

**Universidade Federal de Pernambuco - UFPE**  
**CIN - Centro de Informática**

# **Projeto de Sistemas Digitais**

## **Unidade Lógica Aritmética (ULA)**

Williams Douglas José dos Santos (wdjs)  
Vinícius Felipe Barbosa (vfb2)  
Lucas Yule Rocha de Melo Araújo (lyrma)  
Carlos Eduardo Mendonça Clark (cemc)

RECIFE  
2021

## Sumário

1. Introdução.....	
2. Visão geral do Projeto.....	
3. Detalhamento de cada módulo.....	
3.1 Somador/Subtrator.....	
3.2 Complementador a 2.....	
3.3 Comparador.....	
3.3.1 A igual a B.....	
3.3.2 A maior que B.....	
3.3.3 A menor que B.....	
3.4 A And B e A Xor B.....	
3.4.1 Função AND.....	
3.4.2 Função XOR.....	
3.5 Multiplexadores.....	
3.5.1 Mux 2:1 1Bit.....	
3.5.2 Mux 8:1 1 Bit.....	
3.5.3 Mux 8:1 5 Bits.....	
4. Conclusão.....	

## **1 Introdução**

Neste projeto acadêmico, relativo à disciplina de Sistemas Digitais, estão contidas as etapas de desenvolvimento de uma Unidade Lógica Aritmética(ULA).

A ULA projetada irá realizar operações aritméticas(Soma e Subtração) e operações lógicas(And, Xor, Comparação e Complemento a 2), para cada operação será apresentado Tabela-verdade, assim como o Mapa-K e o circuito obtido a partir dos passos anteriores, para a construção dos circuitos foi utilizado o software Quartus.

## 2 Visão Geral do Projeto

Uma ULA (Unidade Lógica Aritmética), tem como função a realização de operações aritméticas de modo que torne prático e hábil o processo de obter informações a partir dos cálculos. Esse processo é realizado de forma binária, onde são recebidos N inputs (vetores binários), e esses são manipulados por circuitos lógicos e/ou aritméticos. Esses circuitos realizam determinadas operações a partir da combinação e manipulação de portas lógicas.

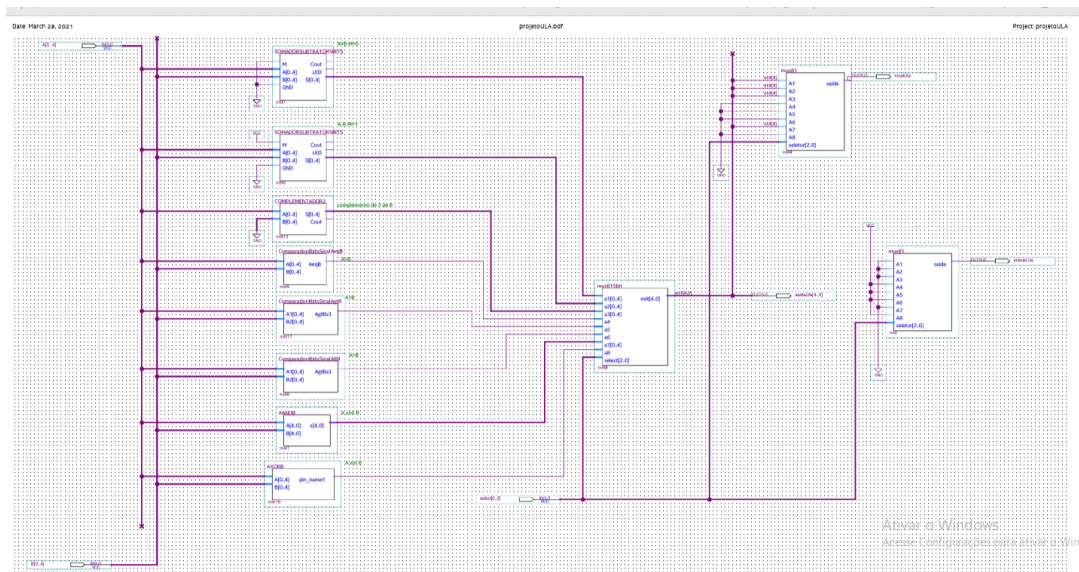
Em nosso projeto, a construção da ULA foi feita tal que ela fosse capaz de realizar algumas operações, que são selecionadas a partir de chaves seletoras. São as operações:

- $F = A + B$
- $F = A - B$
- $F = \text{Complemento a 2 de } B$
- $F = A = B$
- $F = A > B$
- $F = A < B$
- $F = A \text{ AND } B$
- $F = A \text{ XOR } B$

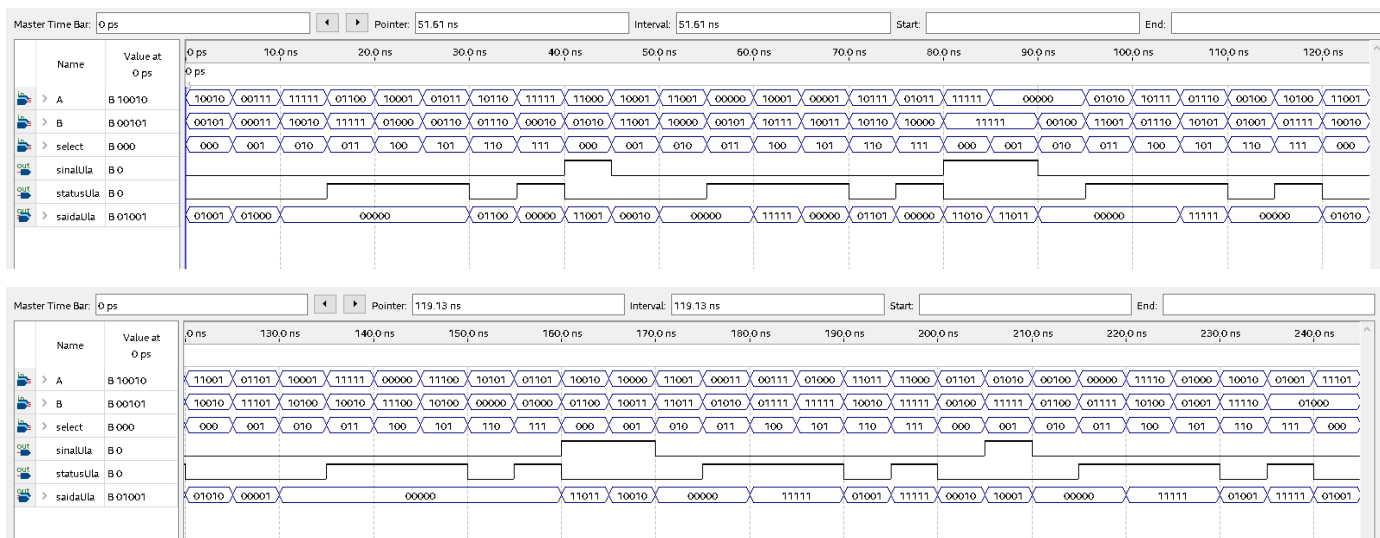
As entradas da ULA são dois vetores de 5 bits, sendo 1 de sinal, e um vetor de 3 bits que representa a chave seletora de operações.

As saídas esperadas da ULA são, 1 vetor resultando o resultado da operação, 1 bit indicando o sinal do resultado da operação (para as operações que retornam um vetor), 1 bit indicando que a operação foi do tipo booleano (status).

### ● Circuito projetado:



## ● Waveform:



## 3 Detalhamento de cada módulo

### 3.1 Somador/Subtrator

Este módulo tem como objetivo a soma ou subtração de duas cadeias binárias de 5 bits. O circuito básico será composto de um circuito somador de 2 bit (1 bit de A e 1 bit de B), que quando posto em paralelo com o outros somadores de 2 bits, será capaz de realizar a soma de 5 bits.

Para o circuito básico de um somador, foi utilizado um *full-adder* de 2 bits, o processo de soma, leva ou não em consideração a presença do Carry-in. Para a formação do somador completo de 5 bits, o C-out de cada somador básico foi ligado ao C-in do próximo somador, criando assim um cascadeamento.

Segue a Tabela verdade, Mapa-K do circuito básico:

A	B	C-in	S	C-out
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
0	1	1	1	1

Tabela-verdade do somador 2 bits

A partir desta tabela-verdade, foi possível gerar os seguintes Mapa-K para a saída S e o Carry-in:

**Mapa-K para a saída S:**

Cin\AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

**Mapa-K para C-out:**

Cin\AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

A partir dos mapas foram obtidas as seguintes funções:

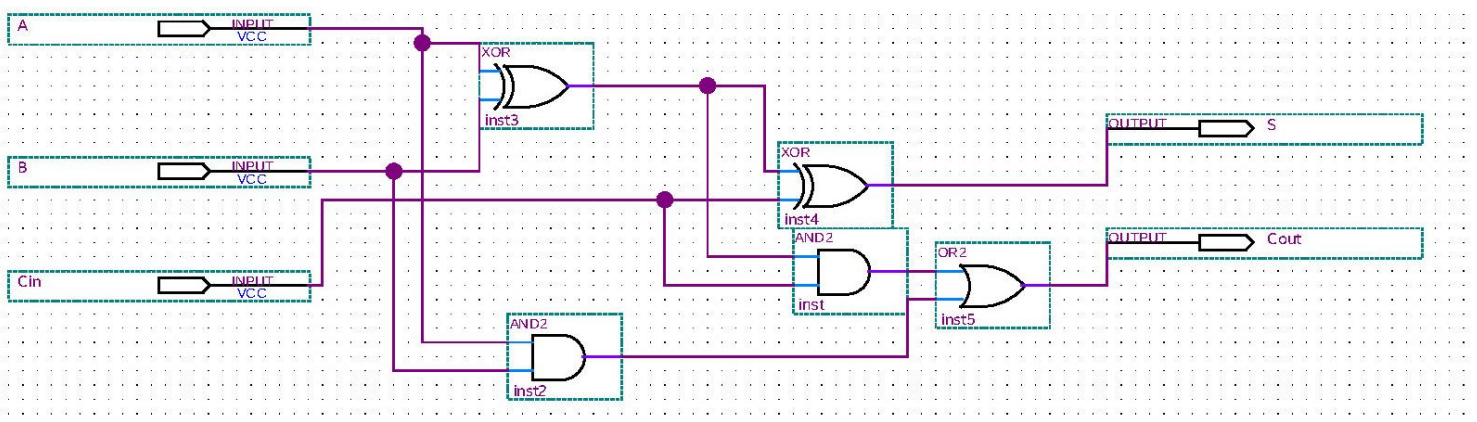
$$S = A'B'Cin + A'BCin' + ABCin + AB'Cin'$$

$$S = (A (+) B) (+) Cin$$

$$Cout = BCin + AB + ACin$$

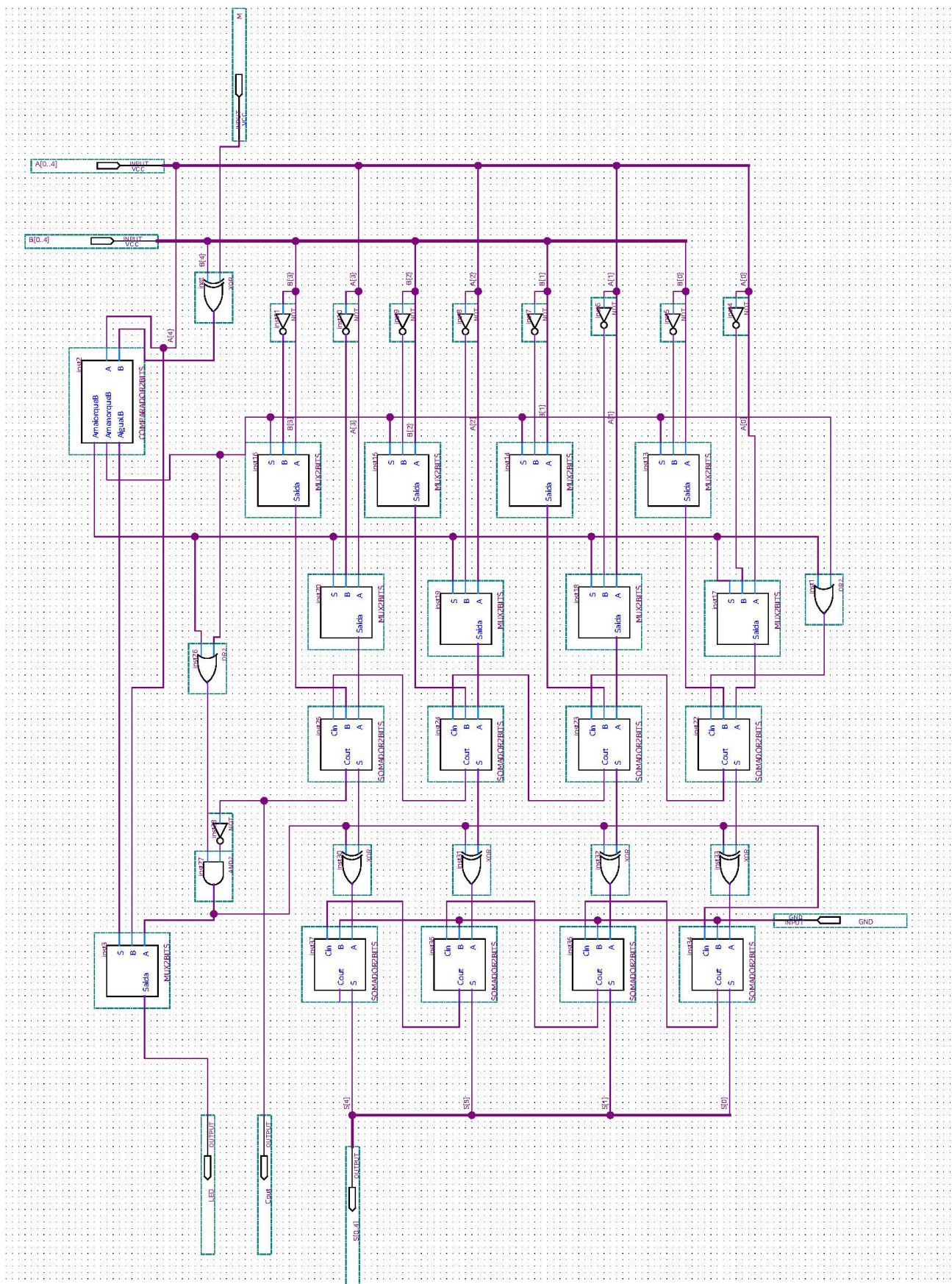
$$Cout = Cin(A(+)B) + AB$$

A partir das funções, foi obtido o seguinte circuito com uso de portas lógicas:



**circuito full-adder de 2 bits.**

Partindo desse circuito, foi possível construir o circuito somador completo de 5 bits :



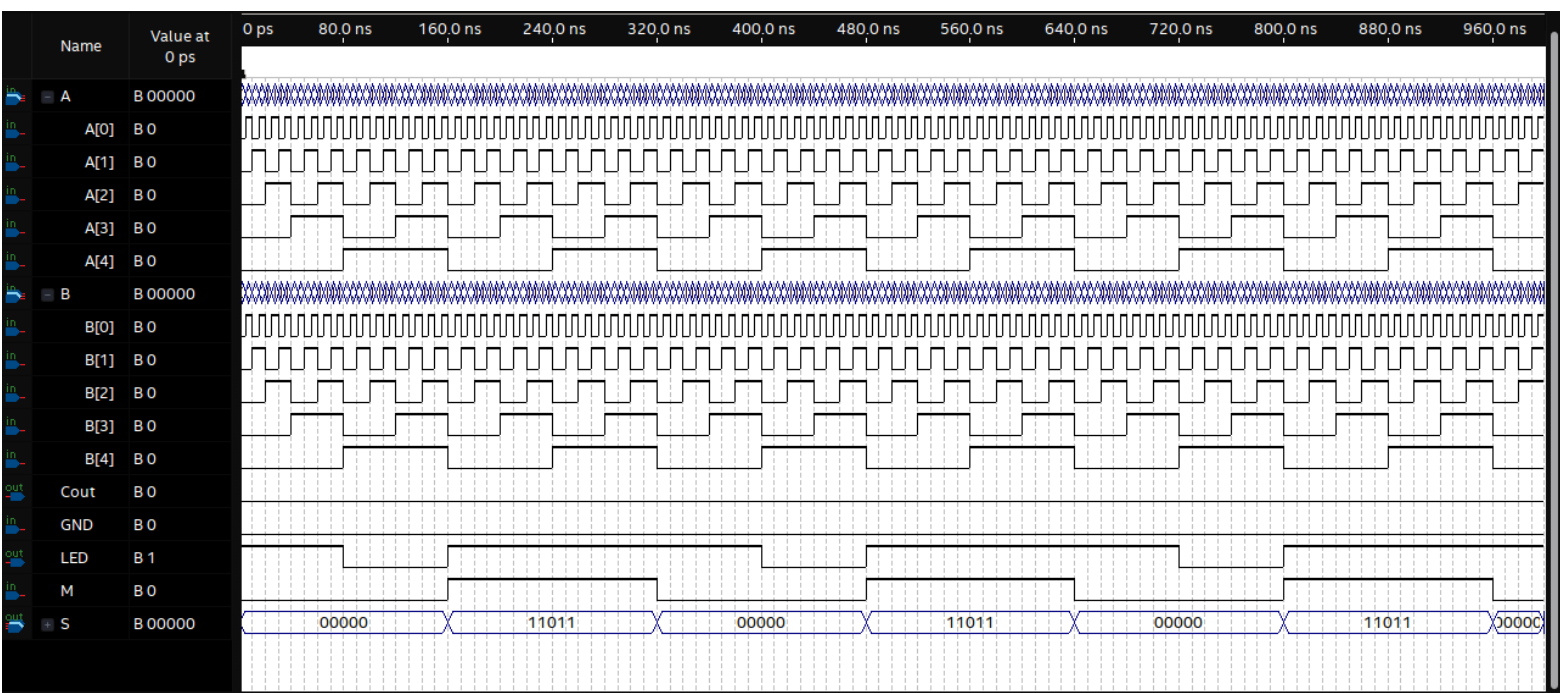
### Implementação do somador de cinco bits.



### Algumas observações sobre o circuito:

- Para cada input do somador, será recebido uma entrada A e uma entrada B, cada uma de 5 bits, sendo 1 bit de sinal e 1 bit de magnitude. A saída do circuito é composta por 5 bits de magnitude (sendo 1 bit o Carry-out) + 1 bit de sinal.
- Sempre que os sinais da entrada forem iguais, seja na soma  $(+)(+)$ , ou na subtração  $(-)(-)$ , o sinal é repetido na saída e é feita a soma.
- O input M é o indicador de modo, caso o modo de subtração seja selecionado, M é 1. então uma operação de soma será realizada; uma porta XOR foi colocada entre M e B, já que em caso dos dois serem 1, B será positivo, já que  $(-)(-) = (+)$ .
- Caso seja realizada uma operação de subtração, a saída será complementada.

### • Simulação no Waveform:





## 3.2 Complementador a 2

Este circuito tem como objetivo, fazer o complemento a dois da entrada B. para complementar a 2 uma cadeia de bits, basta inverter os valores de todos os bits, e por fim, somar 1 bit.

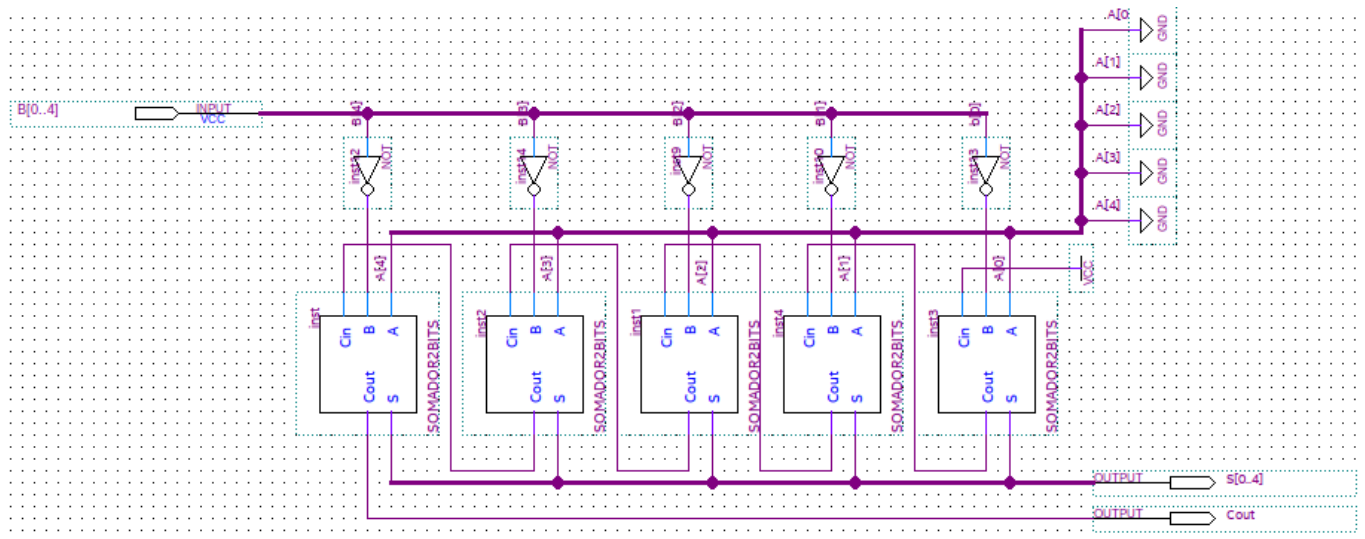
**101011** (invertendo todos os bits) -> **010100**

agora basta somar **1** bit a **010100**

**resultado: 010101.**

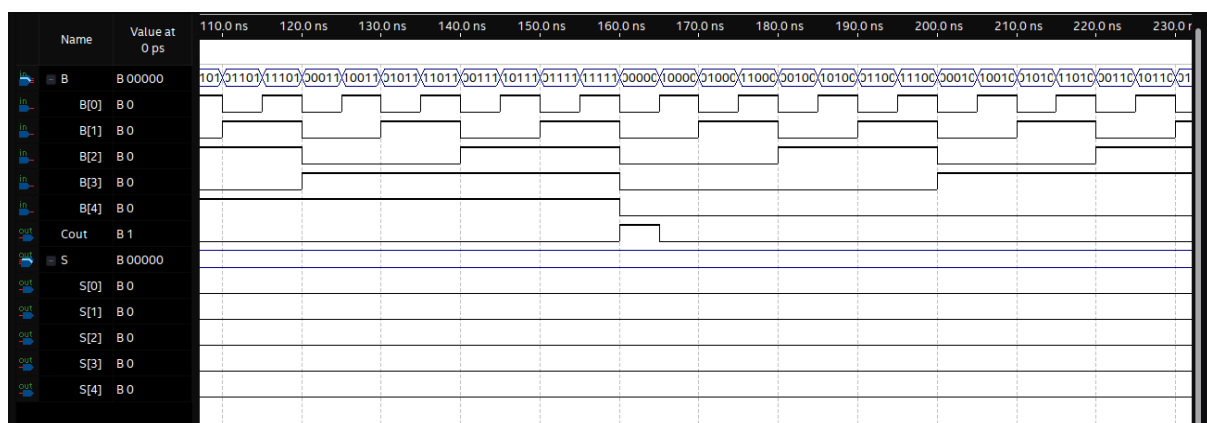
Ou seja, 010101 é o complemento de 2 da cadeia 101011.

Esse processo é feito utilizando um inversor em cada bit de B, depois foram colocados 5 somadores de 1 bit em paralelo, em todos, a entrada A é 0. O Cin do LSB é 1.



**circuito do complementador a 2.**

### Simulação no Waveform:



### 3.3 Circuito comparador

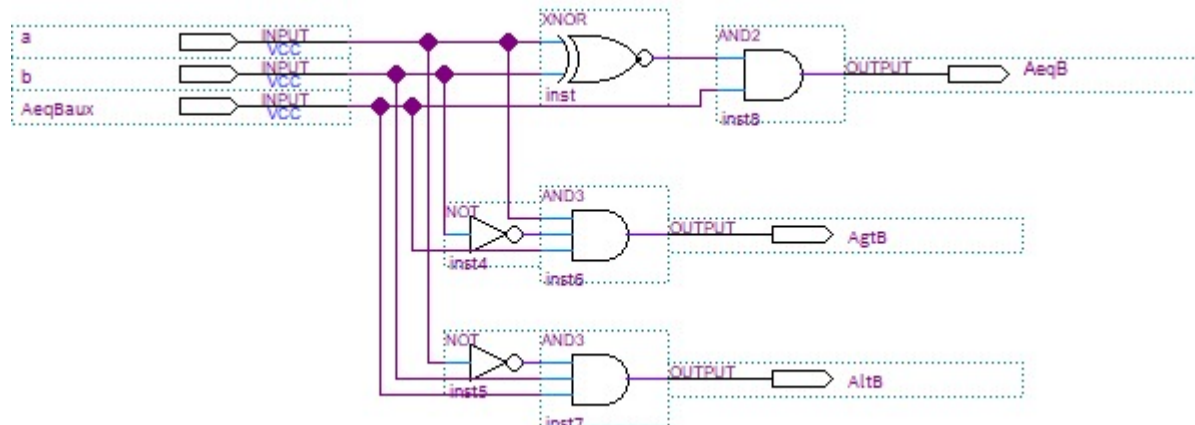
Este módulo tem como objetivo comparar as cadeias de bits e, sobre essa análise, informar se o valor da entrada é igual, maior ou menor, sendo função de quem manipula a ULA o dever de escolher qual funcionalidade e as entradas que devem ser aplicadas. O circuito então dividido em três partes, fica aqui exposto a sua tabela verdade e o circuito digital:

- **Tabela-Verdade:**

A	B	AeqB	AgtB	AltB
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Tabela verdade do circuito comparador de 1 bit

- **Circuito Projetado:**



circuito combinado com as três operações comparativas

Nas operações de maior e menor que, temos alguns detalhes que precisam ser tratados como, “zero negativo” e “zero positivo” que exigia uma saída específica, pois zero é um valor que não permite ser positivo ou negativo, portanto a comparação com saída 1 está errado, outra situação que precisa ser tratado é a comparação com números negativos, que terá um valor invertido, pois, diferente de um número positivo, o número negativo é maior que outro quando ele é “menor”, ou seja, o número mais próximo de zero será o maior.

### 3.3.1 A igual B

Esse circuito tem como objetivo comparar duas cadeias binárias de 5 bits. O circuito básico será composto de um circuito comparador de 1 bit, que quando posto em paralelo com o outros compradores de 1 bit, será capaz de analisar e informar se as duas entradas são iguais.

$\sim(A \oplus B)$

- **Circuito Projetado:**

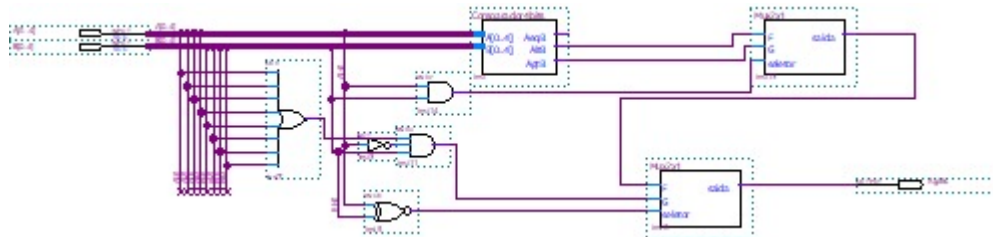


### 3.3.2 A maior que B

Esse circuito tem como objetivo comparar duas cadeias binárias de 5 bits. O circuito básico será composto de um circuito comparador (AltB) de 1 bit, que quando posto em paralelo com o outros compradores de 1 bit, será capaz de analisar se a entrada A é maior do que B comparando os bits da entrada(magnitude) do mais significativo até o último, com base no bit de sinal(O mais significativo).

$A \sim B$

- **Circuito Projetado:**

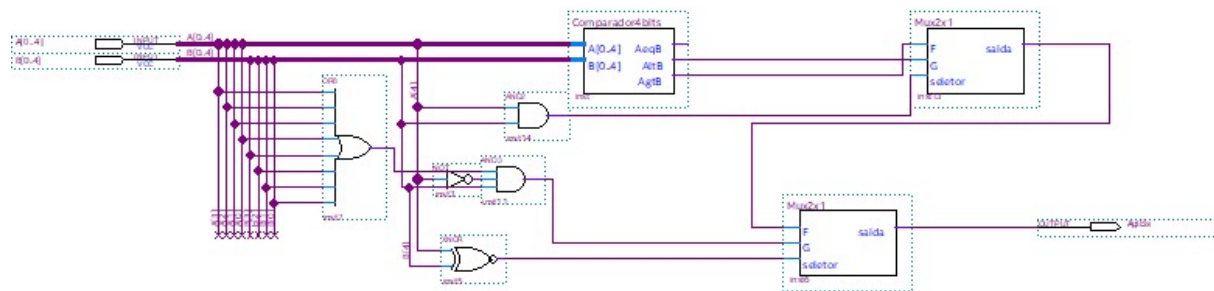


### 3.3.3 A menor que B

Esse circuito tem como objetivo comparar duas cadeias binárias de 5 bits. O circuito básico será composto de um circuito comparador (AltB) de 1 bit, que quando posto em paralelo com o outros compradores de 1 bit, será capaz de realizar a análise se a entrada A é menor do que B comparando os bits(magnitude) com base no bit de sinal(O mais significativo).

$\sim A.B$

- **Circuito Projetado:**



## 3.4 AND e XOR

### 3.4.1 Função AND:

O circuito da função AND tem como objetivo performar a operação de AND bit a bit de dois vetores A e B (inputs) e retornar como saída um vetor S (output) como o resultado de cada operação AND concatenado no vetor.

Cada bit de cada vetor passa por uma porta AND.

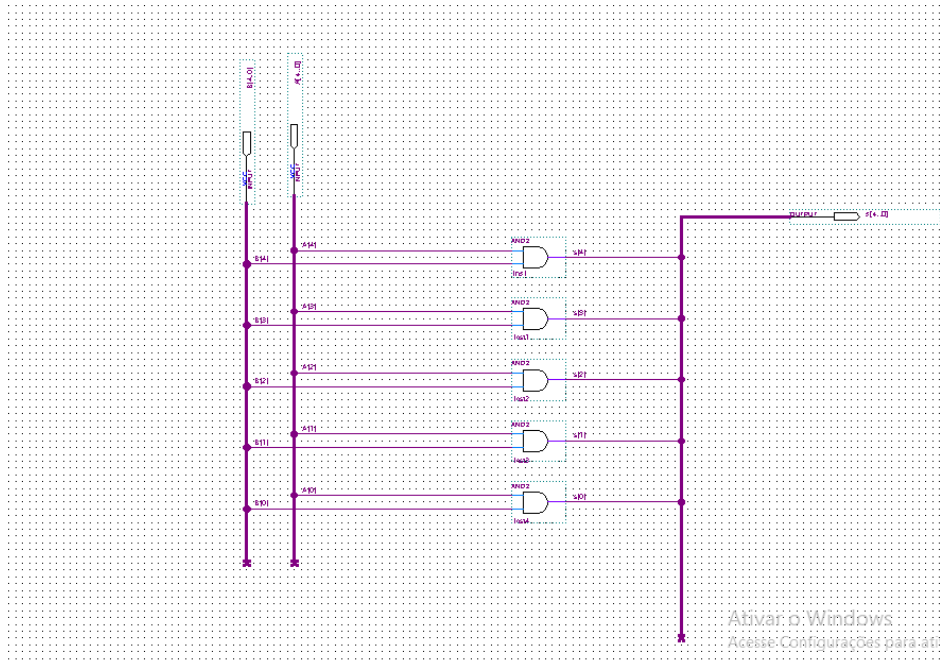
Ex:  $A[0] \text{ AND } B[0] \Rightarrow S[0]$

- **Tabela-Verdade:**

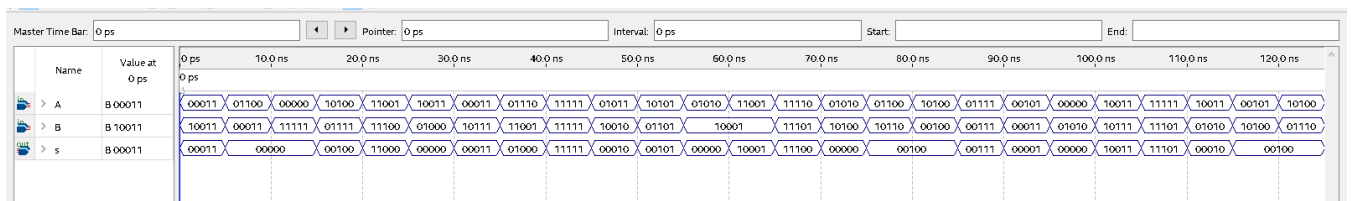
A[i]	B[i]	S[i]
0	0	0
0	1	0
1	0	0
1	1	1

**Tabela verdade da operação A and B bit a bit.**

- **Circuito Projetado:**



- **Waveform:**



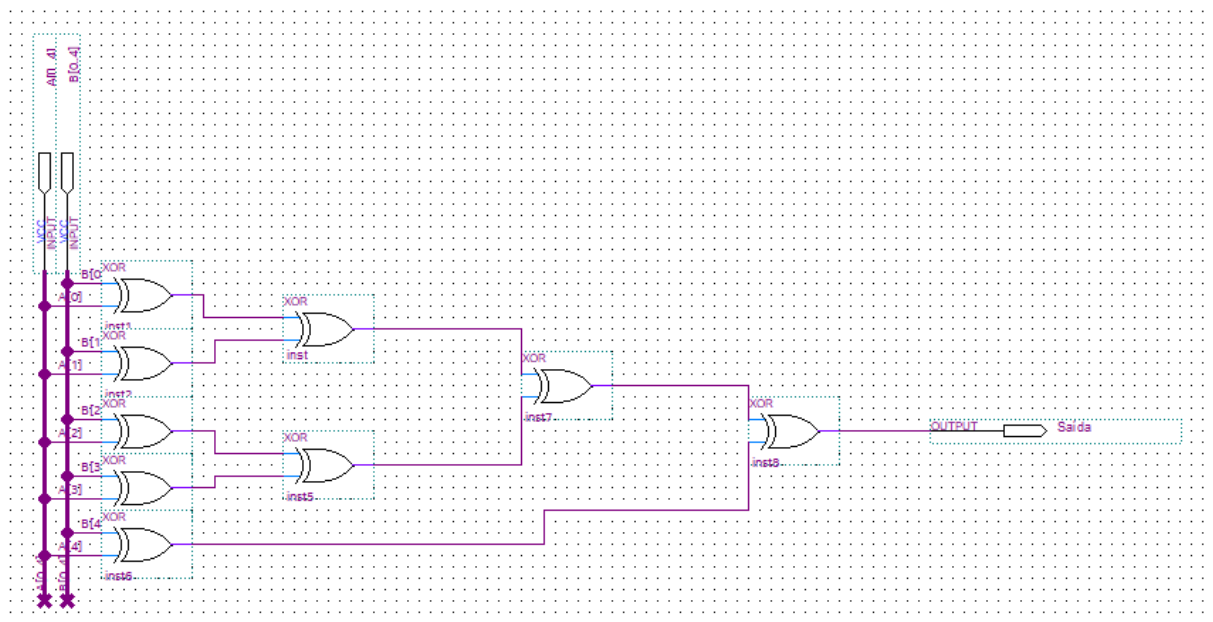
### 3.4.1 Função XOR:

O Circuito XOR recebe como entrada dois vetores (A e B) de tamanho 5, para cada bit de um vetor A paralelo ao B se utiliza uma operação XOR (exemplo: A[0] é paralelo a B[0]), repete as operações XOR até retornar um único bit.

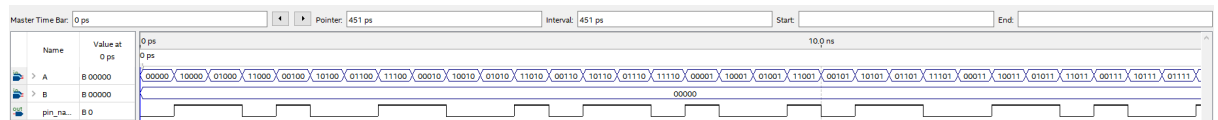
- **Tabela-Verdade:**

A[i]	B[i]	S[i]
0	0	0
0	1	1
1	0	1
1	1	0

- **Circuito Projetado:**



- **Waveform:**



### 3.5 Multiplexadores

Tendo em vista que o projeto pede que seja feita uma seleção a partir de uma entrada seletora de 3 bits, o uso de multiplexador é indispensável, uma vez que servem como chaves seletoras. No projeto foram utilizados **Mux 2:1 1 bit**, **Mux 8:1 1 bit** e **Mux 8:1 5 bits**.

- **Tabela-Verdade:**

s2	s1	s0	Y
0	0	0	i0
0	0	1	i1
0	1	0	i2
0	1	1	i3
1	0	0	i4
1	0	1	i5
1	1	0	i6
1	1	1	i7

**Tabela-Verdade Mux 8:1**

s	Y
0	i0
1	i1

**Tabela-Verdade Mux 2:1**



### 3.5.1 Mux 2:1 1 bit

Utilizado em diversas operações, compondo alguns módulos como artifício fundamental para selecionar ,a partir de uma chave de 1 bit, uma dentre duas entradas.

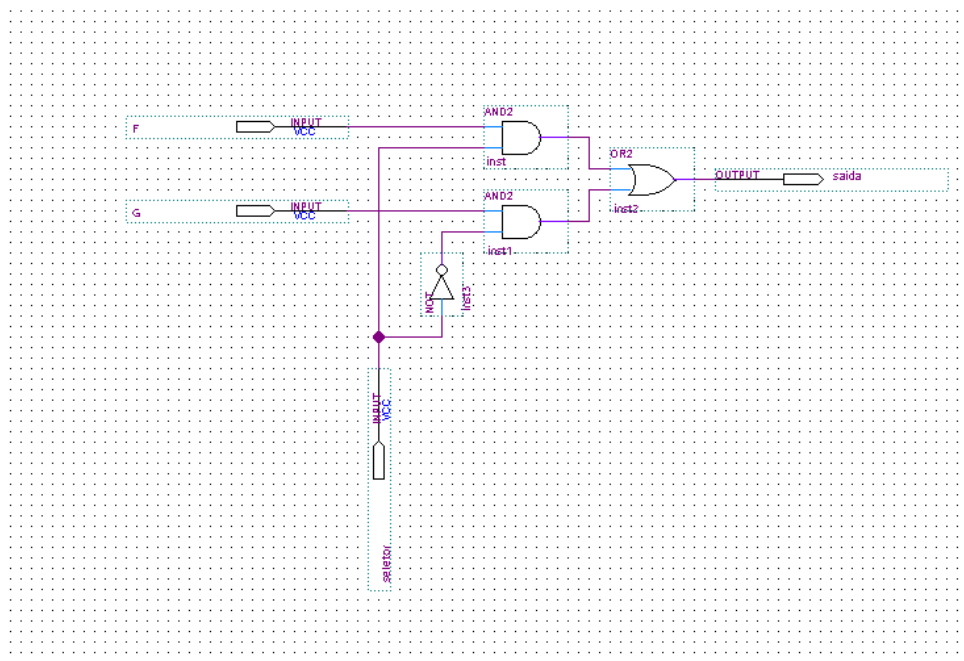
Exemplo:

$s(\text{chave seletora})=1 \Rightarrow i0$  (input 0)

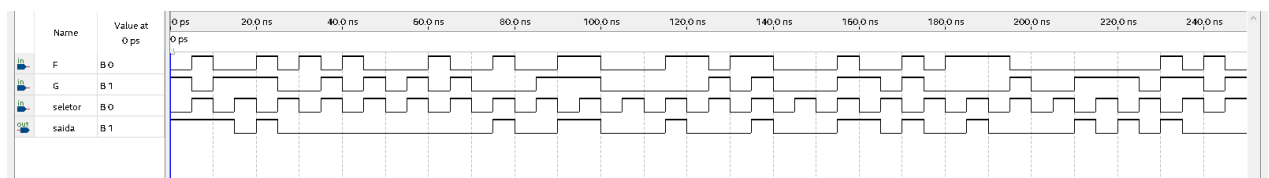
$s(\text{chave seletora})=0 \Rightarrow i1$  (input 1)

De construção simples, utilizando de duas portas AND que recebe cada uma um INPUT e um bit seletor(0 ou 1), e uma porta OR que de acordo com a combinação passará adiante o valor desejado.

- Circuito Projetado:**



- Waveform:**



### 3.5.2 Mux 8:1 1 bit

Utilizado dentro do Mux 8:1 5 bits para multiplexar bit a bit de cada entrada do Mux 8:1 5 bits. Recebe como entrada em um Mux 2:1 1 bit os bits de cada vetor de entrada, como exemplo:

Mux1 = vetorA[4],vetorB[4]

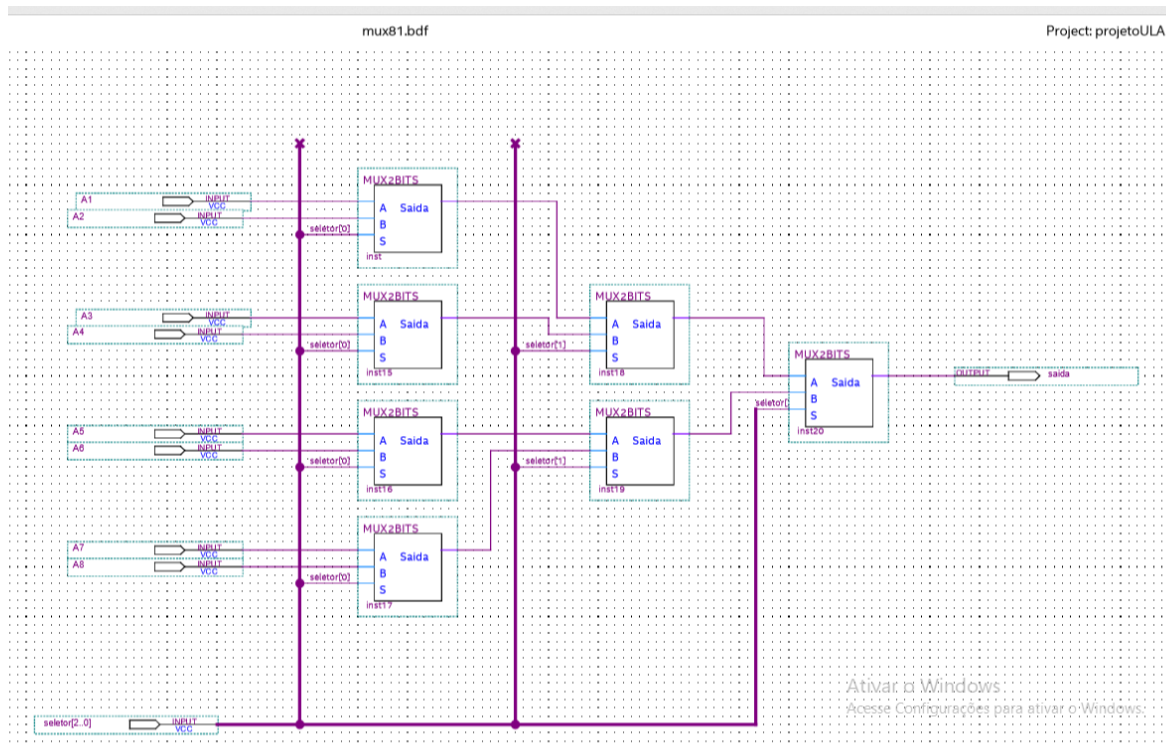
Mux2 = vetorC[4],vetorD[4]

...

A chave seletora é dividida em estágios, uma vez que o Mux 2:1 só recebe a chave de tamanho 1 bit, logo temos 3 estágios de seleção. O primeiro estágio consiste em 4 Mux 2:1, com o bit seletor = s0. O segundo estágio consiste em 2 Mux 2:1, com o bit seletor = s1. O terceiro estágio consiste em um único Mux 2:1, com o bit seletor = s2.

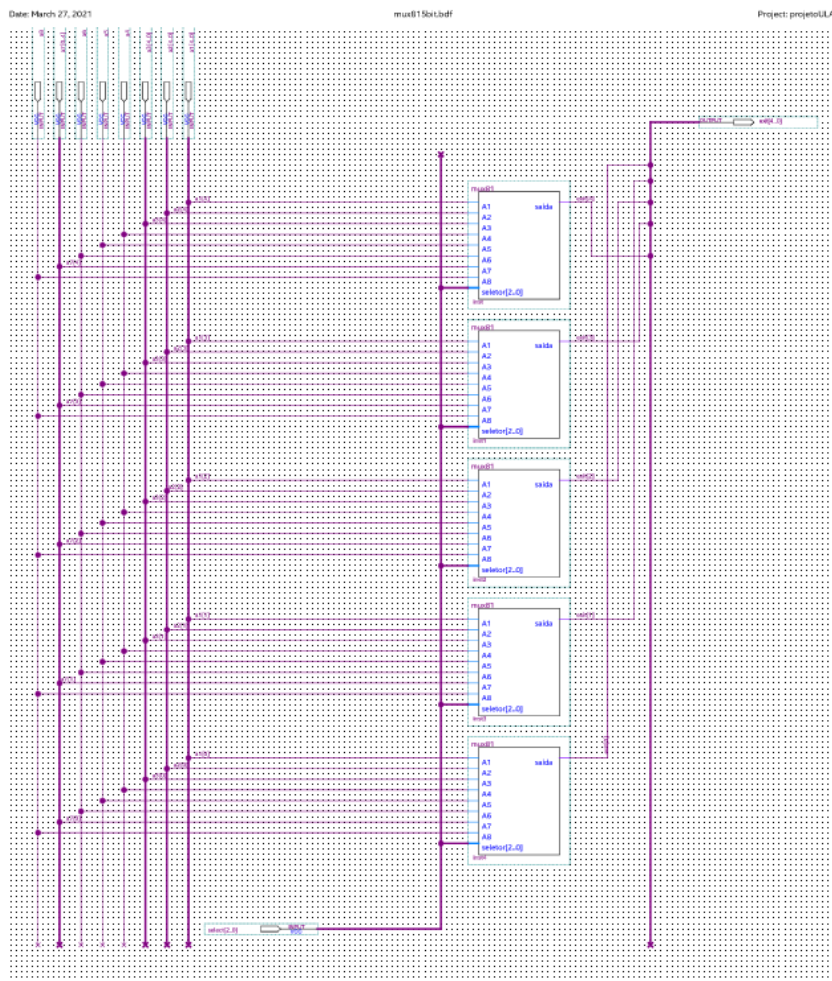
Cada Mux 2:1 multiplexa os bits e passa para o próximo estágio de seleção, restando no final a saída de 1 bit que consistirá em um bit do vetor saída do Mux 8:1 5 bits.

- Circuito projetado:

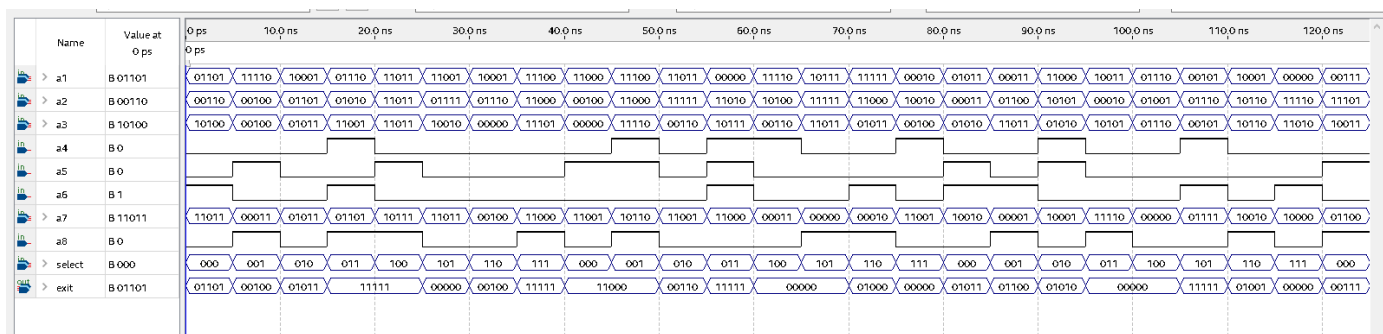




## ● Circuito Projetado:



## ● Waveform:



## 4 Conclusão

O projeto de construção da ULA exige uma organização estrutural de modo que incentive o estudo e a pesquisa acerca dos módulos a serem desenvolvidos e interligados, esse se mostrou o principal desafio. A modularização torna cada instância do projeto uma parte essencial de um todo.

O software Quartus auxilia bastante na praticidade e na facilidade de visualizar o que estivesse sendo desenvolvido. Além disso, a possibilidade de reutilizar os módulos já feitos na forma de “black box” deixa a construção da ULA bem mais intuitiva. Apesar da experiência prática e visual ter sido considerada como um dos pontos positivos da utilização do software, alguns problemas e bugs acabaram por frustrar um pouco a experiência, uma vez que alguns desses problemas requerem uma certa prática e familiarização com o Quartus para saber como proceder.

Por fim, é importante salientar que o projeto serviu como grande incentivo para o estudo e aprendizado do conhecimento adquirido até então na disciplina de Sistemas Digitais, uma vez que instigou a pesquisa e a prática aplicada a tais conhecimentos. É válido afirmar que o conhecimento adquirido e desenvolvido durante o projeto com certeza será útil em diversos âmbitos do curso, bem como na esfera profissional.