**UNIVERSIDADE FEDERAL DA FRONTEIRA SUL**

**CAMPUS CHAPECÓ**

**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**GABRIEL GOIS**

**FERNANDO VIDIMAR**

**TRABALHO FINAL DE CIRCUITOS DIGITAIS:**

CRIAÇÃO DE UM CIRCUITO CAPAZ DE RESOLVER EQUAÇÕES DE PRIMEIRO E SEGUNDO GRAU UTILIZANDO BANCO DE REGISTRADORES E ULA

**CHAPECÓ**

**2023**

**GABRIEL GOIS**

**FERNANDO VIDIMAR**

**TRABALHO FINAL DE CIRCUITOS DIGITAIS:**

CRIAÇÃO DE UM CIRCUITO CAPAZ DE RESOLVER EQUAÇÕES DE PRIMEIRO E SEGUNDO GRAU UTILIZANDO BANCO DE REGISTRADORES E ULA

Trabalho apresentado ao Curso Ciência da computação da Universidade Federal da Fronteira Sul (UFFS) como requisito parcial para aprovação na disciplina Circuitos Digitais.

Professores:   
Geomar Andre Schreiner  
Luciano Lores Caimi

**CHAPECÓ**

**2023**

**SUMÁRIO**

[**1**](#_gjdgxs) **introdução** [3](#_2et92p0)

[**2**](#_tyjcwt) **Descrição da solução 4**

[2.1](#_3dy6vkm) Diagrama com blocos operacionais da solução 4

[2.2](#_3dy6vkm) Módulos que compõem a solução 4

[**3**](#_1t3h5sf) **Relatório de desenvolvimento 5**

[**4**](#_26in1rg) **Equações propostas 7**

**5****Dificuldades e conclusões 7**

# introdução

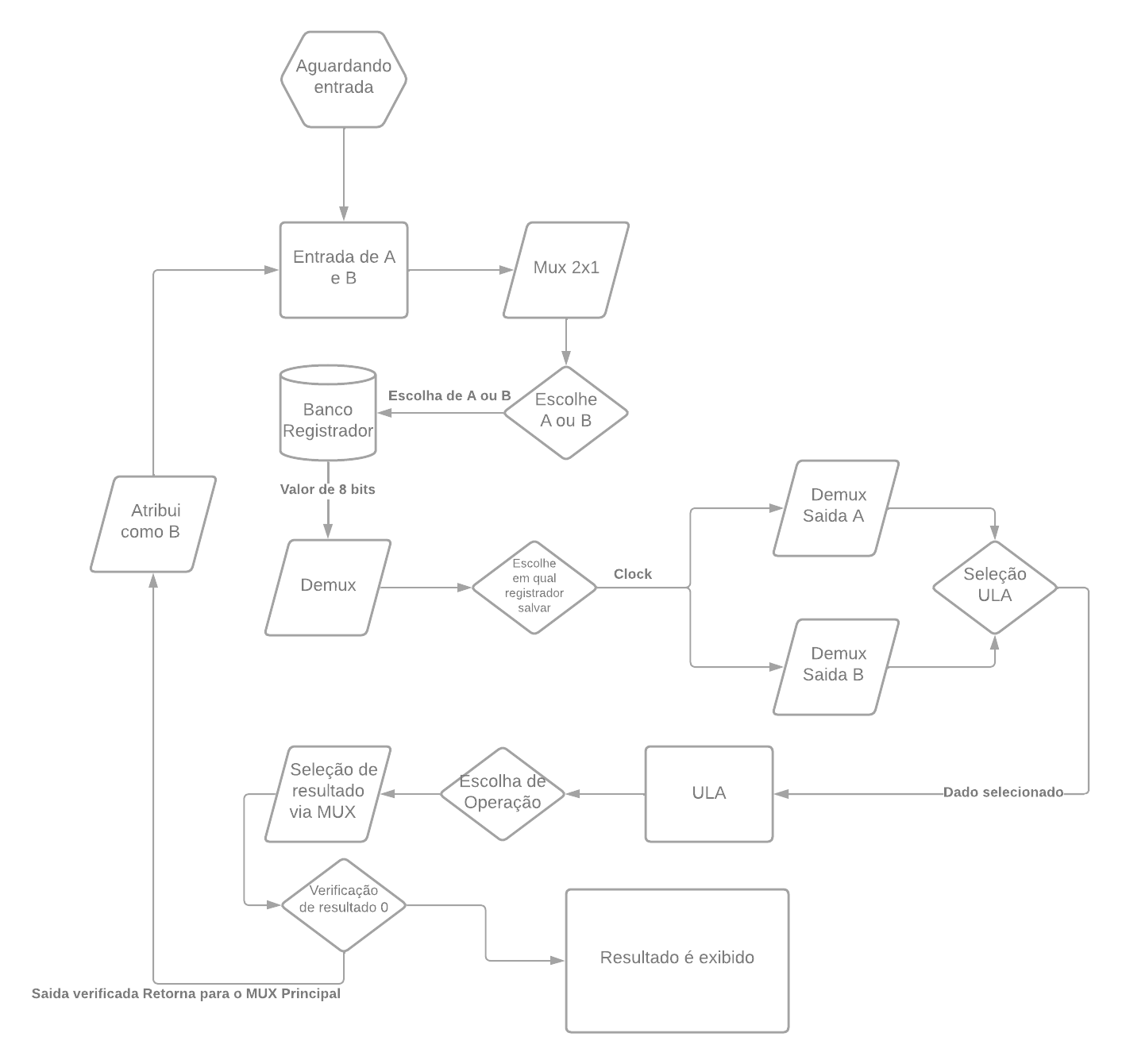
O objetivo principal deste relatório é realçar a aplicação prática do conhecimento na criação de circuitos digitais, com um foco especial na implementação de uma Unidade Lógica Aritmética (ULA) e um banco de registradores. Este projeto tem como seu propósito central a utilização destes componentes para a resolução eficaz e precisa de equações de primeiro e segundo grau. As etapas de construção do circuito foram executadas através da ferramenta de código aberto e gratuita, Logisim Evolution, e envolveram a integração de diversos componentes, incluindo multiplicadores, divisores, multiplexadores e demultiplexadores.

A implementação da Unidade Lógica Aritmética teve como base a ordem estabelecida para o grupo 24.

| SelOps | Operação Lógica/Aritmética |
| --- | --- |
| 000 | A AND B |
| 001 | A \* B |
| 010 | A + B |
| 011 | A NOR B |
| 100 | A AND B |
| 101 | A / B |
| 110 | A XOR B |
| 111 | A - B |

# Descrição da solução

## Diagrama com os blocos operacionais da solução

Para facilitar a compreensão e análise do circuito, apresentamos um diagrama detalhado com blocos operacionais. Este diagrama ilustra claramente a sequência lógica de passos e processos pelos quais o circuito passa para executar suas operações. Cada bloco representa uma função específica ou um componente do circuito, permitindo assim uma visão mais clara e estruturada do funcionamento geral do sistema.   


## **Módulos que compõem a solução**

Para a realização deste projeto, uma gama de módulos, desde os mais básicos até os mais avançados, fram desenvolvidos e implementados. Iniciamos com módulos elementares, como ANDs e ORs de 8 entradas, fundamentais para as operações básicas, e progredimos para módulos mais complexos e interconectados. Cada módulo, independentemente de sua complexidade, será acompanhado de sua respectiva tabela verdade, mapa de Karnaugh e simplificações quando se fizerem necessárias.

## **2.2.1 Componentes de operações básicas**

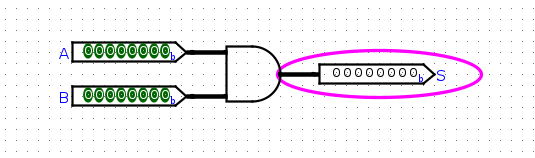
Nesta parte do trabalho, focamos nos componentes fundamentais que são essenciais para as operações lógicas do projeto, esses componentes realizam as operações lógicas básicas necessárias para a estruturação de módulos mais complexos no projeto. A seguir, apresentaremos detalhes sobre o funcionamento de cada um destes componentes no contexto geral do trabalho:

**2.2.1.1 A and B**

O módulo A and B é configurado para processar duas entradas, denominadas "A" e "B", cada uma contendo oito bits. Este componente executa a operação lógica AND bit-a-bit em cada par de bits correspondentes das duas entradas, gerando um resultado que também é expresso em uma saída de oito bits.

A tabela verdade associada a este circuito segue os princípios da tabela verdade AND, realizando uma comparação bit a bit entre as entradas:

| A | B | S |
| --- | --- | --- |
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

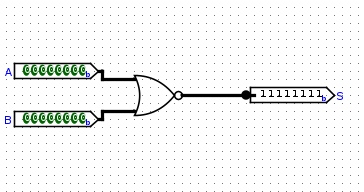
Na sua construção, utilizando o simulador, o componente A AND B foi equipado com dois pinos de entrada, cada um suportando oito bits. O módulo central AND, projetado para processar essas entradas, conta com duas entradas e uma saída, todas com a capacidade de oito bits. Adicionalmente, um pino de saída de oito bits foi integrado para o encaminhamento do resultado processado.  


**2.2.1.2 A nor B**

O módulo A NOR B é projetado para lidar com duas entradas, nomeadas "A" e "B", ambas com oito bits. Este componente realiza a operação lógica OR com saída negada bit-a-bit, onde cada par de bits correspondentes das entradas é comparado, resultando em uma saída também de oito bits.

A tabela verdade deste circuito adere aos princípios da tabela verdade NOR, efetuando uma comparação bit a bit:

| A | B | S |
| --- | --- | --- |
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 0 |

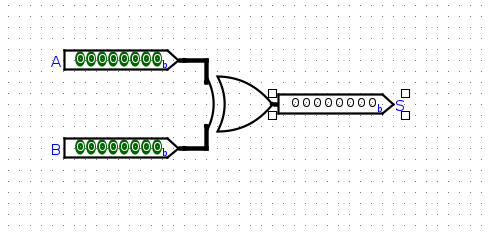
No desenvolvimento deste componente usando o simulador, foram implementados dois pinos de entrada de oito bits. O módulo central NOR, desenhado para trabalhar com estas entradas, é composto por duas entradas e uma saída, todas suportando oito bits. Além disso, foi incluído um pino de saída de oito bits para a transmissão do resultado obtido.  


**2.2.1.3 A xor B**

O módulo A XOR B é configurado para receber duas entradas, identificadas como "A" e "B", cada uma com 8 bits. Este componente executa a operação XOR (exclusive OR) bit a bit, comparando cada par de bits correspondentes das duas entradas para produzir um resultado em uma saída de 8 bits.

A tabela verdade para este circuito segue o padrão da operação XOR, realizando uma comparação bit a bit entre as entradas:

| A | B | S |
| --- | --- | --- |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 0 |

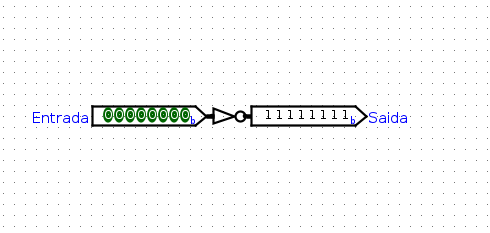
Para a construção deste componente no Logisim Evolution, utilizamos dois pinos de entrada, cada um com capacidade para 8 bits de dados. O componente central XOR, projetado para processar estas entradas, tem duas entradas de 8 bits e uma saída de 8 bits. Adicionalmente, um pino de saída de 8 bits foi integrado para a emissão do resultado processado.  


**2.2.1.4 NOT de 8 bits**

componente NOT de 8 bits é projetado para inverter os bits de sua entrada. Essa operação é conhecida como negação bit a bit, onde cada bit da entrada é invertido individualmente. A funcionalidade é simples, mas essencial em muitas operações lógicas.

A tabela verdade para este componente é direta, refletindo a natureza da operação NOT:

| Entrada | Saída |
| --- | --- |
| 0 | 1 |
| 1 | 0 |

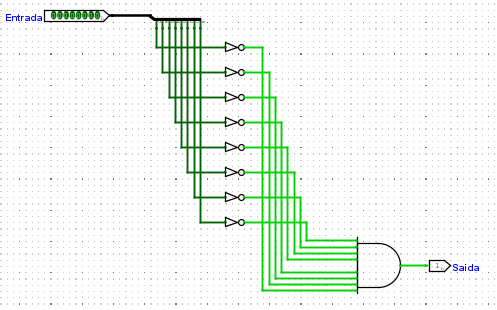
No desenvolvimento deste componente no simulador Logisim Evolution, foi utilizado um pino de entrada com capacidade para 8 bits de dados. Associado a isso, um componente NOT, também de 8 bits, foi implementado para processar a entrada. Por fim, um pino de saída de 8 bits foi estabelecido para transmitir o resultado da inversão dos bits.

**2.2.1.4 Verificador de zero de 8 bits**

O componente "verificar zero 8x1" é projetado para avaliar uma entrada de oito bits e determinar se todos esses bits são zero. A lógica por trás desse componente é baseada no uso de uma porta AND com oito entradas, onde cada entrada é negada, culminando em apenas uma saída. Esta estrutura é eficaz para a verificação pretendida, como demonstrado pela seguinte tabela verdade

| bit 0 | bit 1 | bit 2 | bit 3 | bit 4 | bit 5 | bit 6 | bit 7 | S |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \* | \* | \* | \* | \* | \* | \* | \* | 0 |

Nesta tabela, o asterisco (\*) representa qualquer valor diferente de zero (ou seja, pode ser 0 ou 1). O importante é que se qualquer um dos bits de A a H for 1, a saída será 0. Essa representação simplifica a visualização de que qualquer configuração que não seja todos os zeros resultará em uma saída zero, alinhada com a lógica de negação utilizada no componente.

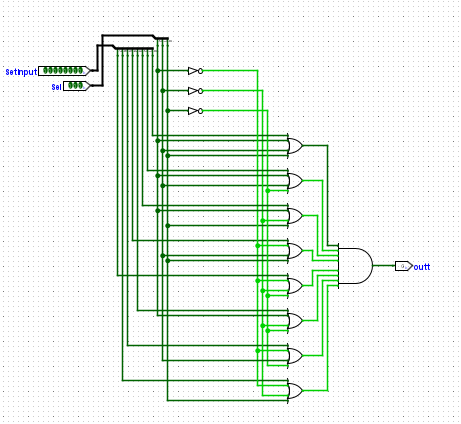
No circuito montado no simulador Logisim Evolution, o componente é composto por uma entrada de oito bits e um componente AND, com todas as suas entradas negadas, resultando em uma saída única. Essa configuração permite verificar eficientemente se todos os bits da entrada são zero:

## **2.2.2 Multiplexadores e demultiplexadores**

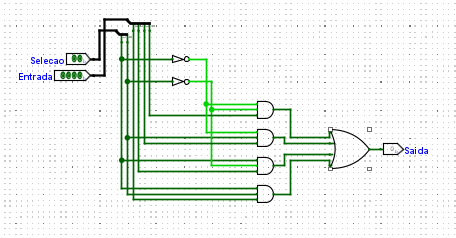
Os componentes de multiplexadores e demultiplexadores desempenham papéis fundamentais no projeto, facilitando a seleção e direcionamento de sinais. O multiplexador, especificamente, é utilizado para escolher um ou mais sinais de entrada e encaminhá-los para uma única saída, funcionando efetivamente como um seletor de dados. Por outro lado, o demultiplexador opera inversamente, distribuindo um único sinal de entrada para uma das várias saídas disponíveis, baseado em critérios de seleção.

Para a realização deste projeto, uma variedade de multiplexadores e demultiplexadores foi implementada, cada um com características específicas adequadas às necessidades do sistema. Entre eles, destacam-se:

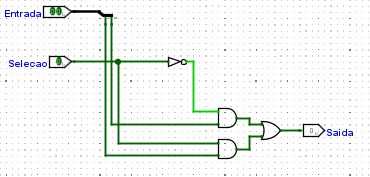
Multiplexador 8x1 de um bit:



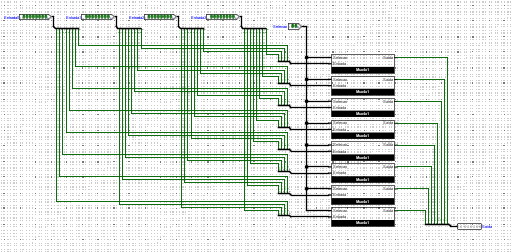
Multiplexador 4x1 de um bit:



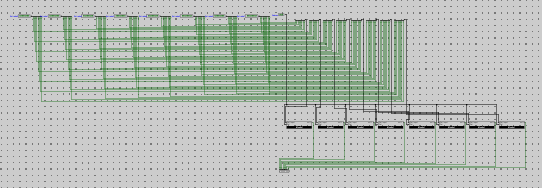
Multiplexador 2x1 de um bit:



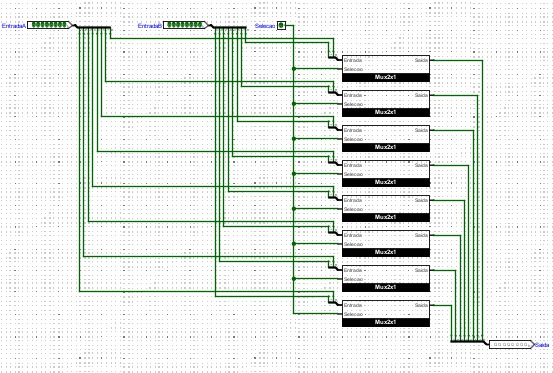
Multiplexador 4x1 de oito bits:



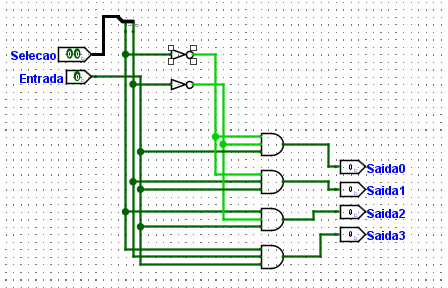
Multiplexador 8x1 de oito bits:



Multiplexador 2x1 de oito bits:



Demultiplexador 2x1 de um bit:



Cada um desses componentes foi cuidadosamente projetado e simulado no Logisim Evolution, demonstrando a flexibilidade e a eficiência que os multiplexadores e demultiplexadores proporcionam ao sistema. Eles são cruciais para o gerenciamento eficiente dos sinais dentro do circuito, permitindo que dados específicos sejam direcionados corretamente para os pontos necessários do projeto.

**2.2.3 Componentes de maior complexidade**

Nesta seção, apresentamos os componentes lógicos de maior complexidade que fazem parte do projeto. Estes componentes são fundamentais para realizar operações lógicas mais elaboradas e são cruciais na estruturação de sistemas integrados dentro do projeto. Vamos detalhar o funcionamento e as características específicas de cada um destes componentes complexos, proporcionando uma compreensão clara de seu papel e contribuição no contexto mais amplo do projeto:

## **2.2.3.1 Componente de Soma Unitaria**

O componente de "Soma Unitária" ou “Somador Unitário” é configurado para processar três bits de entrada, e será componente fundamental num futuro circuito Somador/Subtrator, que realiza a soma ou subtração de números com vários bits. A função deste componente é somar os bits de entrada e fornecer duas saídas: uma com o resultado da soma e outra com o valor do resto. Esse "resto" é crucial para operações de soma em cadeia, onde o resto de uma soma é transferido para a próxima, permitindo a soma bit a bit de números extensos.

Essa operação pode ser melhor entendida na tabela verdade detalhada abaixo:

| A | B | Bit Extra | S | SResto |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Ao lidar com a construção deste circuito, que possui duas saídas distintas, o resultado da soma e o valor do resto, optamos por elaborar tabelas verdade individuais para cada saída. Esta abordagem foi adotada para facilitar a compreensão e implementação de cada função do circuito de forma mais clara e precisa.

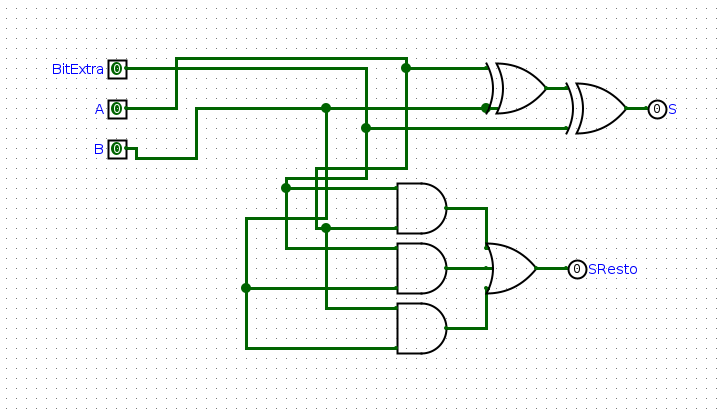
A tabela verdade para a saída do resultado é baseada na premissa de que a saída será 1 sempre que a soma dos bits 1 entre as entradas A, B e BitExtra for um número ímpar. Isso é efetivamente realizado pela equação booleana Axor(BxorC).

| A | B | BitExtra | S |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Por outro lado, a tabela verdade para a saída do resto é construída para refletir a condição de transporte. Neste caso, a expressão booleana simplificada é A.BitExtra + AB + B.BitExtra, indicando as condições sob as quais o resto é gerado.

| A | B | BitExtra | Resto |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

No ambiente do simulador Logisim Evolution, o circuito foi estruturado com três pinos de entrada de um bit cada: BitExtra , A e B. Para determinar a saída do resultado, foram necessários dois componentes XOR. Já para definir o valor do resto, recorremos ao uso de três componentes AND e um OR. Essa configuração é crucial para assegurar que o componente "somador inteiro" funcione corretamente, tornando-o um elemento essencial em sistemas mais complexos, como o Somador/Subtrator, e garantindo a execução precisa de somas de múltiplos bits.

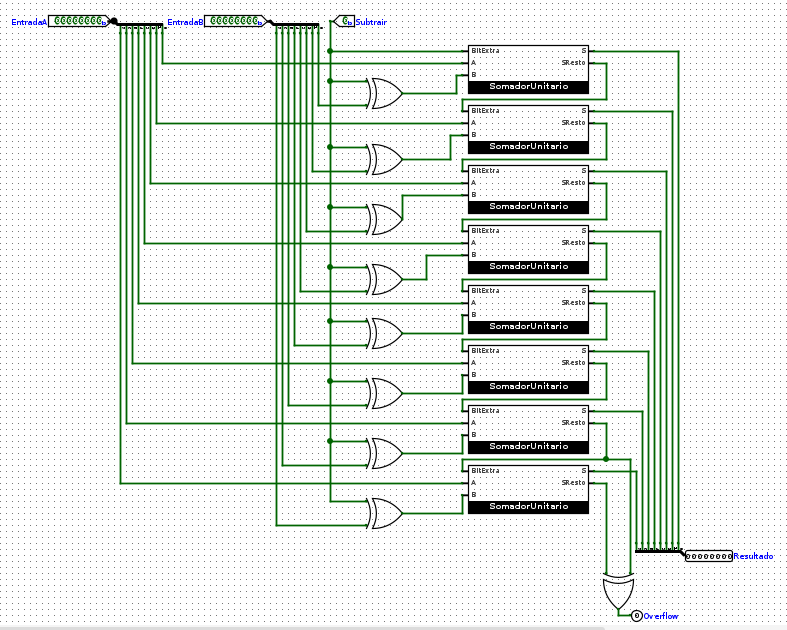


## **2.2.3.2 Componente de Somador/Subtrator**

O componente Somador/Subtrator é projetado para realizar tanto operações de soma quanto de subtração, aproveitando o princípio de que a subtração é, na realidade, uma soma com o segundo valor invertido. No método do complemento de dois, a inversão de um número é efetuada invertendo todos os seus bits e, posteriormente, adicionando 1 ao resultado.

Além das entradas "A" e "B", cada uma com oito bits, o Somador/Subtrator inclui um bit adicional que atua como uma flag para determinar se a operação é uma subtração. Quando esta flag está ativa, o circuito inverte os bits da entrada "B" utilizando a porta XOR. Além de inverter os bits de "B", adiciona 1 ao valor de "BitExtra" na primeira chamada do Somador Unitário. Isso completa o processo de conversão para o complemento de dois. Se a flag de subtração não estiver ativada, o valor de "B" permanece o mesmo.

Para realizar as operações de soma e subtração, o componente chama o circuito de Soma Unitária para cada bit das entradas. Esse componente é crucial, pois realiza a soma bit a bit e calcula o resto para cada par de bits correspondentes das entradas, sempre quando necessário o enviando para a próxima operação. Dessa forma, o Somador/Subtrator, ao utilizar repetidamente o circuito de Soma Unitária, é capaz de efetuar somas e subtrações com precisão e eficiência para números compostos por 8 bits.

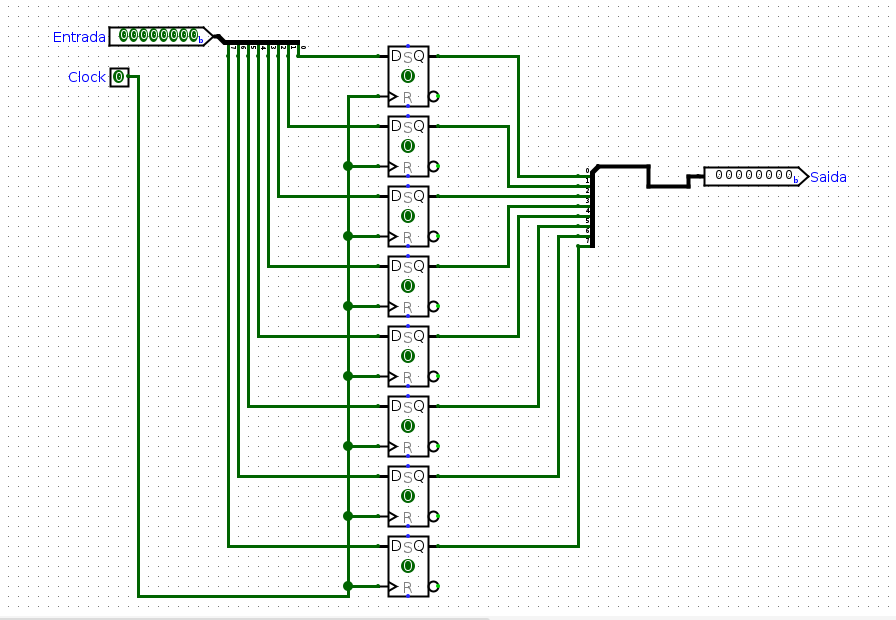


Repare também que circuito no Logisim tem duas saídas: uma para o resultado da soma ou subtração, que é um número de 8 bits, e outra saída que é um bit único. Este bit único é particularmente importante, pois indica se ocorreu um overflow na operação, ou seja, se o resultado excedeu a capacidade máxima que pode ser representada com 8 bits.

## **2.2.3.3 Componentes Registradores**

O registrador, um componente crucial no circuito, tem a função de armazenar temporariamente os dados em uso. Sua estrutura é composta por flip-flops do tipo D, que recebem e mantêm os dados de entrada. Cada bit dos dados de entrada de oito bits é alocado em um flip-flop D individual, e a atualização desses flip-flops ocorre em cada ciclo de clock. A saída do registrador, que representa os dados armazenados, é disponibilizada de forma contínua.

Para a montagem deste registrador no simulador Logisim Evolution, foi utilizado um conjunto específico de componentes. A entrada é composta por um pino que recebe oito bits. Cada um destes bits é direcionado para um dos oito flip-flops do tipo D. Além disso, o circuito inclui um pino de entrada adicional para o sinal de clock, que tem um papel fundamental no controle do momento de atualização dos flip-flops. Finalmente, um pino de saída de oito bits é utilizado para disponibilizar os dados armazenados.



Essa configuração permite que o registrador desempenhe um papel essencial no gerenciamento e na preservação dos dados durante o processamento do circuito, garantindo que as informações estejam disponíveis e atualizadas conforme necessário.

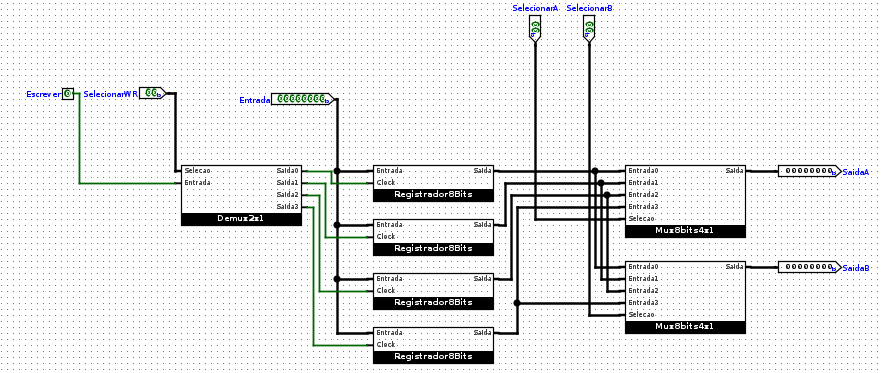
## **2.2.3.4 Componentes Registradores**

O Banco de Registradores é um componente essencial do projeto, desenhado para armazenar dados em uma variedade de registradores e permitir que o usuário escolha qual registrador usar, tanto para armazenamento quanto para recuperação de dados.

Este banco é dividido em duas partes principais. A primeira parte inclui uma entrada de clock de um bit, que controla quando os dados são armazenados nos registradores. Além disso, possui uma entrada de oito bits para os dados a serem armazenados e uma entrada adicional de dois bits, que determina em qual registrador esses dados serão inseridos.

A segunda parte do Banco de Registradores é dedicada à saída de dados. Ela é equipada com duas entradas, cada uma de dois bits, que permitem ao usuário escolher de qual registrador os dados armazenados serão recuperados. Esta funcionalidade torna o banco de registradores extremamente flexível e adaptável às necessidades do usuário.

Internamente, o Banco de Registradores é composto por uma série de componentes interconectados, incluindo dois multiplexadores, um demultiplexador e quatro registradores. Completando esta configuração, estão presentes cinco entradas e duas saídas, que oferecem diversas opções de controle e acessibilidade aos dados armazenados.



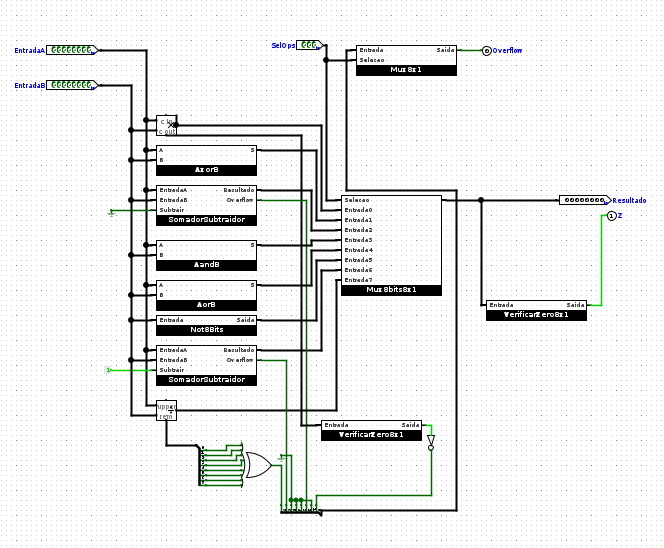
## **2.2.3.4 Componentes Registradores**

A Unidade Lógica Aritmética (ULA) no projeto desempenha um papel crucial, sendo responsável por executar diversas operações aritméticas. Esta unidade permite ao usuário escolher entre diferentes operações aritméticas para serem realizadas em duas entradas distintas.

Dentro do ambiente do simulador, a ULA foi projetada com duas entradas principais, cada uma de oito bits, para receber os dados a serem processados. Além disso, uma entrada adicional de três bits chamada de SelOps é utilizada para selecionar a operação aritmética específica que será aplicada às entradas. O resultado da operação escolhida é então fornecido através da saída do circuito, que é controlada por um multiplexador 8x1.

A ordem e o tipo das operações aritméticas que a ULA pode realizar foram cuidadosamente definidos conforme a tabela de operações do grupo 24. Esta tabela especifica quais operações são possíveis e como elas são selecionadas com base no valor da entrada de seleção de três bits.

Com essa configuração, a ULA se torna um componente versátil e essencial do projeto, capaz de realizar uma ampla gama de cálculos e operações lógicas, fundamentais para o processamento de dados e execução de tarefas dentro do sistema, além de permitir é claro a resolução de equações de primeiro e segundo grau, foco do projeto.  
 Segue a imagem da construção do circuito da nossa ULA



## **2.4 Construção do circuito**

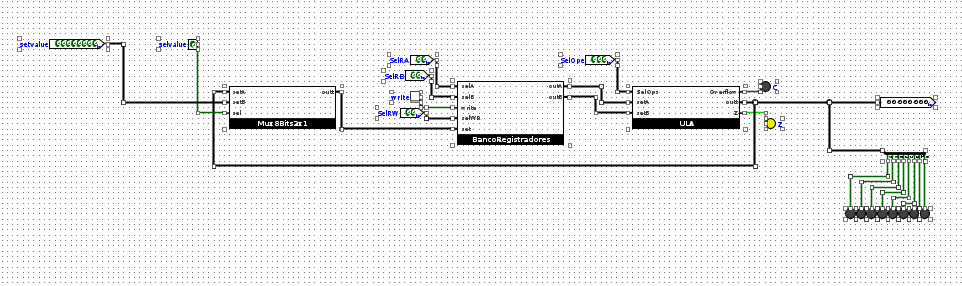
Iniciamos a construção do circuito no simulador Logisim Evolution com uma abordagem baseada na progressão lógica, do simples ao complexo. O ponto de partida foi a elaboração dos componentes mais básicos, que são independentes em termos de funcionalidade. Os primeiros a serem desenvolvidos foram os elementos fundamentais das operações aritméticas, cruciais para a Unidade Lógica e Aritmética (ULA), incluindo as operações XOR, AND, OR, e os circuitos de somadores inteiro e somador/subtrator.

Após estabelecer estas operações básicas, direcionamos nossa atenção para os multiplexadores e demultiplexadores. Estes componentes são particularmente importantes devido à sua ampla aplicabilidade em diversas partes do circuito, atuando como elementos essenciais para a seleção e roteamento de sinais.

O próximo passo foi o desenvolvimento do registrador, um componente que se apoia em flip-flops do tipo D e é fundamental para o armazenamento e manutenção de dados no circuito. Esse desenvolvimento foi um marco importante, pois os registradores são elementos autônomos e têm um papel vital no conjunto do projeto.

Com os componentes básicos prontos, avançamos para a montagem de estruturas mais complexas, como a ULA e o banco de registradores. Estes componentes mais elaborados dependiam da eficiência e funcionalidade dos elementos mais simples construídos nas fases iniciais.

Finalizando o processo, com todos os componentes necessários completos, fomos capazes de montar o circuito principal. Esse circuito foi desenhado para resolver funções de primeiro e segundo graus, ilustrando a aplicação prática e a integração bem-sucedida dos vários componentes desenvolvidos.



## 

# Equações propostas e soluções

Para a resolução de equações utilizando o circuito, duas tipos de funções foram abordadas: uma função de primeiro grau e uma função de segundo grau. Estas funções são expressas e resolvidas da seguinte maneira através do circuito:

Função de Primeiro Grau: Esta função é expressa na forma "y = ax + b", onde 'a' e 'b' são constantes que podem ser definidas pelo grupo. Para calcular o valor de 'y' utilizando o circuito, primeiro inserimos os valores de 'a', 'x' e 'b'. O circuito então realiza a operação aritmética necessária, multiplicando 'a' por 'x' e, em seguida, adicionando 'b' ao resultado dessa multiplicação. Isso resulta no valor de 'y' para a função de primeiro grau dada.

Função de Segundo Grau: A segunda função é uma equação quadrática expressa na forma "y = ax² + bx + c". Aqui, 'a', 'b' e 'c' são constantes que podem ser escolhidas pelo grupo. Para resolver esta função, o circuito precisa realizar múltiplas operações. Primeiro, calcula-se 'ax²', ou seja, 'a' multiplicado por ‘x’ multiplicado 'x'. Em seguida, calcula-se 'bx', o produto de 'b' e 'x'. Finalmente, o circuito soma esses dois resultados com 'c'. A soma final dessas três operações nos dá o valor de 'y' para a função de segundo grau especificada.

# 3.1 Equação de primeiro grau

Para abordar a solução da função de primeiro grau "y = ax + b", o grupo desenvolveu duas abordagens distintas. A primeira é uma solução genérica, aplicável a quaisquer valores de A, B e X, enquanto a segunda é uma solução específica para valores selecionados pelo grupo.

Solução Genérica: A solução genérica foi projetada para ser flexível, permitindo a inserção de qualquer valor para A, B e X. Este método possibilita a aplicação da fórmula "y = ax + b" de forma ampla, atendendo a uma variedade de cenários. A implementação dessa solução no circuito foi documentada e pode ser visualizada na figura correspondente, onde a tabela mostra como diferentes valores de A, B e X são processados para encontrar o resultado de Y.

| Equação: | y=ax+b (generica) |  |  |
| --- | --- | --- | --- |
| Passo | Entrada | Controle | Resultado |
| 1 | InsereValor = a | SelecionarValor = 1  SelecionarRW = 00  Escrever = 0 -= 1-> Q | Insere o valor “a” no R0OO  RO0 =a |
| 2 | InsereValor = X | SelecionarValor = 1  SelecionarRW = 01  Escrever = 0 -= 1-> Q | Insere o valor "X” no R01  RO1=X |
| 3 | InsereValor = b | SelecionarValor = 1  SelecionarRW = 10  Escrever = 0 -= 1-> Q | Insere o valor "b” no R10  R1i0=b |
| 4 |  | SelecionarRA = 00  SelecionarRB = 01  SelOp = 001  SelecionarValor = O  SelecionarRW = 11  Escrever = 0 -> 1-> Q | Realiza a multiplicação do valor no ROO, pelo valor no  RO01 e insere o R11.  R11=a\*X |
| 5 |  | SelecionarRA = 11  SelecionarRB = 10  SelOp = 010 | Realiza a soma do valor no R11 pelo valor do R10 e exibe nos LEDs e no pino de saida  saida = a\*X+b |

Solução Específica: Para a solução específica, o grupo escolheu os valores 5, 7 e 7 para A,B e X respectivamente. Com base na solução genérica já desenvolvida, a aplicação prática dessa solução específica também foi registrada e apresentada em uma tabela. Esta abordagem permitiu ao grupo verificar a funcionalidade do circuito em calcular o resultado de uma função de primeiro grau com valores predefinidos.

| Equação:5.x+7 | | | |
| --- | --- | --- | --- |
| Entrada x1 para variavel x: 7 | | | |
| Resultado da equação: y = 42 | | | |
| ATENÇÃO! ESTA TABELA E UMA SOLUÇÃO PARA 5x+7 QUANDO x = 7. | | | |
| Passo | Entrada | Controle | Resultado |
| 1 | InsereValor = 5 | SelecionarValor = 1  SelecionarRW = 00  Escrever = 0 -> 1-> 0 | Insere o valor 5 no RO0  R00 =5 |
| 2 | InsereValor = 7 | SelecionarValor = 1  SelecionarRW = 01  Escrever = 0 -> 1-> 0 | Insere o valor 7 no RO1  R01=7 |
| 3 | InsereValor = 7 | SelecionarValor = 1  SelecionarRW = 10  Escrever = 0 -> 1-> 0 | Insere o valor 7 no R10  R10=7 |
| 4 |  | SelecionarRA = 00  SelecionarRB = 01  SelOp = 001  SelecionarValor = 0  SelecionarRW = 11  Escrever = 0 -> 1-> 0 | Realiza a multiplicação do valor no R00 (valor 5), pelo valor no R01 (valor 7) e insere o R11.  R11=35 |
| 5 |  | SelecionarRA = 11  SelecionarRB = 10  SelOp = 010 | Realiza a soma do valor no R11 pelo valor do R10 e exibe nos LEDs e no pino de saida  saída = 42 |

# 3.2 Equação de segundo grau

O grupo seguiu uma metodologia semelhante na abordagem da função do segundo grau, começando com a elaboração de uma solução genérica para qualquer equação quadrática. Esta etapa preliminar envolveu o desenvolvimento de um método que pudesse ser aplicado a qualquer conjunto de valores para as variáveis A, B, C e X. O resultado desse esforço foi documentado e pode ser visto na tabela correspondente à solução genérica para funções do segundo grau.

| Passo | Entrada | Controle | Resultado |
| --- | --- | --- | --- |
| 1 | InsereValor = a | SelecionarValor = 1  SelecionarRW = 00  escrever = 0 ->1 -> 0 | Insere o valor .. no ROO  RO00=a |
| 2 | InsereValor = X | SelecionarValor = 1  SelecionarRW = 01  Escrever = 0 -> 1-> O | Insere o valor X\* no RO1  ROI=X |
| 3 |  | SelecionarRA = 00  SelecionarRB = 01  SelOp = 001 SelecionarValor = O  SelecionarRW = 00  Escrever = 0 -> 1-> O | Realza a multiplicação do valor no R00 pelo valor no R0O1 e o  insere no ROO  RO0 = a\*X |
| 4 |  | SelecionarRA = 00  SelecionarRB = 01  SelOp = 001 SelecionarValor = O  SelecionarRW = 00  Escrever = 0 -> 1-> O | Realza a icação do valor  no R00 pelo valor no R01 e o insere no ROO  R00 = aºXº |
| 5 | InsereValor = b | SelecionarValor = 1  SelecionarRW = 10  Escrever = 0 -> 1->0 | Insere o valor "b” no R10  r10=b |
| 6 |  | SelecionarRA = 01  SelecionarRB = 10  SelOp = 001  SelecionarValor = 0  SelecionarRW = 01 Escrever = 0 -> 1-> O | Realza a multiplicação do valor no R01 pelo valor no R10 é o  insere no RO1  RO1 = b\*X |
| 7 | InsereValor = c | SelecionarValor = 1  SelecionarRW = 10  Escrever = 0-> 1-> 0 | Insere o valor "c no R10  Ri0=c |
| 8 |  | SelecionarRA = 0O  SelecionarRB = 01  SelOp =010  SelecionarValor = O  SelecionarRW = 11  Escrever = 0-> 1-> 0 | Soma o valor do ROO com o valo] do R01 e insere no R11  R11 = ax² +bx |
| 9 |  | SelecionarRA = 10  SelecionarRB = 11  SelecionarOperacao = 010 | Soma o valor do R10 com o valo] do R11 e mostra o resultado nos  LEDS € pino de saida  saida= ax²+bx+c |

Após estabelecer a solução genérica, o grupo concentrou-se na equação específica "y = 2x² + 3x + 5". Para isso, foram escolhidos os valores 2, 3, 2 e 5 para A, B, C e X, respectivamente. Com esses valores definidos, a equipe aplicou a metodologia genérica previamente desenvolvida para resolver esta equação quadrática específica. A solução e o processo de cálculo foram novamente registrados em uma tabela, que ilustra como a equação "2x² + 3x + 5" foi resolvida com o valor específico de X igual a 2, resultando em 19.

| Passo | Entrada | Controle | Resultado |
| --- | --- | --- | --- |
| 1 | InsereValor = 2 | SelecionarValor = 1  SelecionarRW = 00  escrever = 0 ->1 -> 0 | Insere o valor .. no ROO  RO00=2 |
| 2 | InsereValor = 2 | SelecionarValor = 1  SelecionarRW = 01  Escrever = 0 -> 1-> O | Insere o valor X\* no RO1  ROI=2 |
| 3 |  | SelecionarRA = 00  SelecionarRB = 01  SelecionarOperacao =001 SelecionarValor = 0  SelecionarRW = 00  Escrever = 0 -> 1-> 0 | Realza a multiplicação do valor no R00 pelo valor no R0O1 e o  insere no ROO  RO0 = 4 (2\*2) |
| 4 |  | SelecionarRA = 00  SelecionarRB = 01  SelOp = 001  SelecionarValor = 0  SelecionarRW = 00  Escrever = 0 -> 1-> 0 | Realza a icação do valor  no R00 pelo valor no R01 e o insere no ROO  R00 = 8 (2\*2²) |
| 5 | InsereValor = 3 | SelecionarValor = 1  SelecionarRW = 10  Escrever = 0 -> 1-> 0 | Insere o valor "b” no R10  R10=3 |
| 6 |  | SelecionarRA = 01  SelecionarRB = 10  SelOp = 001  SelecionarValor = 0  SelecionarRW = 01  Escrever = 0 -> 1-> 0 | Realza a multiplicação do valor no R01 pelo valor no R10 é o  insere no RO1  RO1 = 6 (3\*2) |
| 7 | InsereValor = 5 | SelecionarValor = 1  SelecionarRW = 10  Escrever = 0-> 1-> Q | Insere o valor de c no R10  R10=5 |
| 8 |  | SelecionarRA = 00  SelecionarRB = 01  SelecionarOperaçaão = 010 SelecionarValor = 0  SelecionarRW = 11  Escrever = 0-> 1-> 0 | Soma o valor do R00 com o valor do R01 e insere no R11  R11 = 14 (8+6) |
| 9 |  | SelecionarRA = 10  SelecionarRB = 11  SelecionarOperacao = 010 | Soma o valor do R10 com o valor do R11 e mostra o resultado nos  LEDS e pino de saida  saida= 19 (14+5) |

# 

1. **Dificuldades e conclusões**

Este projeto representou um marco significativo em nossa jornada acadêmica, desafiando-nos e fornecendo uma plataforma valiosa para aplicar e aprofundar nossos conhecimentos. Foi particularmente relevante para solidificar nossa compreensão em portas lógicas e álgebra booleana, elementos centrais na construção dos componentes do circuito.

Além dos aspectos técnicos, o projeto foi uma jornada significativa de autoaprendizado. Aprendemos a importância de buscar soluções criativas e aprimorar nossa habilidade de pesquisa independente. A necessidade de recorrer a fontes de conhecimento alternativas, como vídeos, artigos e tutoriais, foi crucial para a elaboração do projeto. Essas experiências nos ensinaram a adaptar teorias a situações práticas diversas e a buscar conhecimento além dos limites tradicionais.

Essas lições serão inestimáveis em nossa trajetória acadêmica e profissional futura, fornecendo uma base robusta para enfrentar novos desafios e oportunidades.