



**UNIVERSIDADE FEDERAL DO CARIRI
CENTRO DE CIÊNCIAS E TECNOLOGIA
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**EDUARDO PEREIRA DE SOUSA FILHO
SEBASTIÃO SOUSA SOARES**

**RELATÓRIO FINAL
PROJETO CALCULADORA HEXADECIMAL**

**JUAZEIRO DO NORTE - CE
2024**

SUMÁRIO

1. INTRODUÇÃO.....	3
2. DESENVOLVIMENTO.....	3
2.1. Estrutura geral.....	4
2.2. Conversores.....	5
2.3. Soma.....	6
2.4. Comparação.....	9
2.5. Subtração.....	10
2.6. Multiplicação.....	11
2.7. Multiplexador (MUX).....	12
2.8. Circuito “Calculador”	13
2.9. Testes e validação.....	14
3. CONSIDERAÇÕES FINAIS.....	15
REFERÊNCIAS.....	16

1. INTRODUÇÃO

A disciplina de Circuitos Digitais tem como foco, em suma, o estudo dos sistemas digitais, suas estruturas lógicas e seu comportamento. Ao longo das aulas, foi proposto e desenvolvido um projeto que visava criar uma calculadora, por meio do *software “Logisim”* (simulador lógico de circuitos). A proposta é de que a mesma seja capaz de realizar operações básicas (soma, subtração e multiplicação) e determinar o maior, entre dois números inteiros escolhidos pelo usuário, representados na base hexadecimal por *displays* de sete segmentos.

Nesse sentido, o presente relatório apresenta todo o processo de desenvolvimento do mesmo.

PROJETO CALCULADORA HEXADECIMAL

Ciência da Computação 2024.1
Circuitos Digitais - Prof. Ramon Nepomuceno
Eduardo Pereira da Sousa Filho
Sebastião Sousa Soares

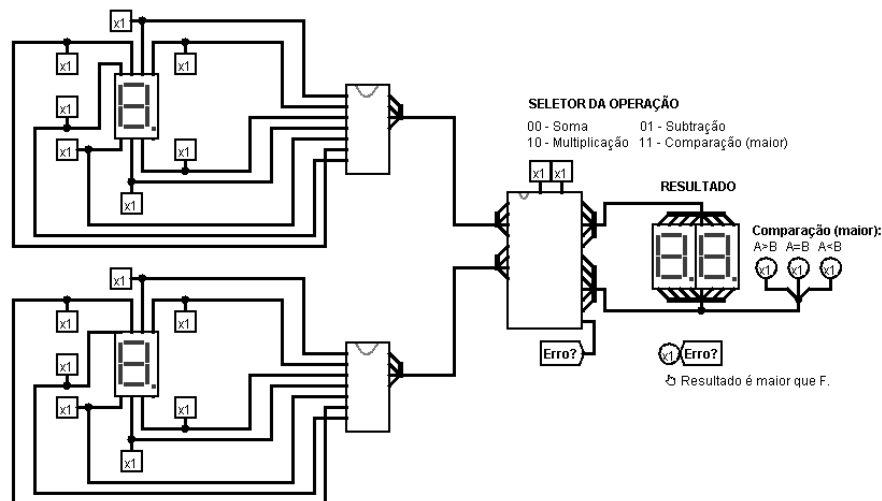


Imagem 1: calculadora de números hexadecimais.

2. DESENVOLVIMENTO

Tendo em vista a proposta apresentada para o projeto, esse foi desenvolvido por etapas, iniciando com o circuito principal (main) e prosseguindo com os demais, associados às operações, ao multiplexador e outros circuitos necessários.

O método principal de auxílio complementar às aulas expositivas e de laboratório da disciplina, foi o canal do *YouTube* “Pedro Souza” para uso do Logisim. Ademais, a lógica por trás dos circuitos foi desenvolvida através da cooperação, trabalho em equipe e orientação do professor.

2.1. Estrutura geral

A princípio, foi desenvolvido o circuito principal que receberia os valores, representados na base hexadecimal por um display de sete segmentos.

Posteriormente, a esse seriam adicionados os conversores das representações do display para valores na base binária. Devido a sua alta complexidade e tamanho do circuito, foi utilizada a Tabela Verdade do Logisim, que permitia manipular os valores da entrada e da saída e gerar o circuito automaticamente. De forma análoga, o resultado das operações também viria a precisar de um conversor, dessa vez para a sua representação compatível com o display.

Além disso, seria necessário a inclusão de um “circuito calculador”, que conteria as operações e um sistema de seleção, para que o usuário pudesse escolher entre elas (mais tarde veremos se tratar de um multiplexador).

A estrutura básica foi construída e atualizada conforme os demais circuitos eram criados, resultando na tela apresentada na imagem 2.

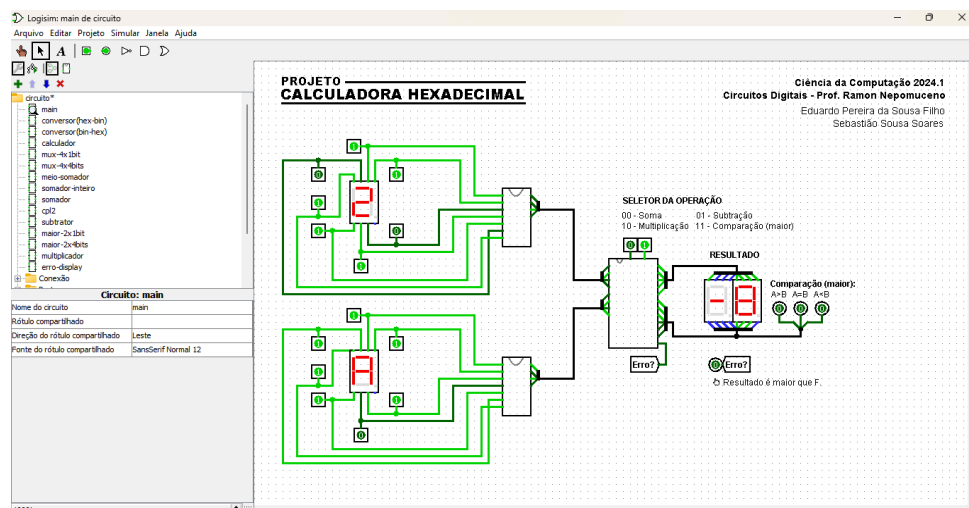


Imagem 2: circuito principal ou *main*.

2.2. Conversores

A entrada e a saída principal de dados, estará representada por displays de sete segmentos, para que haja uma correspondência visual dos valores inseridos e retornados pela calculadora. Assim sendo, é necessário que haja a conversão dos *bits* lidos pelo display (representados pelas letras de “a” a “g”) em dígitos binários, sendo que o maior hexadecimal com um caractere (F, ou 15 na base decimal), é lido em 4 bits, “1111”, na base binária. Portanto, essa será a quantidade de bits padrão de cada número da entrada e também da saída, para efetuar os cálculos.

Sabendo que receberíamos uma entrada de 7 bits de entrada, precisamos converter para binário. Portanto, foi utilizada a tabela abaixo para identificar os valores e realizar a conversão.

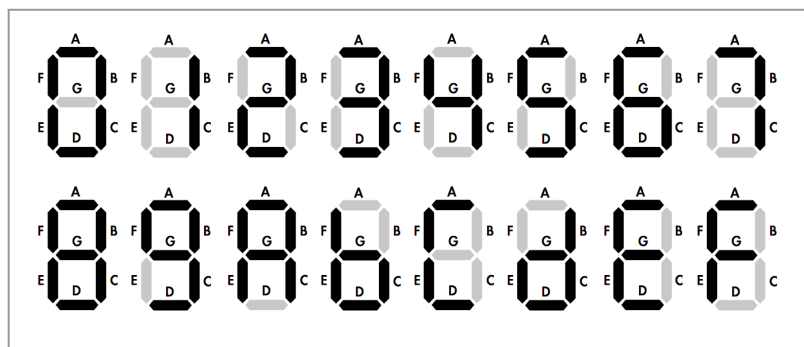


Imagem 3: valores hexadecimais representados no display de sete segmentos.

Ademais, foi realizado o processo inverso para exibir o resultado (convertendo o número binário do resultado em sua versão representável no display). A imagem 4 mostra parte de um desses conversores criados pela tabela verdade do Logisim.

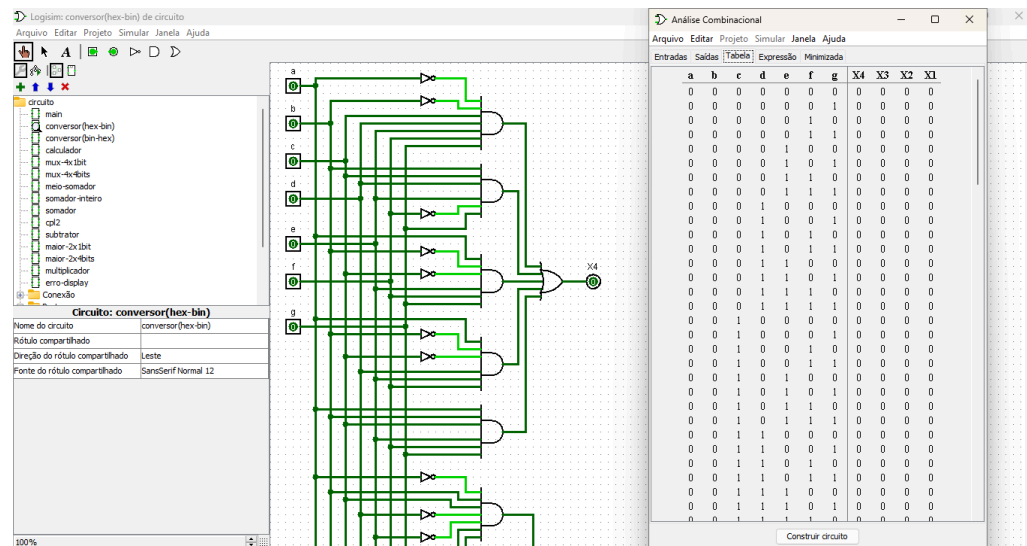


Imagem 4: parte do circuito que converte entradas representáveis no display em números binários e da tabela verdade utilizada.

2.3. Soma

O circuito somador é amplamente utilizado para a montagem do calculador, em que como operação inicial, é tido como base para as outras operações aritméticas. Logo, é responsável por somar números binários, essenciais em cálculos computacionais. Para a composição básica de somadores digitais são divididos em dois meios para a montagem completa: o meio-somador e o somador completo (somador completo possui formas alternativas).

Meio somador (Half Adder):

O entendimento do meio somador é papel fundamental para “construir” uma calculadora, já que seu funcionamento e implementação é essencial para a compreensão de operações mais complexas (subtração e multiplicação). O meio somador é composto por duas entradas, A e B, representando os bits a serem somados, e duas saídas, sendo uma para o bit de soma e outra para o transporte.

A operação lógica do meio somador envolve a adição dos bits de entrada e a definição da saída de soma e do transporte para cálculos posteriores. Um Meio Somador recebe dois bits de entrada A e B e produz dois bits de saída: o Bit de

Soma ($\Sigma = A + B$) e o Bit de Carry — pode ser considerado o bit a ser transferido para a próxima porta, utilizado no nosso circuito como “c0”, desse modo, possuindo a mesma finalidade. Para casos em que a entrada menos significativa (denominada carry) seja considerada, usamos a denominação de circuitos de Somador Completo.

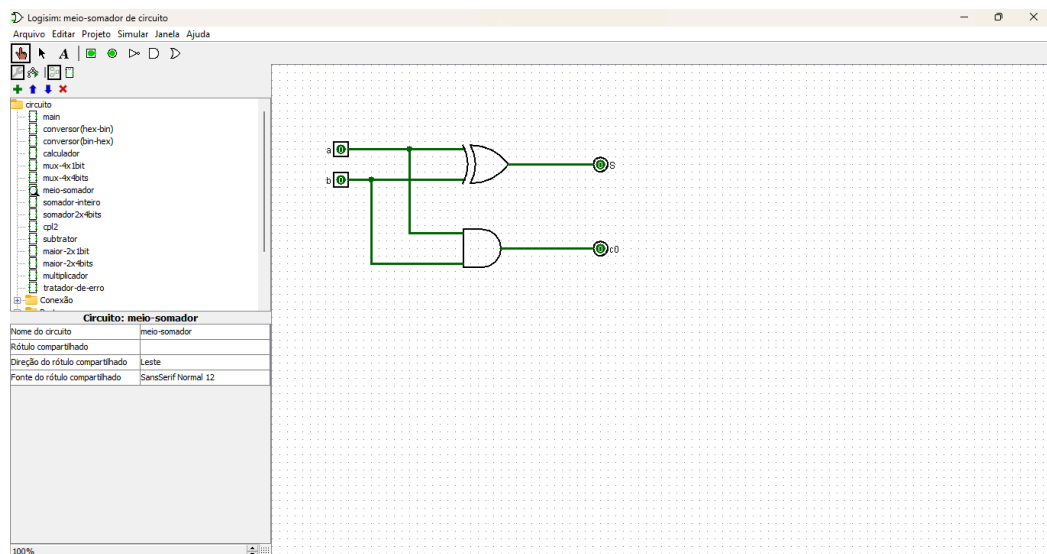


Imagem 5: circuito meio-somador.

Somador completo (Full Adder):

Este dispositivo é responsável por realizar operações de adição binária. O somador completo é um circuito digital capaz de adicionar dois bits de entrada e um bit de carry, gerando um resultado de soma e um carry de saída. Sua principal funcionalidade no projeto é a capacidade de lidar com operações de adição binária utilizando portas lógicas para realizar os cálculos. Nas duas entradas principais, são adicionados os números binários, dos quais o usuário deseja obter um resultado.

Além disso, como supracitado na parte do meio somador, ele possui uma entrada extra, chamada de "carry in", que é como se fosse um "empréstimo" de uma soma anterior. Depois de fazer o cálculo, o somador completo tem duas saídas: uma é o resultado da soma, que é como a caixa onde o número final fica guardado. A outra é o "carry out", que é como um "empréstimo" que pode ser usado na próxima soma.

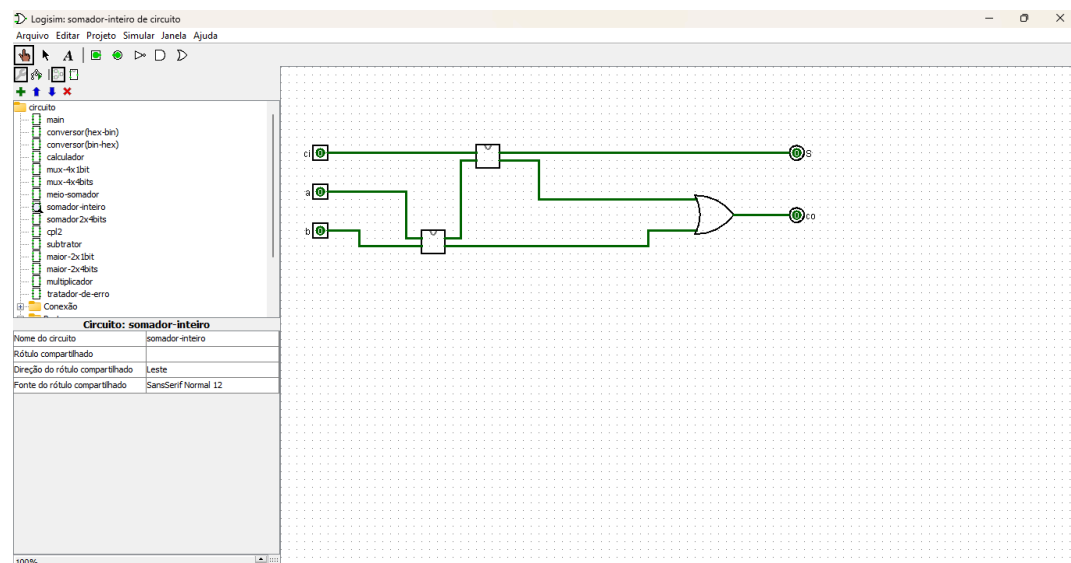


Imagem 6: circuito somador completo.

Por fim, o circuito somador pode ser montado a partir do somador completo, para dois valores ou mais de qualquer tamanho. No nosso circuito, utilizamos a soma de dois valores de 4 bits:

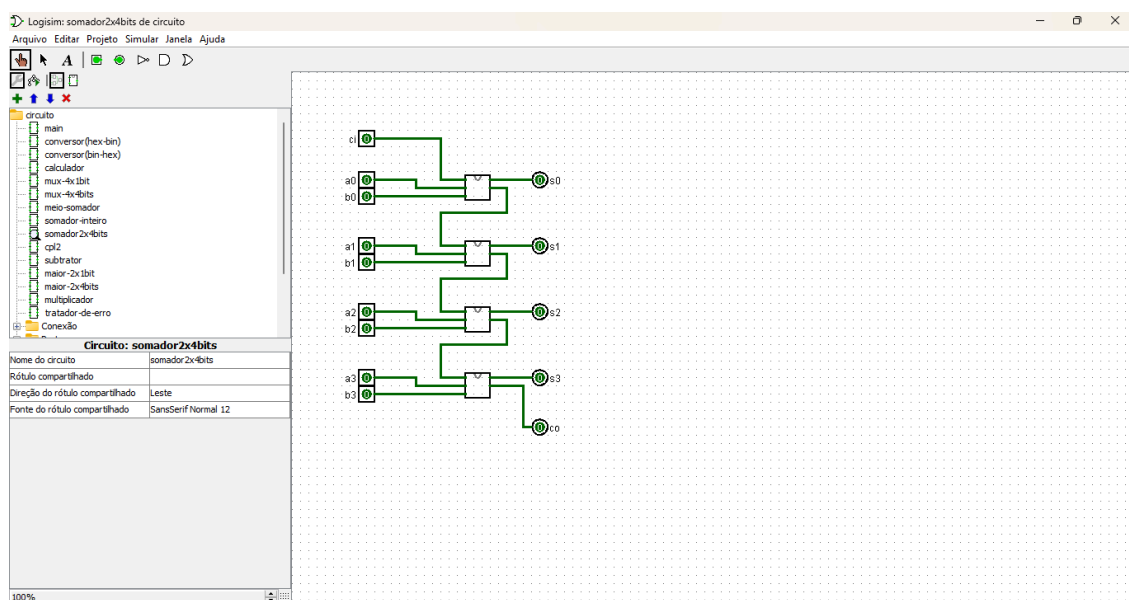


Imagem 7: circuito somador 2x4 bits.

2.4. Comparação

A comparação em circuitos digitais envolve a análise de dois valores (geralmente representados por bits) para determinar se eles são iguais por meio da comparação de duas entradas binárias (A e B) para definir se um é maior que o outro ou se iguais. Essa informação é crucial para a tomada de decisões em circuitos, e no projeto, já que esse circuito foi utilizado em outra operação, como na subtração.

O comparador de dois números de 4 bits é constituído de uma versão menor que compara 2 bits e determina o maior, retornando 0 para o primeiro e 1 para o segundo, além de retornar um bit de diferença que envia 1 se A e B forem diferentes. Assim, o circuito maior combina quatro desses menores para verificar em cada bit, se são iguais, pois, se não, considera qual o maior e ao final, retorna: um bit para diferença, outro para identificar o maior (0 para A e 1 para B) e mais quatro com o maior valor dentre os dois. Esse maior valor é exibido no display de saída quando a operação de comparação é seleccionada.

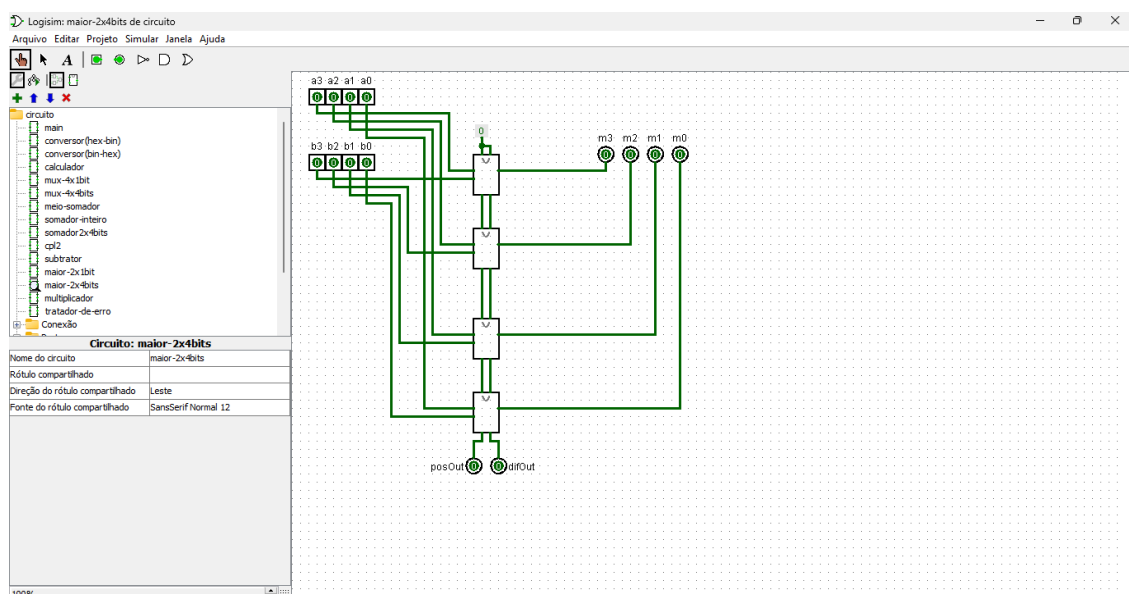


Imagem 8: circuito comparador (maior valor).

2.5. Subtração

Um circuito subtrator é projetado para subtrair um número binário de outro, produzindo o resultado em formato binário. Ele consiste em portas lógicas, como portas OR, AND e NOT. Também foi utilizado o complemento de 2, para que seja viável a representação de um valor negativo. Ao invés de $A-B$, faremos $A+(-B)$, tendo também um bit disponível para demonstração do sinal do resultado, sendo 1 quando negativo, 0 quando positivo.

- Funcionamento do subtrator:
 - O circuito subtrator recebe dois números binários de 4 bits, o minuendo (o número a ser diminuído) e o subtraendo (o número a ser subtraído do outro);
 - O minuendo (A) é inserido no primeiro conjunto de entradas do somador, enquanto o subtraendo (B) é inserido no circuito do complemento de 2.
 - O resultado final é obtido pela saída do somador, representando a subtração binária dos oito números de entrada.
 - No complemento de dois, foi utilizado um raciocínio de inversão do qual 0 torna-se 1 e 1 torna-se 0, após o processo de inversão, soma com 0001 (utilizando o nosso somador), assim, dando o complemento de 2, o qual irá ser utilizado na operação. Além disso, quando finalizado o processo de soma de $A+(-B)$ e obtido o resultado da subtração, tal valor irá retornar um valor que pode ou não ser negativo.
 - Comparação no subtrator: no circuito, não é necessário passar toda vez pelo processo de conversão. Foi adicionado o comparador, junto da subtração, a fim de utilizar o conversor só quando o subtraendo for maior do que o minuendo (resultado vem negativo, ou seja, em complemento de 2). Assim, se $A > B$, o resultado não precisará passar pelo conversor e se $B > A$, o resultado passará pelo conversor.

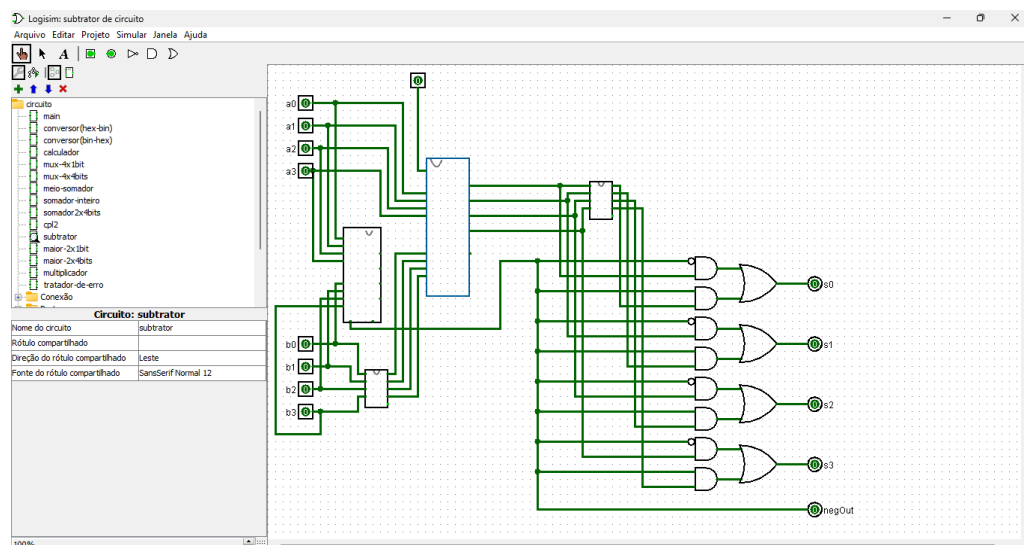


Imagem 9: circuito da subtração.

2.6. Multiplicação

A multiplicação em circuitos digitais envolve combinar cada bit de um dos operandos, multiplicando-o, por cada bit do outro operando. Os resultados parciais são então somados para obter o produto final.

O método utilizado no nosso projeto é o da multiplicação por deslocamento. Nesse método, um dos operandos é deslocado bit a bit (A), enquanto o outro operando é usado para gerar os produtos parciais (B). Esses produtos parciais são então somados para obter o resultado final. Logo, este circuito é projetado para receber dois sinais de entrada e produzir um sinal de saída que é o resultado da multiplicação dos dois sinais de entrada juntos. A operação é obtida por meio do uso de circuitos integrados projetados especificamente para tarefas de multiplicação.

Quando a multiplicação resulta em um valor maior que 15 em decimal, 1111 em binário, ou F em hexadecimal, o multiplicador emite o bit de “erro”, para evitar que o resultado incompleto seja exibido como resultado.

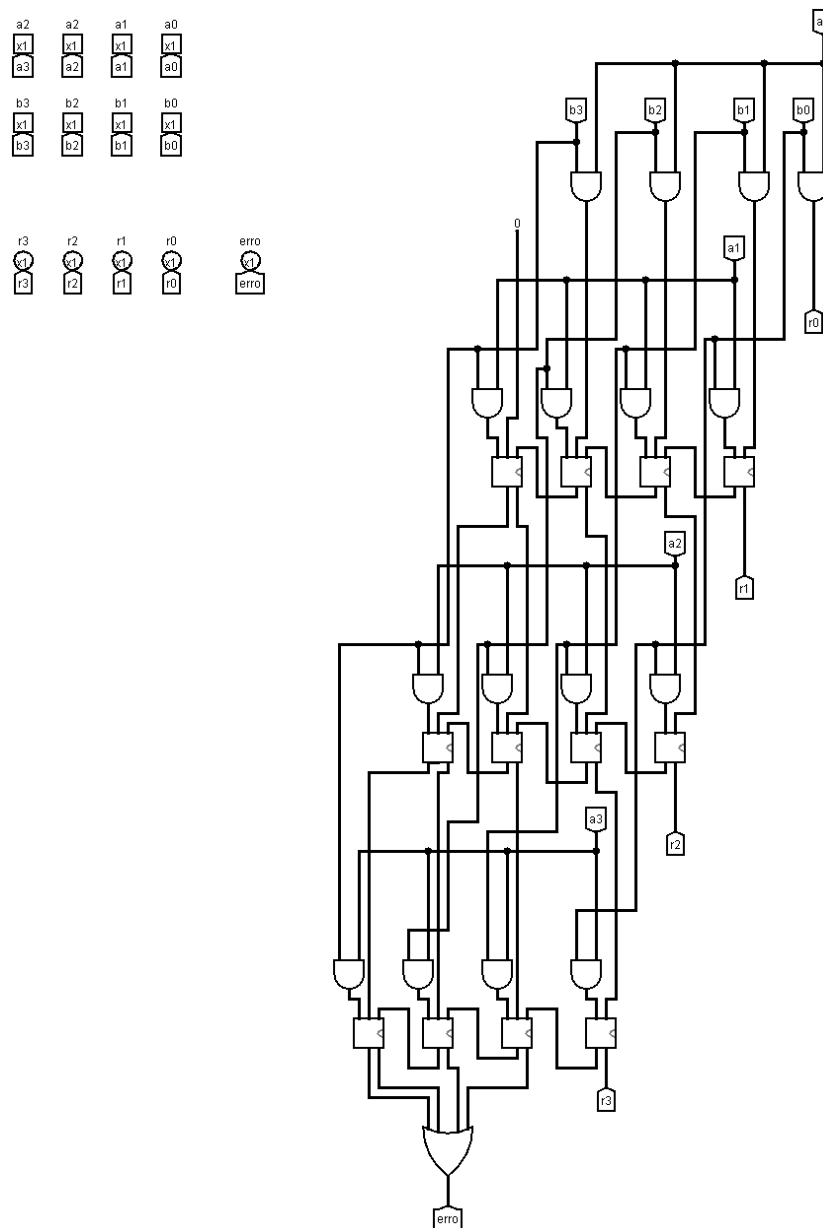


Imagem 10: circuito multiplicador.

2.7. Multiplexador (MUX)

O Multiplexador ou MUX, é o circuito responsável por receber pelo menos dois valores e a partir de um seletor de 1 bit ou mais enviar à saída o resultado correspondente. Na nossa calculadora, ele é utilizado para escolher entre as quatro operações, aquela que o usuário desejar.

Desse modo, para desenvolver o nosso MUX, nós utilizamos, para começar, uma versão reduzida que seleciona entre quatro valores de 1 bit (*MUX 4x1 bit*). Por tanto, são seis entradas (*op1*, *op2*, *aX*, *bX*, *cX* e *dX*, sendo *op1* e *op2* referentes ao seletor) e uma saída *r*. Assim, quando o seletor for “00”, então $r = aX$, quando “01”, $r = bX$ e assim sucessivamente.

De maneira análoga, o *MUX 4x4 bits* (16 entradas, além do seletor) é construído da mesma forma, porém, sendo analisados bit a bit os 4 números da entrada. O circuito principal está representado na imagem abaixo.

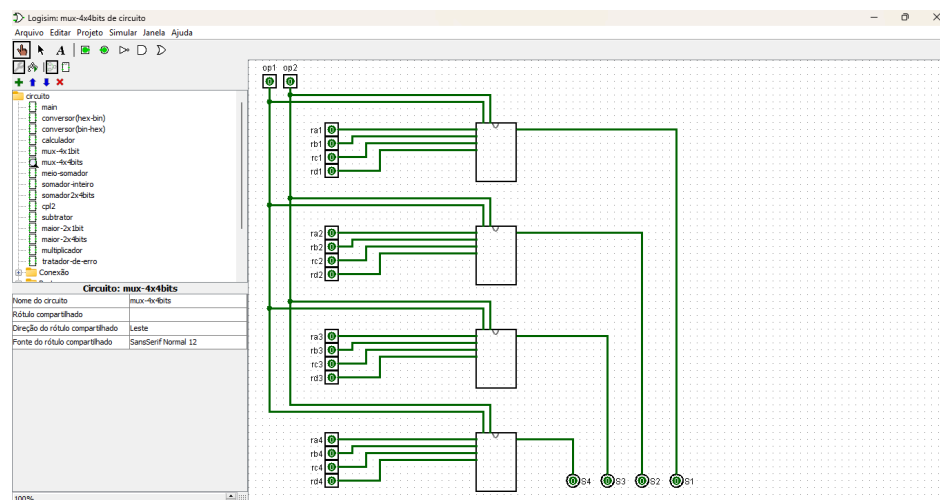


Imagem 11: MUX de quatro entradas de quatro bits.

2.8. Circuito “Calculador”

O circuito Calculador é o responsável por receber as entradas já convertidas em binário, realizar os cálculos, receber a operação e enviá-la para o MUX e retornar as saídas: resultado (7 bits, já convertido em hexadecimal), comparação (3 bits), negativo (1 bit) e erro (1 bit). Vejamos cada saída do calculador:

- **Resultado:** valor de 0 a 1111 (na base binária).
- **Comparação:** o primeiro bit retorna 1, se $A > B$; o segundo, se $A = B$; e, o terceiro, se $A < B$, sendo que se a operação de comparação não for selecionada, todos retornam 0.

- **Negativo:** é representado por um bit único que enviará 1 diretamente para a entrada *g* de um display que ficará ao lado do que exibe o resultado, indicando a negatividade do número (só é enviado se a operação for subtração e $A < B$).
- **Erro:** esse bit indica que o resultado ultrapassou o valor máximo capaz de ser representado por um caractere hexadecimal (F, ou 15 em decimal). O mesmo só retorna 1 se a operação for soma ou multiplicação e o resultado precisar de mais do que quatro bits para ser representado na base binária.

Abaixo, vê-se, na imagem 5, o circuito Calculador.

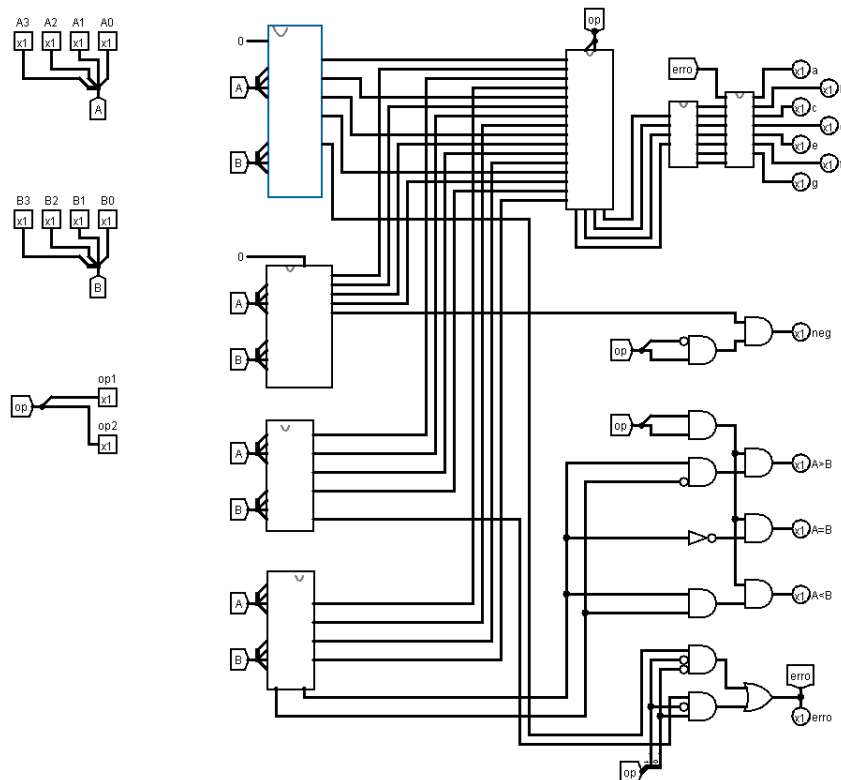


Imagem 12: circuito Calculador, responsável por processar e retornar os dados.

2.9. Testes e validação

Os testes do projeto foram conduzidos de forma estratégica, utilizando valores máximos e mínimos permitidos pelo sistema hexadecimal (0x0 a 0xF) para validar o funcionamento correto das operações de soma, subtração, multiplicação e

comparação. Verificamos especificamente o comportamento do bit de erro, ativado em casos de estouro de 4 bits, e o bit de negativo, acionado em operações de subtração com resultados negativos.

Após a validação dos circuitos individuais, testamos o sistema integrado, focando na correta operação do multiplexador (MUX) e dos conversores entre as bases binária e hexadecimal. Conforme testagem foram feitos ajustes necessários e ao final foi confirmado o funcionamento correto do circuito, com os resultados sendo exibidos corretamente nos displays de sete segmentos.

3. CONSIDERAÇÕES FINAIS

O desenvolvimento do projeto da calculadora hexadecimal proporcionou uma aplicação prática valiosa dos conhecimentos adquiridos na disciplina de Circuitos Digitais. Ao longo do processo, foram enfrentadas algumas dificuldades especialmente na construção e integração dos diversos componentes, mas o trabalho em equipe foi fundamental para superá-las.

Por fim, o projeto completo foi disponibilizado no GitHub, no repositório “*CalculatorLogisim*”, permitindo que outros estudantes possam explorar o mesmo.

REFERÊNCIAS

WIDMER, Neal S. Sistemas digitais: princípios e aplicações. 12. ed. São Paulo: Pearson Education do Brasil, c2019. 1034 p. ISBN 9788543025018.

SOARES, S. S.; SOUSA FILHO, E. P. *CalculatorLogisim* [repositório]. 2024. Disponível em: <https://github.com/SebastiaoSoares/CalculatorLogisim>. Acesso em: 20 set. 2024.

Pedro Souza. [CIRCUITOS DIGITAIS] Aula 15 - Utilização do Logisim. Ano de publicação. Disponível em: <https://www.youtube.com/watch?v=32ritDV6XxU>. Acesso em: 23 ago. 2024.