

**HENRY JOSÉ QUINTÃO COSTA HENRIQUES**

**IMPLEMENTAÇÃO DE UMA ULA SOBRE OPERAÇÕES EM  
COMPLEMENTO-2**

Itajaí (SC), de 29 de junho de 2023



**UNIVALI**

**UNIVERSIDADE DO VALE DO ITAJAÍ**

**IMPLEMENTAÇÃO DE UMA ULA SOBRE OPERAÇÕES EM  
COMPLEMENTO-2**

por

**HENRY JOSÉ QUINTÃO COSTA HENRIQUES**

Relatório de implementação de uma ULA sobre  
operação de complemento-2 em Introdução à Engenharia da Computação

Coorientador(a): DOUGLAS ROSSI

Itajaí (SC), de 29 de junho de 2023

# **IMPLEMENTAÇÃO DE UMA ULA SOBRE OPERAÇÕES EM COMPLEMENTO-2**

HENRY JOSÉ QUINTÃO COSTA HENRIQUES

/ 29 de junho de 2023

Professor: DOUGLAS ROSSI

Disciplina: Introdução à Engenharia da Computação

Linha de Pesquisa: Sistemas Digitais

Número de páginas: 18

## **RESUMO**

Este trabalho tem como objetivo a implementação de uma Unidade Lógica Aritmética que opera em complemento-2. Esse circuito deve realizar as seguintes operações sobre operandos inteiros: passagem direta(Pass), decremento de 1, incremento de 1, adição, subtração e lógica básica AND, OR e NOT. Para isso, será utilizado o software de simulação de circuitos digitais, o Logisim, e desse modo, por meio de implementação e concatenação lógica de sub-circuitos, junto a utilização dos princípios básicos de algebra booleana alinhado a circuitos aritméticos, obtém-se o bloco funcional de Unidade Lógica Aritmética. Além disso, será implementado, também, status de operação Overflow Flag e Zero Flag a fim de analisar possíveis resultados da saída de dados.

## LISTA DE ILUSTRAÇÕES

Figura 1.	Interface do Software Logism.....	7
Figura 2.	Circuito meio-somador .....	8
Figura 3.	Circuito somador completo .....	9
Figura 4.	Circuito somador de quatro bits .....	9
Figura 5.	Circuito meio-subtrator .....	10
Figura 6.	Circuito subtrator completo.....	11
Figura 7.	Circuito subtrator de 4 bits.....	11
Figura 8.	Circuito Incrementador de 4 bits .....	11
Figura 9.	Circuito Decrementador de 4 bits .....	12
Figura 10.	Circuito de passagem direta .....	13
Figura 11.	Circuito convencionais de algebra booleana de 4 bits.....	13
Figura 12.	Circuito de convenção para complemento-2.....	14
Figura 13.	Circuito de status Overflow-Flag .....	15
Figura 14.	Circuito de status Zero-Flag .....	15
Figura 15.	ULA que realiza operações em complemento-2 sobre valores de 4 bits. ....	16
Figura 16.	Resultado final da ULA.....	17

## LISTA DE TABELAS

Tabela 1.	Tabela verdade meio-somador .....	8
Tabela 2.	Tabela verdade somador-completo .....	9
Tabela 3.	Tabela verdade meio-subtrator .....	10
Tabela 4.	Tabela verdade subtrator completo .....	10

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>6</b>
<b>1.1</b>	<b>CRITÉRIOS .....</b>	<b>6</b>
<b>1.2</b>	<b>METODOLOGIA DA PESQUISA .....</b>	<b>6</b>
<b>1.3</b>	<b>FERRAMENTAS DE PROJEÇÃO .....</b>	<b>7</b>
<b>2</b>	<b>DESENVOLVIMENTO .....</b>	<b>8</b>
<b>2.1</b>	<b>PROJETO DO SUB-CIRCUITO DE ADIÇÃO .....</b>	<b>8</b>
<b>2.1.1</b>	<b>Meio-Somador .....</b>	<b>8</b>
<b>2.1.2</b>	<b>Somador Completo .....</b>	<b>9</b>
<b>2.1.3</b>	<b>Somador de 4 Bits.....</b>	<b>9</b>
<b>2.2</b>	<b>PROJETO DO SUB-CIRCUITO DE SUBTRAÇÃO .....</b>	<b>10</b>
<b>2.2.1</b>	<b>Meio-Subtrator .....</b>	<b>10</b>
<b>2.2.2</b>	<b>Subtrator Completo .....</b>	<b>10</b>
<b>2.2.3</b>	<b>Subtrator de 4 bits .....</b>	<b>11</b>
<b>2.3</b>	<b>PROJETO DO SUB-CIRCUITO INCREMENTADOR DE 4 BITS ...</b>	<b>11</b>
<b>2.4</b>	<b>PROJETO DO SUB-CIRCUITO DECREMENTADOR DE 4 BITS .</b>	<b>12</b>
<b>2.5</b>	<b>PROJETO DO SUB-CIRCUITO PASSAGEM DIRETA .....</b>	<b>12</b>
<b>2.6</b>	<b>PROJETO DOS SUB-CIRCUITOS DE ALGEBRA BOOLENA AND, OR E NOT .....</b>	<b>13</b>
<b>2.7</b>	<b>PROJETO DOS SUB-CIRCUITOS DE COMPLEMENTO-2 .....</b>	<b>14</b>
<b>2.8</b>	<b>PROJETO DOS SUB-CIRCUITOS DE STATUS DE OPERAÇÃO ..</b>	<b>15</b>
<b>2.8.1</b>	<b>Overflow-Flag .....</b>	<b>15</b>
<b>2.8.2</b>	<b>Zero-Flag .....</b>	<b>15</b>
<b>2.9</b>	<b>PROJETO FINAL DA ULA .....</b>	<b>16</b>
<b>3</b>	<b>CONCLUSÃO .....</b>	<b>17</b>
	<b>Referências .....</b>	<b>18</b>

# 1 INTRODUÇÃO

A implementação desta ULA requer noções fundamentais de projeção de somadores aritméticos e suas variações, estes podem ser encontrados no capítulo 6 Funções de Lógica Combinacional (FLOYD, 2009), no qual será o sustentáculo de todo o desenvolvimento do circuito.

Esse microprocessador deve realizar as seguintes operações sobre operandos inteiros: passagem direta(Pass), decremento de 1, incremento de 1, adição, subtração e lógica básica AND, OR e NOT.

## 1.1 CRITÉRIOS

De modo a atingir os fins técnicos exigidos pelo avaliador, a ULA deve atender aos seguintes critérios:

- Criou um bloco lógico (recurso do LOGISIM) correspondente à implementação da ULA?
- Fez uso de elementos de iteração que permite testar a ULA, isto é, gerar valores de entrada, selecionar o tipo de operação e visualizar as saídas?
- Documentou o circuito identificando as entradas/saídas usando as mesmas nomenclaturas do enunciado?

Ao final desse relatório, é necessário que todas as exigências vigentes acima tenham sido cumpridas.

## 1.2 METODOLOGIA DA PESQUISA

Os meios metodológicos utilizados neste relatório estão totalmente alinhados com o conteúdo passado durante o período do curso, portanto, as ferramentas são, se não totalmente, análogas as matérias equivalentes mostradas em sala:

- Mapa de Karnaugh: método de simplificação gráfica;

- Álgebra Booleana;
- Circuitos Aritméticos;

### 1.3 FERRAMENTAS DE PROJEÇÃO

Sob esse viés, o espaço de trabalho digital utilizado para desenvolver a ULA foi o software gratuito de simulação de circuitos digitais, o Logisim. Esta ferramenta permite o desenho e simulação através de uma interface gráfica. Por meio dele, é possível criar desde circuitos simples a sistemas mais complexos, muito popular entre professores para aprendizado de lógica digital em universidades e cursos técnicos.

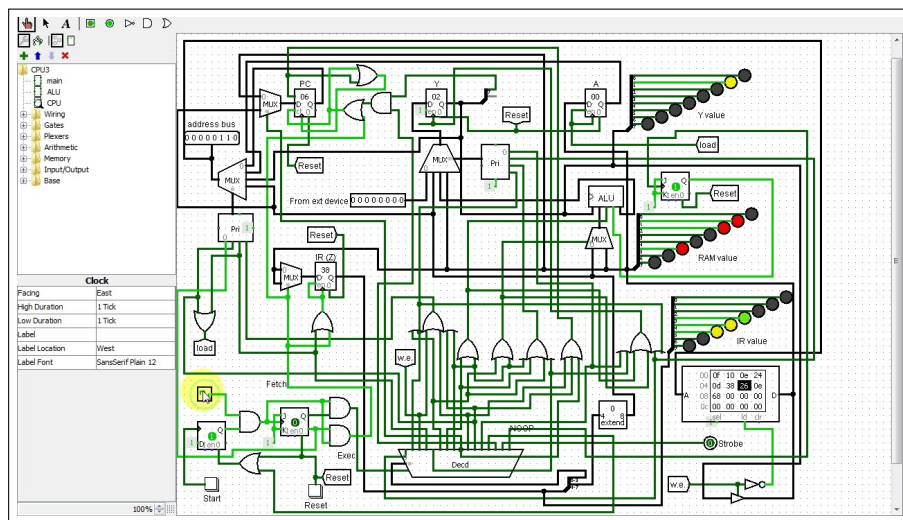


Figura 1. Interface do Software Logisim  
Fonte: Adaptado de (BURCH, 2002)



## 2 DESENVOLVIMENTO

Em primeira instância, é importante definir a ordem de implementação das operações. Foi definido, a fim de otimizar o tempo de projeção dos circuitos, que a operação de adição seria o primeiro sub-circuito a ser desenvolvido, pois as demais operação não booleanas são, em suma, variações do raciocínio utilização no desenvolvimento dessa.

### 2.1 PROJETO DO SUB-CIRCUITO DE ADIÇÃO

Deve-se compreender, inicialmente, que esse circuito é composto por um somador de 4 bits que foi projetado a partir da concatenação lógica de quatro somadores básicos, e estes, por sua vez, por intermédio de meio-somadores.

#### 2.1.1 Meio-Somador

Desse modo, inicia-se o projeto criando um circuito meio somador, que se utiliza da porta XOR como principal componente da sua lógica, respeitando o comportamento de entrada e saída de dados descrito na tabela a seguir:

Tabela 1. Tabela verdade meio-somador

Cin		Cout	
A	B	S	Ts
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Desenhando o circuito lógico que descreve o comportamento dessa tabela, obtém-se o seguinte circuito meio-somador:

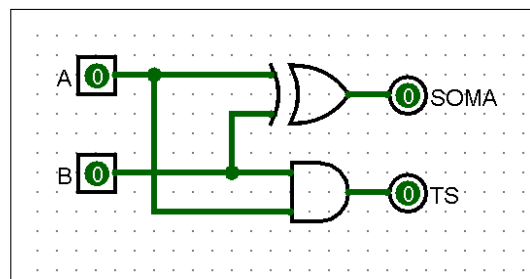


Figura 2. Circuito meio-somador

### 2.1.2 Somador Completo

Outrossim, também faz-se necessário a utilização de um somador completo, que respeita a lógica da seguinte tabela verdade, sendo TE: Transporte de entrada e TS: Transporte de saída:

Tabela 2. Tabela verdade somador-completo

C <sub>in</sub>		C <sub>out</sub>	
A	B	S	T <sub>s</sub>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Desenhando o circuito conforme a lógica descrita acima, obtém-se o seguinte resultado:

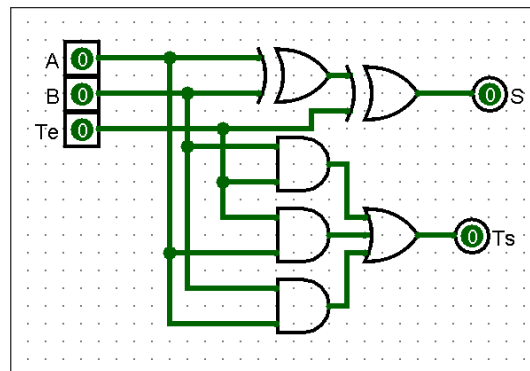


Figura 3. Circuito somador completo

### 2.1.3 Somador de 4 Bits

Dessa maneira, concatenando ordenadamente um bloco meio somador e três blocos somadores completos, junto a isso, uma porta XOR para armazenamento da saída que ultrapasse 4 bits de saída, obtém-se um somador de 4 bits, finalizando, assim, a projeção do sub-circuito de adição:

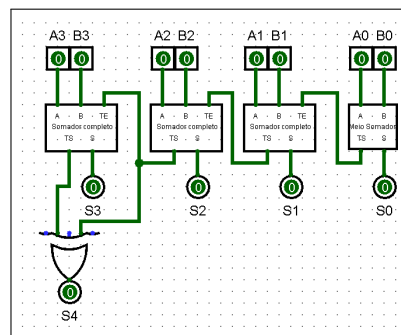


Figura 4. Circuito somador de quatro bits

## 2.2 PROJETO DO SUB-CIRCUITO DE SUBTRAÇÃO

Logo, após criar o sub-circuito de adição, é possível inferir a lógica utilizada nele para criar os sub circuitos de subtração, incremento e decremento. Nesse caso, continuará-se a partir do da criação da lógica do circuito de subtração.

### 2.2.1 Meio-Subtrator

O circuito de subtração utiliza lógica semelhante ao de adição, descrita pela seguinte tabela:

Tabela 3. Tabela verdade meio-subtrator

Cin		Cout	
A	B	S	Ts
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Assim, obtém-se o seguinte circuito:

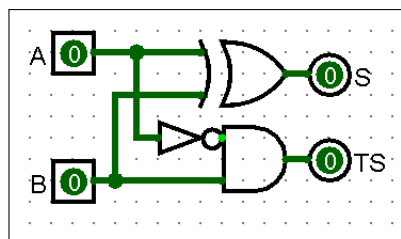


Figura 5. Circuito meio-subtrator

### 2.2.2 Subtrator Completo

Assim, do mesmo modo que foi projetado o circuito de Somador completo, cria-se, com lógica análoga, o circuito de Subtrator Completo, respeitando a seguinte lógica:

Tabela 4. Tabela verdade subtrator completo

Cin		Cout	
A	B	S	Ts
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Dessa maneira, obtém-se o seguinte circuito:



Esse circuito, basicamente, adiciona 1 unidade do valor de entrada. Assim, se na entrada for digitado o valor 8 - 1000(2), a saída será o valor 9 - 1001(2).

Para projetar um circuito incrementador de 4 bits, concatena-se, ordenadamente, 4 blocos de circuitos meio-somadores de modo que a saída TS na entrada B, deixando somente a primeira entrada disponível para alteração, que, logicamente, será o bit responsável por incrementar o valor principal.

## 2.4 PROJETO DO SUB-CIRCUITO DECREMENTADOR DE 4 BITS

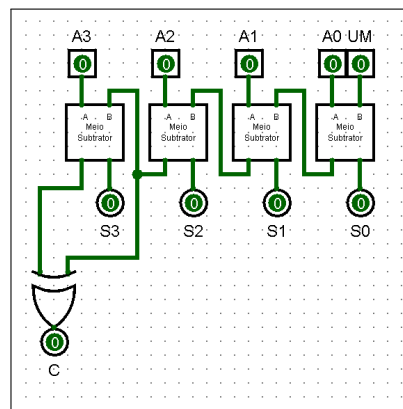


Figura 9. Circuito Decrementador de 4 bits

Esse circuito, basicamente, subtrai em 1 unidade do valor de entrada. Assim, se na entrada for digitado o valor 8 - 1000(2), a saída será o valor 7 - 111(2).

Para projetar um circuito decrementador de 4 bits, concatena-se, ordenadamente, 4 blocos de circuitos meio-subtratores de modo que a saída TS na entrada B, deixando somente a primeira entrada disponível para alteração, que, logicamente, será o bit responsável por decrementar o valor principal.

## 2.5 PROJETO DO SUB-CIRCUITO PASSAGEM DIRETA

O circuito de passagem direta apenas reflete o valor da entrada principal na saída, sem efetuar nenhuma operação lógica. Ele possui um circuito simples que descreve essa passagem:

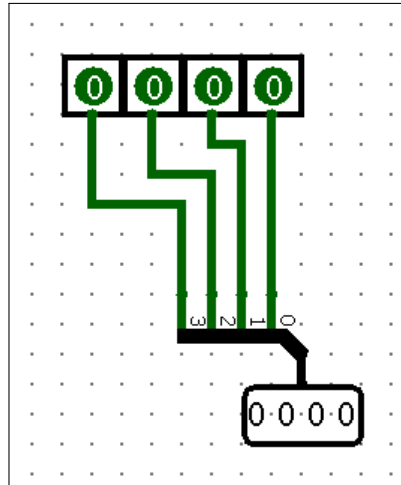


Figura 10. Circuito de passagem direta

## 2.6 PROJETO DOS SUB-CIRCUITOS DE ALGEBRA BOOLENA AND, OR E NOT

Estes circuitos funcionam iguais aos seus equivalentes convencionais, porém, operando sobre 4 bits ao invés de 2 convencionais. Portanto, eles realizam operações sobre as entradas conforme seu funcionamento habitual.

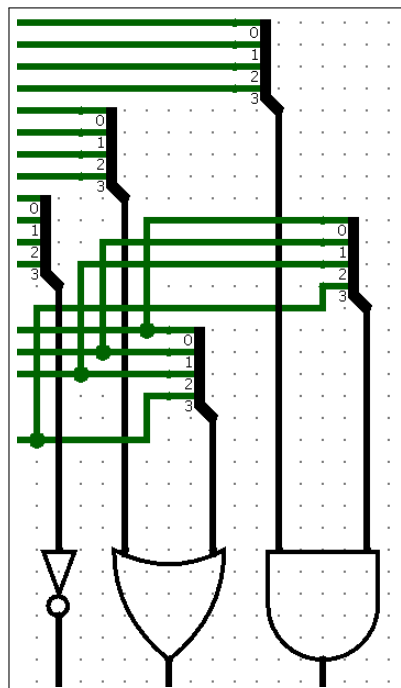


Figura 11. Circuito convencionais de algebra booleana de 4 bits

## 2.7 PROJETO DOS SUB-CIRCUITOS DE COMPLEMENTO-2

O complemento-2 é o equivalente aos números negativos em lógica binária, para utilizá-los, a sequência habitual de 0...15 é dividida ao meio, logo, os números de 0 à 7 são dedicados aos números positivos, enquanto os números de 8 à 15 aos negativos. Isso inferi que os números em que a quarta posição iniciar em zero(0001,0010,0011,0100,0101,0110,0111) são os positivos, enquanto os iniciados em um(1000,1001,1010,1011,1100,1101,1110,1111) são negativos. Portanto, além disso, é importante entender que existe uma lógica para converter os números binários positivos para negativos. Para isso, inverte-se os números da entrada, trocando o que é zero pelo o que é um e vice versa e acrescentando um. Por exemplo: para converter 5(0101) para seu equivalente negativo, primeiro inverte-se seu valor binário(0101 para 1010) e acrescenta 1( $1010 + 1 = 1011$ ). Assim, os -5, em binário é 1011. Desta maneira, para traduzir essa lógica para um circuito que faça essa transformação, é importante entender que é necessário fazer essa lógica ao contrário, pois irá-se trabalhar com valores negativos já na entrada de dados, a saída também vai ser um valor negativo. Para resolver esse problema, criará-se um bloco funciona que fará a convenção dos valores somente para os displays de entrada e saída. O que induz que os valores de entrada e saída em complemento-2 são meramente visuais. Ou seja, quando um valor negativo é digitado, a operação utilizada é seu valor original, porém, ao passar pelo bloco de complemento-2 na saída, ele é novamente convertido para seu valor negativo, mostrando no display de saída o valor correto da conta. O circuito que descreve essa lógica é visto a seguir:

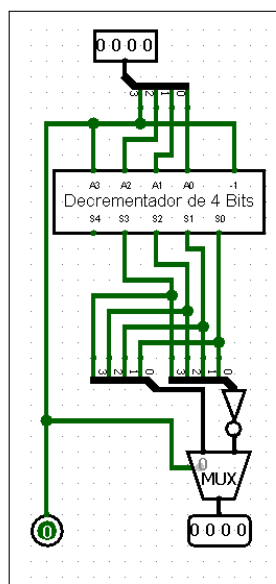


Figura 12. Circuito de convenção para complemento-2

## 2.8 PROJETO DOS SUB-CIRCUITOS DE STATUS DE OPERAÇÃO

Status de operações são utilizados para avaliar condição desejadas para a análise de dados, neste caso, serão avaliados os status de OverFlow Flag e Zero Flag.

### 2.8.1 Overflow-Flag

O status de Overflow-Flag refere-se ao caso em que a saída de dados não possui uma quantidade suficiente de bits necessária para representar a operações realizadas na ULA. Quando isso ocorre, um saída deve acionar em nível alto representando essa condição, ou seja, o Overflow-Flag.

Como nos circuitos anteriores haviam saídas extras geradas pela construção dos circuitos, que representavam justamente a extrapolação das operações, foi necessário criar somente um circuito que passava essas saída extras para acionar o nível de Overflow-Flag. Para isso, utilizá-se um MUX sincronizados com as operações, assim se ocorresse uma extrapolação de apresentação do resultado em alguma operação, isso seria evidência por esse circuito.

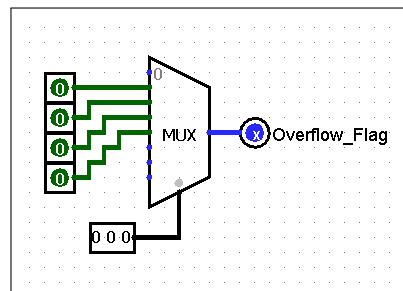


Figura 13. Circuito de status Overflow-Flag

### 2.8.2 Zero-Flag

O status de Zero-Flag deve acionar uma saída em nível alto sempre que o resultado de uma operação é igual a 0. o circuito deve descreve isso pode ser representando com um bloco que recebe a saída invertida conectada numa porta AND que acionará quando a saída for zero.

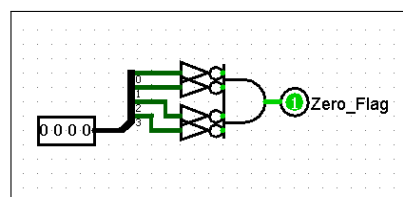


Figura 14. Circuito de status Zero-Flag



## 2.9 PROJETO FINAL DA ULA

Após o termino da criação dos sub-circuitos, pode-se passar para o projeto final. O centro da lógica da ULA se baseia num multiplexificador que operará passando o resultado das operações conforme a escolha da operação do usuário, para isso, concatena-se a saída de dados dos sub-circuitos num MUX de 4 e bits de dados e 3 bits de seleção(8 entradas de 0 à 7) de modo que as operações repõem uma ordem de numeração binário que pode ser utilizadas no botão select a fim de representar ordenadamente o resultado das operações num display.

Os blocos de implementação da lógica de complemento-2 são adicionados na entrada e na saída a fim de fazer a conversão dos valores. Vale mencionar também que os status de operação devem ser adicionados fora da lógica principal, mas ainda captando os dados desejados.

Logo, o resultado final do circuito pode ser visto a seguir:

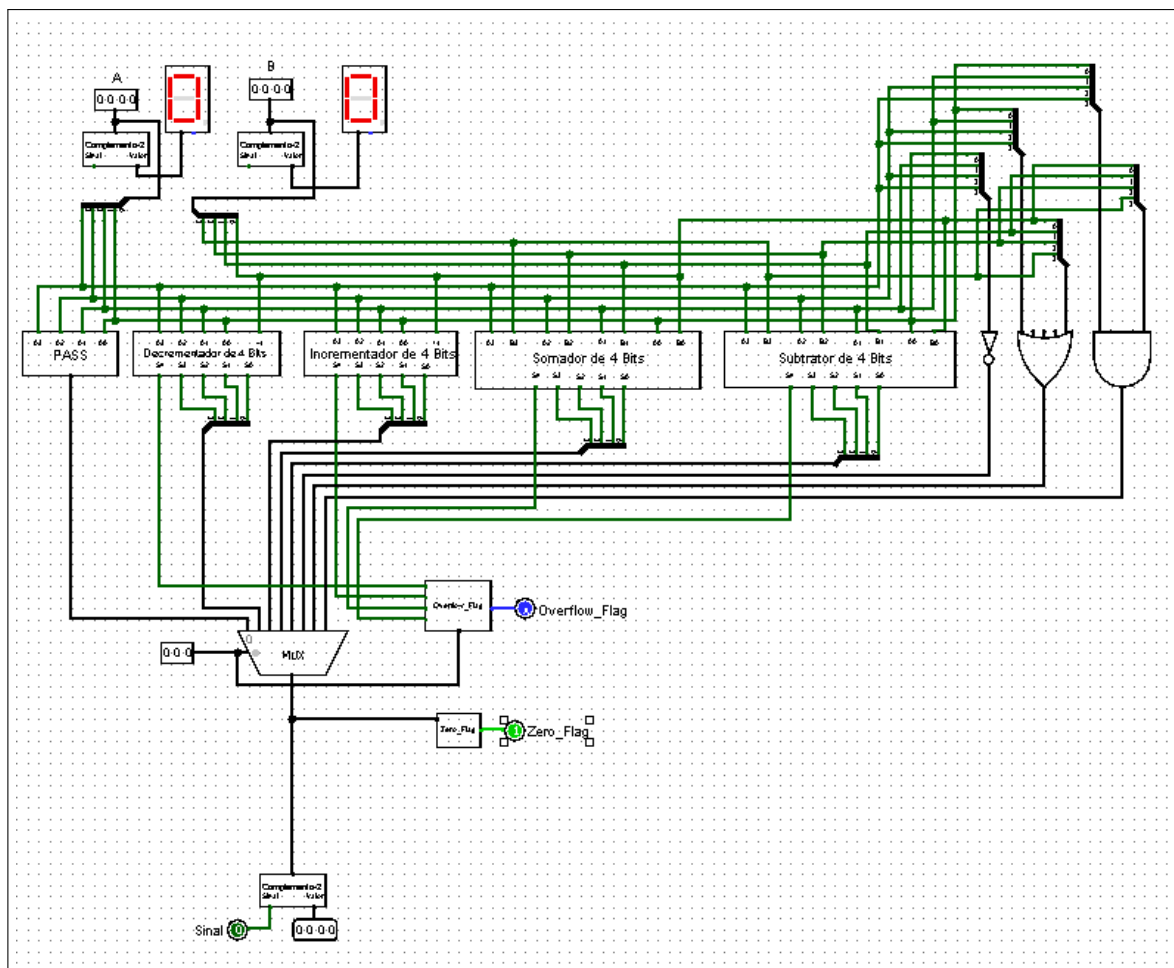


Figura 15. ULA que realiza operações em complemento-2 sobre valores de 4 bits.

### 3 CONCLUSÃO

Desse modo, o trabalho proponha a projeção de uma ULA que realize as operações de passagem direta(Pass), decremento de 1, incremento de 1, adição, subtração e lógica básica AND, OR e NOT sobre complemento-2 em entradas de 4 bits. Além disso, que fosse sinalizado os status de Overflow-Flag e Zero-Flag.

Para isso, foi necessário a criação de sub-circuitos que realizassem essas operações separadamente e estes, fossem formados por outro circuitos mais simples, responsáveis por construir lógicas mais complexas quando ordenadamente juntadas.

Houve, também, a necessidade construção de lógicas específicas sem uma base booleana, mas bem estruturas e funcionais dentro da proposta de desenvolvimento da ULA, como no caso do circuito de complemento-2, Overflow Flag e Zero-Flag.

Ao final do trabalho, juntou-se, de maneira, organizada, concatenada e ordenada, todos os blocos lógicos de sub-circuitos junto a um multiplexificador que teve o papel de distribuir o resultado das operações conforme a escolha do usuário por meio da entrada Select, obtendo-se, dessa maneira, o mais simples microprocessador:

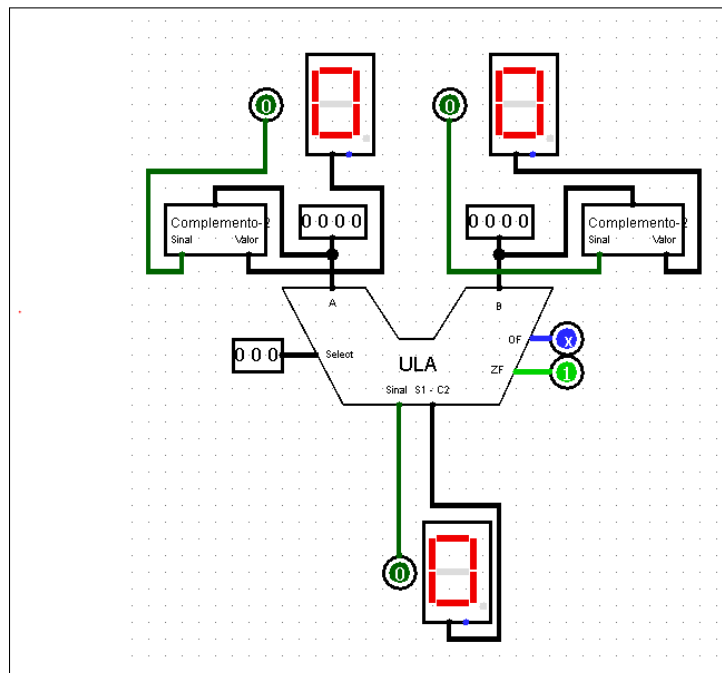


Figura 16. Resultado final da ULA

## REFERÊNCIAS

BURCH, C. Logisim: A graphical system for logic circuit design and simulation. **Journal on Educational Resources in Computing (JERIC)**, ACM New York, NY, USA, v. 2, n. 1, p. 5–16, 2002.

FLOYD, T. **Sistemas digitais: fundamentos e aplicações**. [S.l.]: Bookman Editora, 2009.