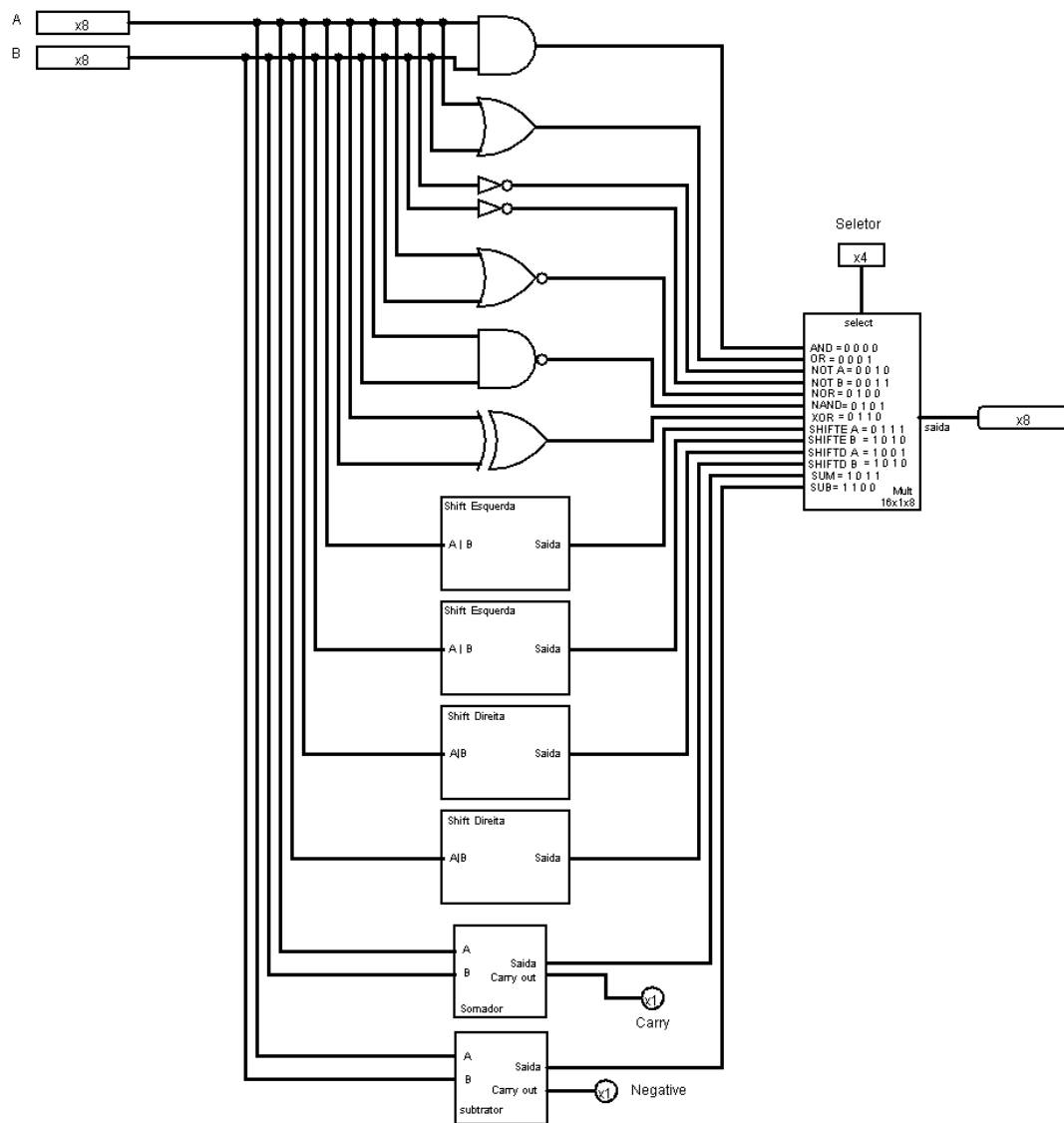
Subtração: Subtrai B de A (Resultado=A-B).

Flags:

Carry: Indica transbordo em operações de soma ou subtração.

Negative: É ativado se o bit mais significativo for 1.

Imagem do Circuito no Logisim



Descrição dos Testes

Entradas:

A: Configurado manualmente por meio de chaves no Logisim.

B: Configurado manualmente por meio de chaves no Logisim.

Seleto: Configurado para selecionar as operações suportadas pela ULA.

Conexões:

As entradas A e B são conectadas às chaves.

A Saída é conectada a um visualizador de resultado.

Resultados Esperados:

Para cada operação definida no **Seleto**, o valor correto é exibido na **Saída**.

Exemplo de Teste:

Configuração:

A=00011000

B=00000111

Op=1000(Soma)

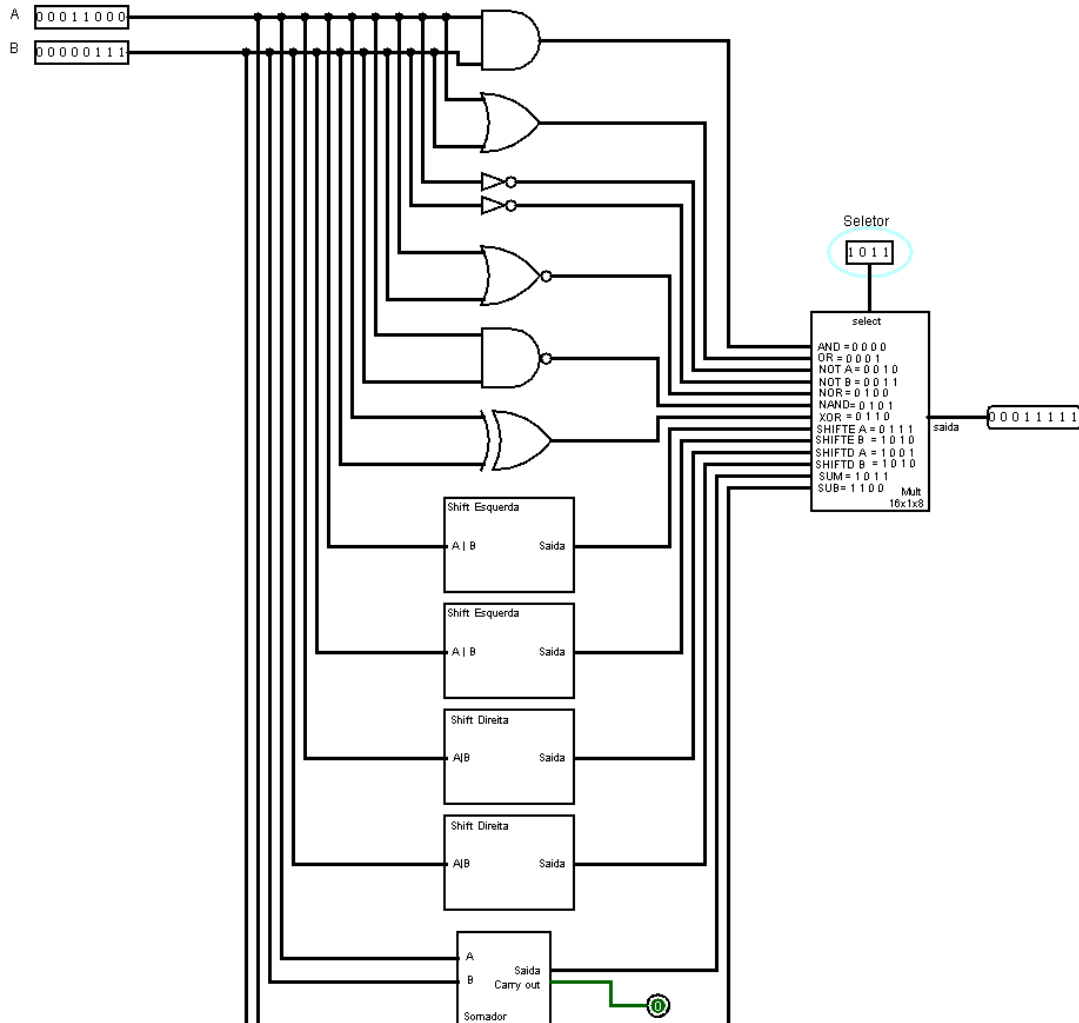
Resultado Esperado:

R=00111111

Carry=0

Negativo=0

Imagem do teste



COMPONENTE 11: Extensor de Sinal de 4 Bits para 8 Bits

Extensor de Sinal de 4 Bits para 8 Bits

Descrição do Componente

Pinos

Entradas:

Entrada de Dados: Linha de 4 bits contendo o número binário de entrada que será estendido para 8 bits.

Saídas:

Saída de Dados: Linha de 8 bits que contém o valor resultante após a extensão do sinal.

Lógica e Funcionamento

O bit mais significativo da entrada é considerado o bit de sinal do número de 4 bits.

Para números positivos, os 4 bits mais significativos da saída são preenchidos com 000.

Para números negativos, os 4 bits mais significativos da saída são preenchidos com 111.

Os 4 bits menos significativos da saída são uma cópia exata da entrada.

Preservação do Valor:

Essa lógica assegura que o valor numérico representado no formato de complemento de dois é mantido corretamente ao ser convertido para 8 bits.

Imagem do Circuito no Logisim



Descrição dos Testes

Entradas:

Dados de Entrada: Configurados com valores binários de 4 bits que incluem números positivos e negativos.

Conexões:

A entrada é conectada ao circuito de extensão de sinal.

A saída é conectada a um display para visualizar os resultados.

Resultados Esperados:

Para a entrada sendo números positivos, os bits mais significativos da saída devem ser 000.

Exemplo: Entrada = 0101A -> Saída = 00000101.

Para entrada = 1 (números negativos), os bits mais significativos da saída devem ser 111.

Exemplo: Entrada = 1101 -> Saída = 11111101

O valor numérico na saída deve corresponder exatamente ao valor da entrada no formato de complemento de dois, mas agora representado com 8 bits.

COMPONENTE 12: Implementação de uma Máquina de Estados com Portas Lógicas

1. Máquina de Estados de 4 Estados

Descrição do Componente

Pinos:

Entradas de Dados:

XXX: Entrada de controle (0 ou 1), que influencia as transições de estado.

Clock: Sinal de clock, responsável pela sincronização das transições entre os estados.

Set: Entrada de controle para forçar a máquina a iniciar em um estado predefinido.

Reset: Entrada de controle para forçar a máquina a voltar para o estado inicial AAA.

Saídas:

X1 e X2: Saídas que indicam o estado atual da máquina. A codificação dos estados é a seguinte:

A=00

B=01

C=10

D=11

Lógica e Funcionamento

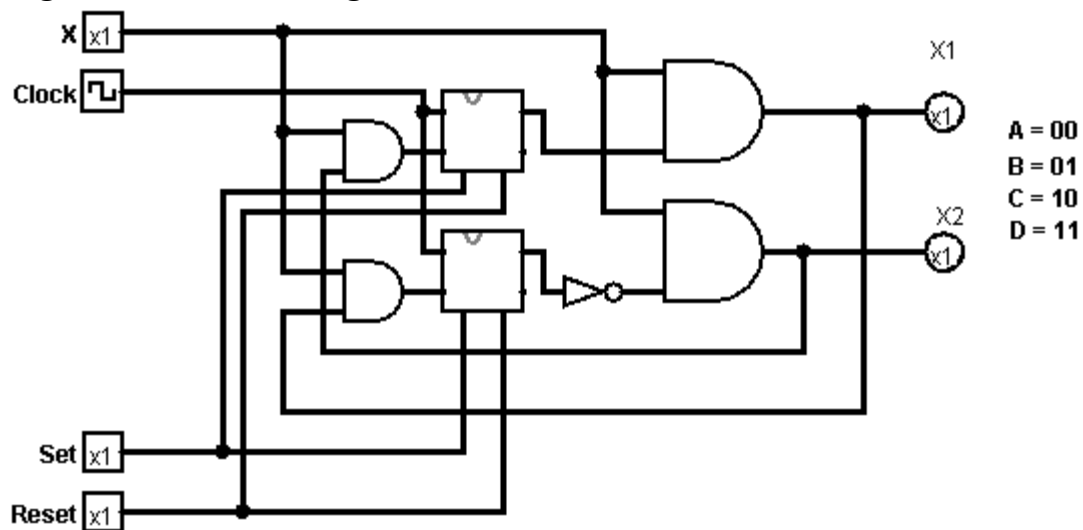
A máquina de estados possui 4 estados possíveis (**A,B,C,D**), e as transições entre esses estados são controladas pela entrada **X** e pelos sinais de controle set e reset. O sinal de clock sincroniza as transições. As saídas **X1** e **X2** indicam o estado atual da máquina.

As combinações de **X1** e **X2** determinam o estado atual da máquina:

Estado Atual	X1X2
A	00
B	01
C	10
D	11

A lógica de transição entre os estados depende da entrada **X** e do valor atual de **X1** e **X2**.

Imagem do Circuito no Logisim



COMPONENTE 13: Contador Síncrono

Descrição do Componente

Pinos:

Entradas:

Clock: Pulso de clock que controla as transições entre os estados do contador.

Ligar: Habilita ou desabilita a contagem (1 para habilitar, 0 para desabilitar).

Saídas:

Contador: Conjunto de saídas (**Q0, Q1, ..., Qn-1**) que representam o valor binário atual do contador.

Lógica e Funcionamento

O contador síncrono é um circuito sequencial que gera uma sequência de estados (em binário) de forma sincronizada com um sinal de clock. Diferentemente dos contadores assíncronos, as transições nos flip-flops ocorrem simultaneamente, minimizando problemas de atraso de propagação.

Funcionamento Geral:

A cada pulso de **Clock**, o contador incrementa ou decrementa o valor armazenado (dependendo da configuração).

Se o contador atingir o valor máximo (por exemplo, 111 para um contador de 3 bits), ele volta ao estado inicial (000) no próximo pulso.

Controle com Entradas:

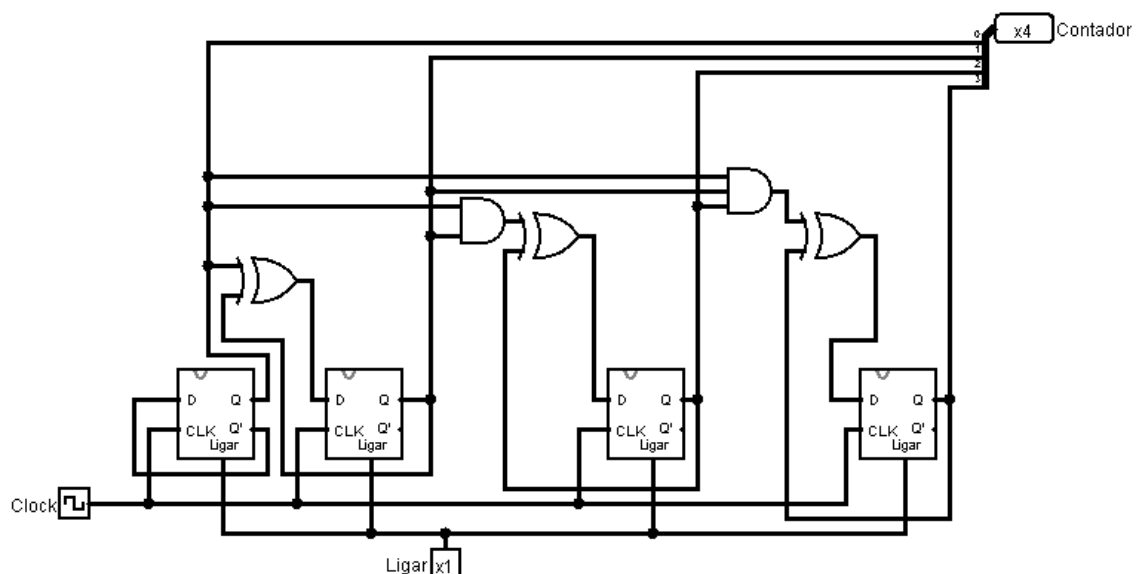
Ligar: Quando em 1, permite a contagem. Quando em 0, o contador mantém o estado atual, independentemente do clock.

Estrutura Interna:

Composto por flip-flops conectados de forma que o estado atual seja atualizado em sincronia com o clock.

Portas lógicas combinacionais determinam as condições para a mudança de estado.

Imagem do Circuito no Logisim



Descrição dos Testes

Entradas:

O sinal de Clock é fornecido como um pulso contínuo para testar as transições do contador.

O sinal **Ligar** é alternado entre 0 e 1 para testar se a contagem para corretamente.

Conexões:

As saídas do contador (**Q0, Q1, ..., Qn**) são conectadas a um visualizador para visualização dos estados.

O sinal de clock é conectado aos flip-flops, garantindo transições sincronizadas.

Resultados Esperados:

Quando **Ligar** = 1, o contador percorre a sequência binária incremental em cada pulso de clock.

Quando **Ligar** = 0, o contador mantém o estado atual, independentemente do clock.

COMPONENTE 14: Detector de Paridade Ímpar

Descrição do Componente

Pinos:

Entradas:

Entrada: Conjunto de 4 bits (entradas binárias) que representam o número a ser analisado.

Saídas:

Saída: Indica se o número de bits iguais a 1 na entrada é ímpar.

Saída: = 1: Número ímpar.

Saída: = 0: Número par.

Imagem do Circuito no Logisim

Descrição dos Testes

Entradas:

A **Entrada** é configurada por uma chave de 4 bits no Logisim para fornecer diferentes combinações de bits.

Conexões:

As entradas binárias são conectadas ao circuito lógico composto por portas AND, OR e NOT, implementando a lógica de paridade ímpar.

A **Saída** é conectada a um visualizador para visualização.

Resultados Esperados:

Quando o número de bits em 1 na entrada é ímpar, o visualizador correspondente a **Saída** acende (1).

Quando o número de bits em 1 na entrada é par, o visualizador correspondente a **Saída** permanece apagada (0).

COMPONENTE 15: Resolva um problema de otimização lógica utilizando mapas de Karnaugh e implemente o circuito otimizado

COMPONENTE 16: Decodificador de 7 Segmentos: Projete um circuito que converta um número binário de 4 bits para os sinais necessários para acionar um display de 7 segmentos (formato hexadecimal).

Descrição do Componente

Pinos:

Entradas:

4 bits de entrada, representados por: **A, B, C, D** (sendo A o bit mais significativo e D ao menos significativo).

Saídas:

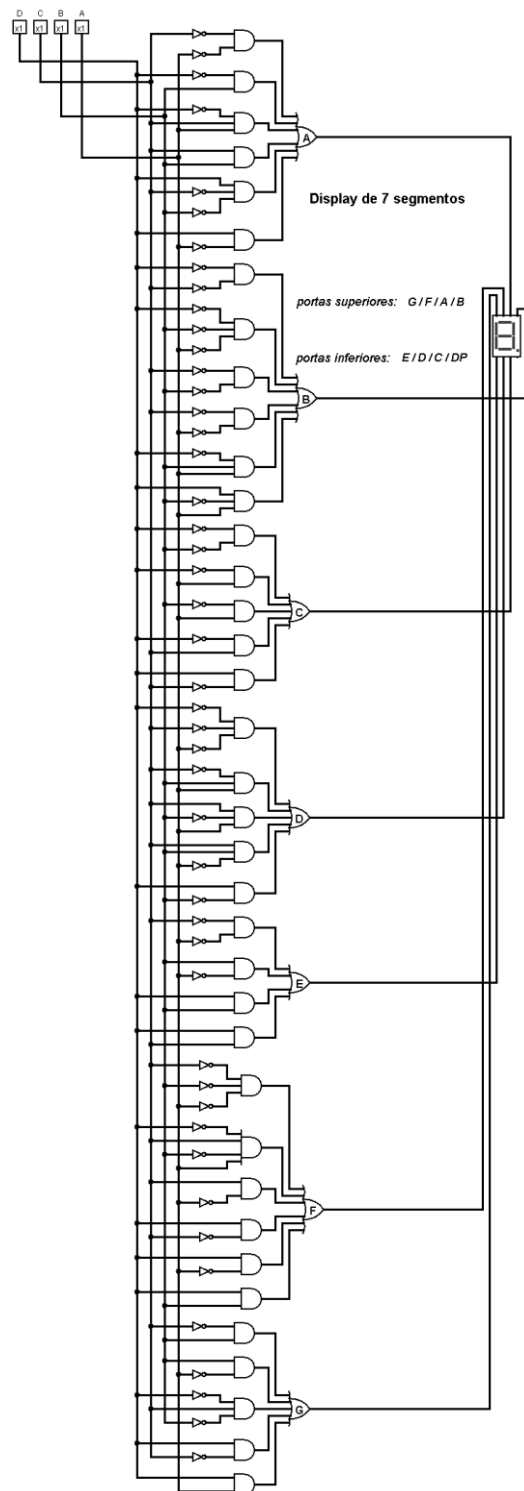
Display de 7 segmentos

Lógica e Funcionamento

O decodificador de 7 segmentos recebe um número binário de 4 bits como entrada e aciona os segmentos apropriados de um display de 7 segmentos. O número binário de 4 bits pode representar valores hexadecimais de 0 a 15 (de 0000 a 1111). Cada combinação binária aciona os segmentos correspondentes para mostrar o número hexadecimal na tela.

Binário				Hex	A	B	C	D	E	F	G
A	B	C	D								
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	0	1	1	0	0	0	0
0	0	1	0	2	1	1	0	1	1	0	1
0	0	1	1	3	1	1	1	1	0	0	1
0	1	0	0	4	0	1	1	0	0	1	1
0	1	0	1	5	1	0	1	1	0	1	1
0	1	1	0	6	1	0	1	1	1	1	1
0	1	1	1	7	1	1	1	0	0	0	0
1	0	0	0	8	1	1	1	1	1	1	1
1	0	0	1	9	1	1	1	1	0	1	1
1	0	1	0	A	1	1	1	0	1	1	1
1	0	1	1	B	0	0	1	1	1	1	1
1	1	0	0	C	1	0	0	1	1	1	0
1	1	0	1	D	0	1	1	1	1	0	1
1	1	1	0	E	1	0	0	1	1	1	1
1	1	1	1	F	1	0	0	0	1	1	1

Imagem do Circuito no Logisim



Descrição dos Testes

Entradas:

Configurar os valores das entradas binárias (A, B, C, D) representando números de 0 a 15 (binários de 0000 a 1111).

Conexões:

As entradas de dados (A-D) são conectadas a chaves para fornecer os valores binários de 4 bits para o display de 7 segmentos.

Resultados Esperados:

Para a entrada 0000 (valor binário 0), os segmentos devem acender de acordo com a representação do número 0 em hexadecimal (1111110).

Para a entrada 0001 (valor binário 1), os segmentos devem acender conforme a representação do número 1 em hexadecimal (0110000).

E assim por diante, para os valores binários de 0000 a 1111, os segmentos devem acender conforme a tabela de correspondência.

COMPONENTE 17: Detector de Número Primo

Descrição do Componente

Pinos:

Entradas:

4 bits de entrada, representados por: **A, B, C, D** (sendo A o bit mais significativo e D ao menos significativo).

Saída:

Saída: Saída que indica se o número representado pelos 4 bits é primo. A saída será 1 se o número for primo e 0 caso contrário.

Lógica e Funcionamento

Os números representados por 4 bits vão de 0 a 15 (em binário: de 0000 a 1111).

O objetivo do circuito é detectar se o número binário corresponde a um número primo. Os números primos entre 0 e 15 são: 2, 3, 5, 7, 11, 13.

O circuito deve gerar uma saída 1 para esses números e 0 para os outros. Isso pode ser feito com a ajuda de portas lógicas após a implementação de um mapa de Karnaugh.

Tabela Verdade

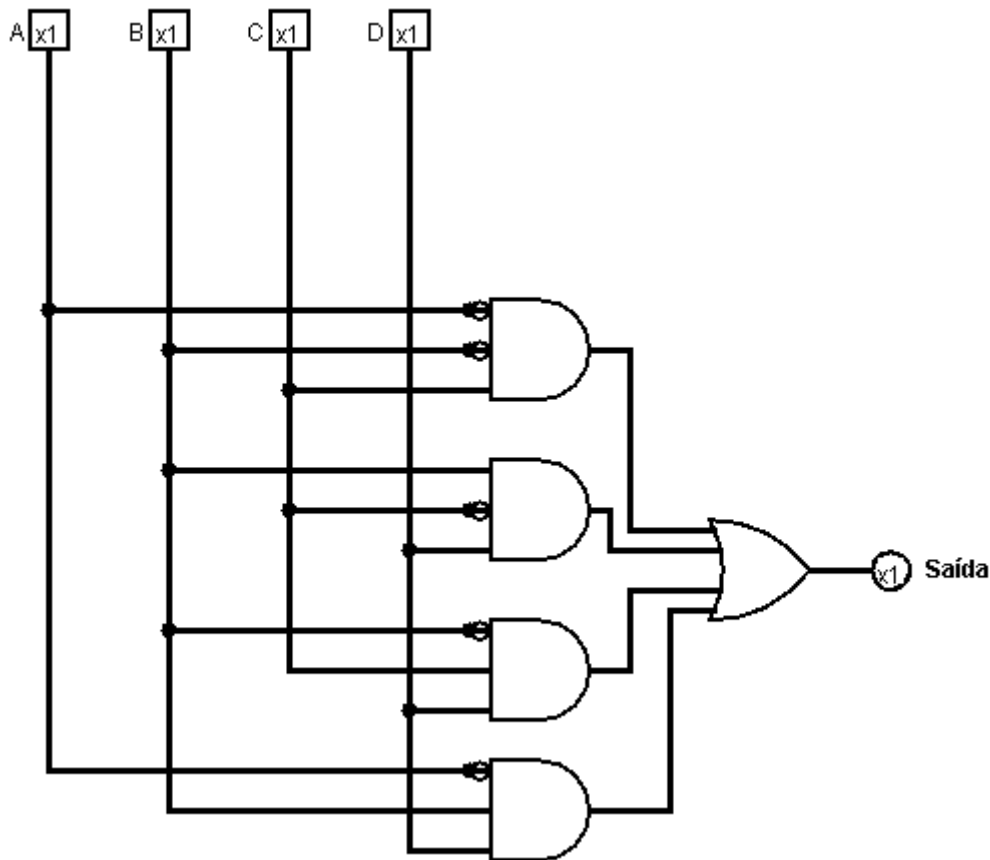
Aqui está a tabela verdade que descreve a relação entre os números binários de 4 bits e a saída (**Saída:**):

A	B	C	D	Decimal	Saída
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	2	1
0	0	1	1	3	1
0	1	0	0	4	0
0	1	0	1	5	1
0	1	1	0	6	0
0	1	1	1	7	1
1	0	0	0	8	0
1	0	0	1	9	0
1	0	1	0	10	0
1	0	1	1	11	1
1	1	0	0	12	0
1	1	0	1	13	1
1	1	1	0	14	0

1	1	1	1	15	0
---	---	---	---	----	---

Nota: Apenas os números 2, 3, 5, 7, 11 e 13 são primos.

Imagem do Circuito no Logisim



Descrição dos Testes

Entradas:

Configurar os valores de entrada binária (A, B, C, D) representando números de 0 a 15.

Conexões:

As entradas de dados (A-D) são conectadas a chaves para fornecer os valores binários de 4 bits.

A saída (**Saída**) será conectada a um indicador visual para mostrar se o número é primo (**Saída = 1**) ou não (**Saída = 0**).

Procedimento de Teste:

Defina as entradas A, B, C, D para representar cada valor binário de 0000 a 1111 (de 0 a 15 em decimal).

Observe a saída S para cada valor binário e verifique se ela corresponde ao comportamento esperado de números primos (apenas 2, 3, 5, 7, 11, 13 devem gerar **Saída = 1**).

Para números não primos, a **Saída** deve ser 0.

Resultados Esperados:

Para a entrada 0000 (valor binário 0), a **Saída** deve ser 0.

Para a entrada 0010 (valor binário 2), a **Saída** deve ser 1.
Para a entrada 0101 (valor binário 5), a **Saída** deve ser 1.
Para a entrada 0111 (valor binário 7), a **Saída** deve ser 1.
Para a entrada 1011 (valor binário 11), a **Saída** deve ser 1.
Para a entrada 1101 (valor binário 13), a **Saída** deve ser 1.
Para todas as outras entradas, a **Saída** deve ser 0.

3. Conclusão

- Resuma os resultados obtidos, destacando a precisão dos componentes implementados e testados.
 - Comente a relevância do aprendizado na prática de circuitos digitais.
-

4. Referências

- Cite quaisquer materiais de estudo ou manuais utilizados, incluindo documentação do Logisim e apostilas de circuitos digitais.