

**UNIVERSIDADE FEDERAL DE PERNAMBUCO  
SISTEMAS DIGITAIS**

Eduardo Gabriel de Souza Pedroza(egsp)  
Mateus Rafael Correia de Melo(mrcm)  
Matheus Enrico Carmo dos Santos Barros(mecsb)  
Ryann André Dias da Silva(rads)

**Projeto da ULA - Equipe 9**

Recife, PE

2025

## SUMÁRIO

<b>1 VISÃO GERAL.....</b>	<b>3</b>
<b>2 TABELAS DA VERDADE E CÁLCULOS.....</b>	<b>9</b>
<b>3 CIRCUITO PROJETADO DE CADA MÓDULO E SIMULAÇÃO.....</b>	<b>10</b>
<b>4 CIRCUITO COM TODO SISTEMA CONECTADO E SIMULAÇÃO.....</b>	<b>29</b>
<b>5 CONCLUSÃO.....</b>	<b>30</b>

## 1 VISÃO GERAL

Aqui está o PDF, do circuito da ULA completa:

 [uladefinitiva.pdf](#)


### SOMADOR:

Foram usados dois condicionadores de sinal, que preparam o numero para a soma, transformando ele em uma representação em complemento de 2 apenas se for negativo.

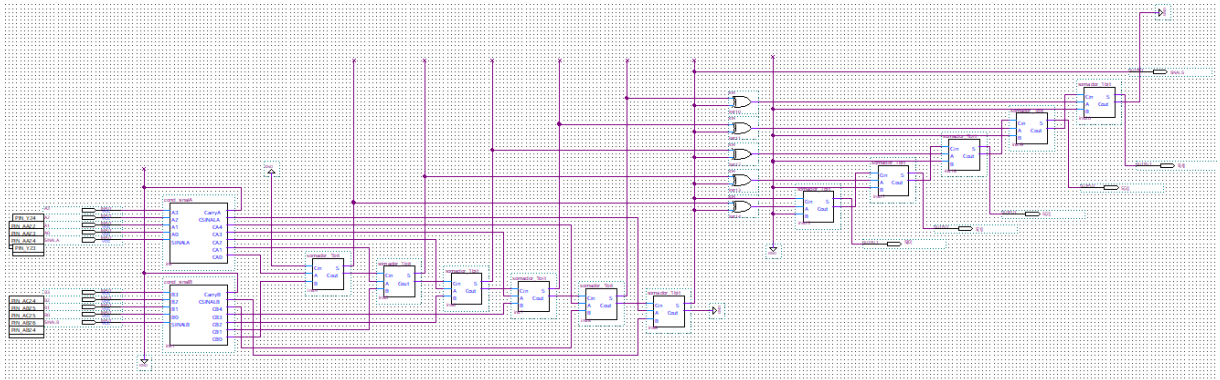
As portas XOR decidem se o bit vai ser invertido ou não de acordo com o bit de sinal. Analisando a tabela verdade da porta XOR, quando o bit de sinal é 0, a saída vai ser o proprio bit do input. Quando a saída for 1, o bit é invertido. Exemplo  $1 \text{ XOR } 1 = 0$ .

Após esse processo de inversão, formando o complemento de 1 se o numero for negativo, é feita a soma com 000X, sendo X o bit de sinal, que entra como Carry IN do primeiro somador de 1 bit. Se o numero for positivo ( $X = 0$ ), o número vai ser somado com 0000 e portanto não vai ser alterado.

XOR		
A	B	$X=(A \oplus B)$
0	0	0
0	1	1
1	0	1
1	1	0



Depois do processo de condicionamento de sinal, é feita a soma de forma padrão e o resultado (caso seja negativo estará sendo representado em complemento de 2), passa por um novo condicionador para ser reconvertido para sinal-magnitude.

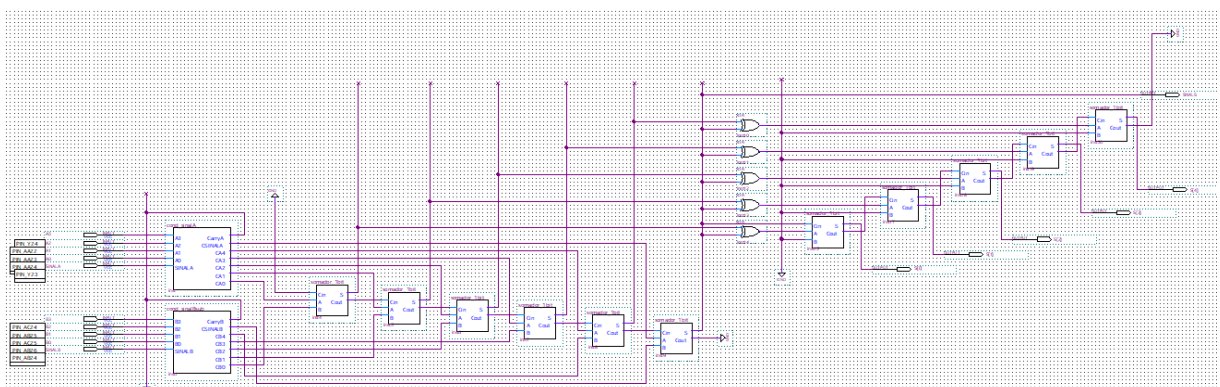


### Subtrator:

O processo é quase idêntico ao do somador, mudando apenas o condicionador de sinal do B, que agora faz o processo inverso: complemento de 2 se o numero for POSITIVO e apenas repete trocando o sinal se for negativo. Foi decidido fazer dessa maneira já que, por exemplo, se a operação for  $x - (-y)$  acaba virando uma soma de  $x + y$ , então basta repetir o y, trocando seu bit de sinal para 0 e realizar a soma.

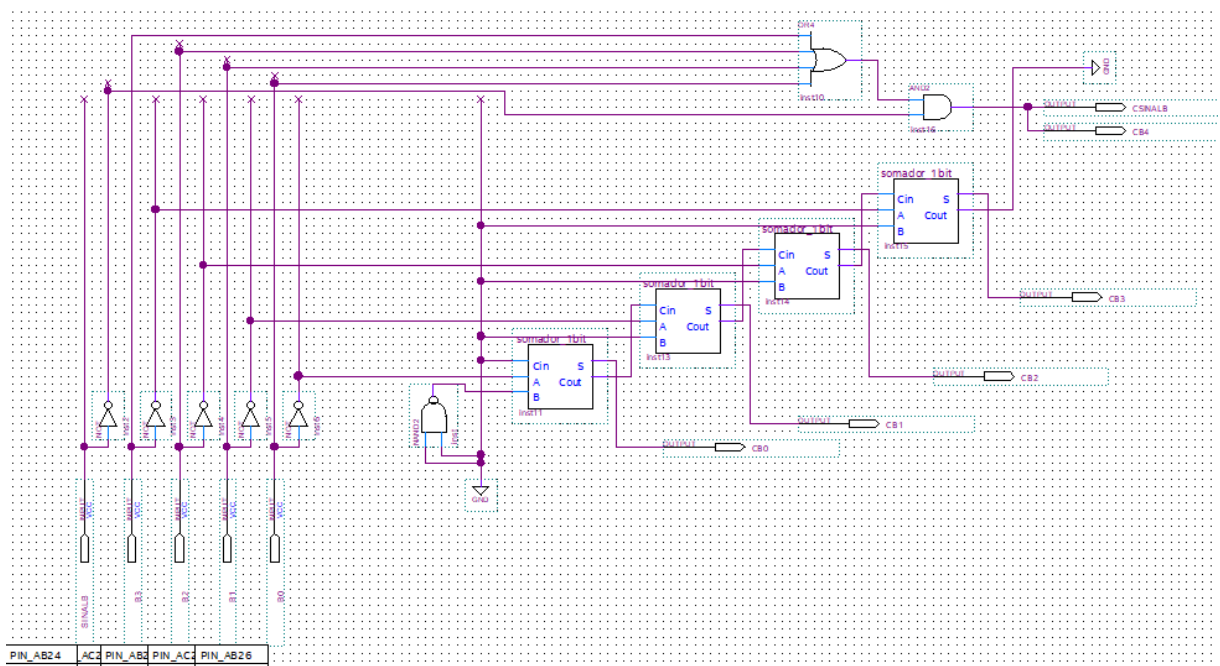
Agora a porta lógica usada para decidir a conversão para comp2 não é mais a XOR, sim a XNOR, que agora faz o processo de inverter caso o bit de sinal for 0.

XNOR		
A	B	$X = (\overline{A \oplus B})$
0	0	1
0	1	0
1	0	0
1	1	1



## Comp2B:

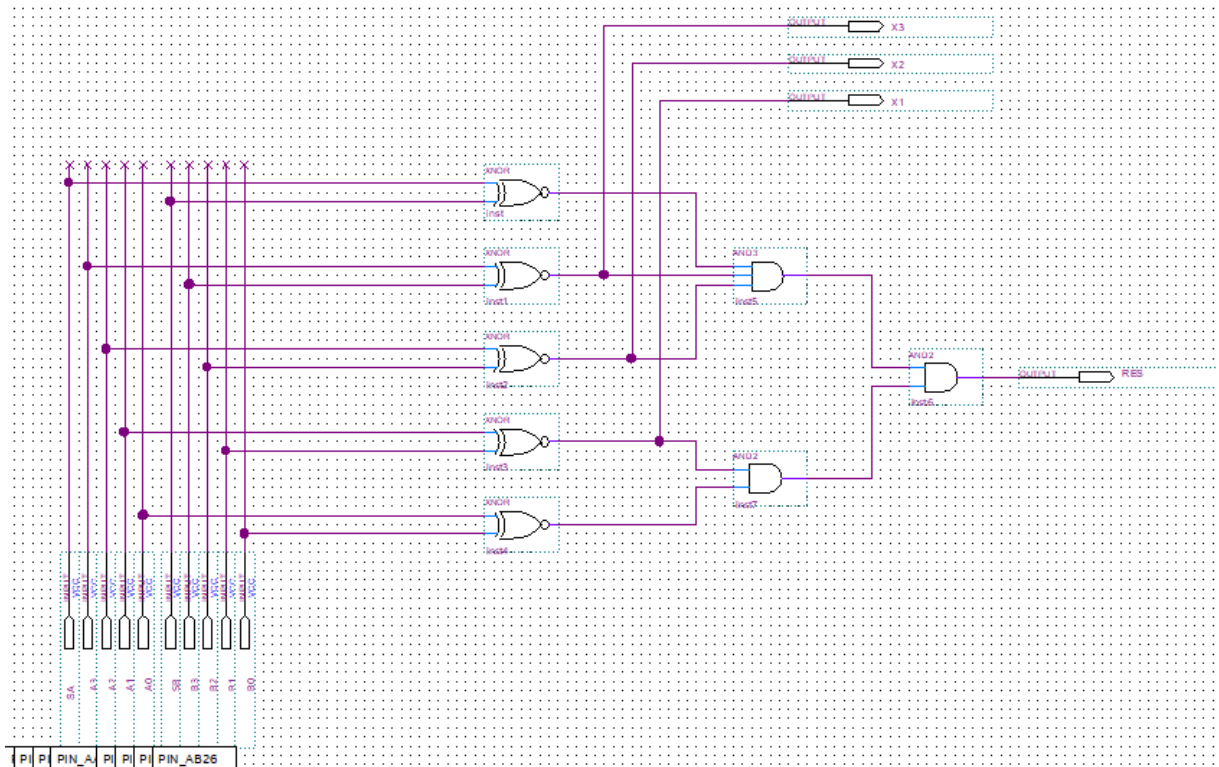
Inverte todos os bits da magnitude e soma 0001. Foi feita uma lógica pra que o bit de sinal so se inverte caso a magnitude for diferente de 0 e caso a magnitude for igual a 0, o bit de sinal também vai ser sempre 0, para evitar uma representação de -0.



## Igualdade:

Foi feita apenas a operação de XNOR bit a bit, ja que essa porta retorna 1 caso os bits forem iguais.

XNOR		
A	B	$X=(\overline{A \oplus B})$
0	0	1
0	1	0
1	0	0
1	1	1



## 7 SEGMENTOS:

Foi feita a tabela verdade com 31 possibilidades (0 à 30) e a partir da tabela foram geradas equações simplificadas para cada segmento através dos mapask

PHOTO-2025-06-26-13-42-18.jpg

display das dezenas:

DISPLAY6BITS0\_OUT

display das unidades:

DISPLAY6BITS1\_OUT

## MAIOR QUE:

A lógica principal deste bloco foi integrar operações lógicas para determinar a relação "maior que". Um valor B é considerado \*\*maior que A\*\* se, e somente se, o bit de B for 1 e o bit de A for 0 em uma dada posição, considerando a comparação bit a bit.

A verificação ocorre assim: para cada par de bits (por exemplo, A0 e B0), se A0 for 1 e B0 for 0, essa condição indica que o bit de A é "maior" nessa posição. Essa informação é então invertida e enviada para uma porta AND.

Para bits de ordem superior (como A1-3 e B1-3), as saídas do bloco "Igual" são utilizadas. Se os bits de ordem mais alta (por exemplo, A3 e B3) são iguais, a lógica então verifica se os bits seguintes (A2 e B2) são diferentes, e assim sucessivamente.

Portas NOT, AND, OR, XOR e XNOR foram empregadas. As portas NOT foram usadas nas entradas SA (Sinal de A), SB (Sinal de B) e nas entradas B0-B3.

As comparações bit a bit, como A0-B0, foram processadas por portas AND de duas entradas. Comparações adicionais para entradas de ordem superior (A3-B3, A2-B2, A1-B1) utilizaram uma entrada extra do bloco "Igual".

### **MENOR QUE:**

O resultado para "menor que" é derivado da lógica de "igualdade" e "maior que". Utilizamos uma porta XNOR com as saídas "maior que" e "igual". A saída de "menor que" será 1 apenas quando as saídas de "maior que" e "igual" forem ambas 0, 0.

### **OPERAÇÕES AND XOR:**

Ambas as operações, AND e XOR, foram implementadas utilizando um método bit-wise, que compara os bits individualmente e retorna uma saída para cada comparação. As saídas de sinal (SA e SB) também foram comparadas dessa forma.

### **MUX DE PALAVRA DE 6 BITS:**

 mux palavra de 6 bit.pdf

O componente denominado MUX2 foi desenvolvido com a finalidade de controlar qual sinal será exibido no display ou nos LEDs. Sua principal função é selecionar, por meio dos sinais de controle, se o resultado da operação (soma ou subtração) será exibido no display; caso contrário, o valor correspondente será direcionado para os LEDs.

A necessidade de um multiplexador com 8 entradas e 1 saída, em que cada entrada e a saída possuem 6 bits, motivou a criação do MUX2. Para isso, o MUX2 foi

implementado como um conjunto de 6 multiplexadores menores, cada um com 8 entradas e 1 saída de 1 bit. Dessa forma, combinando esses 6 MUXs, foi possível obter um único multiplexador com 8 entradas de 6 bits e 1 saída de 6 bits.

O MUX2 utiliza três sinais de seleção (S0, S1 e S2), os quais são compartilhados entre todos os 6 MUXs menores, garantindo assim a seleção coordenada e simultânea dos bits correspondentes de cada palavra de 6 bits.

## **IMPLEMENTAÇÃO E INTEGRAÇÃO DE CIRCUITO LÓGICO:**

- Detalhes da Arquitetura e Conectividade dos Blocos:

A arquitetura do circuito foi desenvolvida com foco na modularidade e na capacidade de realizar diversas operações lógicas e aritméticas. As conexões foram estabelecidas da seguinte forma:

- Processamento Lógico (XOR, AND, Complemento de 2B)

Entradas: Todos os bits dos operandos A e B, incluindo os bits de sinal, foram conectados aos blocos XOR e AND. Os bits do operando B, incluindo o bit de sinal, foram conectados ao bloco de Complemento de 2B.

Saídas: As saídas dos blocos XOR, AND e Complemento de 2B foram direcionadas para as entradas de um primeiro Mux 8x1 de 6 bits. Este Mux é controlado pelos seletores S2, S1 e S0. As entradas não utilizadas neste Mux foram conectadas ao terra (GND).

- Processamento Aritmético (Somador e Subtrator)

Entradas: Ambos os blocos, Somador e Subtrator, recebem todos os bits dos operandos A e B, incluindo os bits de sinal.

Saídas: As saídas do Somador e do Subtrator são conectadas a um segundo Mux 8x1 de 6 bits. Similarmente ao primeiro Mux, este é controlado pelos seletores S2, S1 e S0, e as entradas não utilizadas foram conectadas ao GND.

- Ramificação e Saídas Finais

Saída do Segundo Mux (Somador/Subtrator): A saída deste segundo Mux se ramifica em duas direções:

Uma ramificação vai para um decoder, que representa o resultado da operação aritmética selecionada.

A outra ramificação é direcionada para um terceiro Mux 8x1 de 6 bits.



Terceiro Mux 8x1 de 6 bits: Este Mux recebe a saída do Mux de operação de adição/subtração e a saída do Mux que combina as operações AND, XOR e Complemento de 2B. As entradas não utilizadas também são conectadas ao GND. As saídas deste terceiro Mux são conectadas aos LEDs para visualização do resultado.

- Operadores de Comparação de Magnitude

Bloco de Magnitude: Os operadores de comparação (maior que, menor que e igual) estão contidos dentro de um bloco de Magnitude.

Mux 8x1 de 1 bit para Magnitude: A saída do bloco de Magnitude é conectada a um Mux 8x1 de 1 bit. Este Mux é controlado pelos seletores S2, S1 e S0, e suas entradas não utilizadas são conectadas ao GND.

Visualização da Magnitude: A saída deste Mux de 1 bit é conectada a um único LED, que indica o resultado da comparação de acordo com o operador selecionado.

- Decodificação de Entradas

Para fins de visualização ou depuração, todos os bits do operando A são conectados a um decoder e todos os bits do operando B são conectados a outro decoder.


## 2 TABELAS DA VERDADE E CÁLCULOS

Todos os mapas K e a Tabela Verdade, estão nesse link:

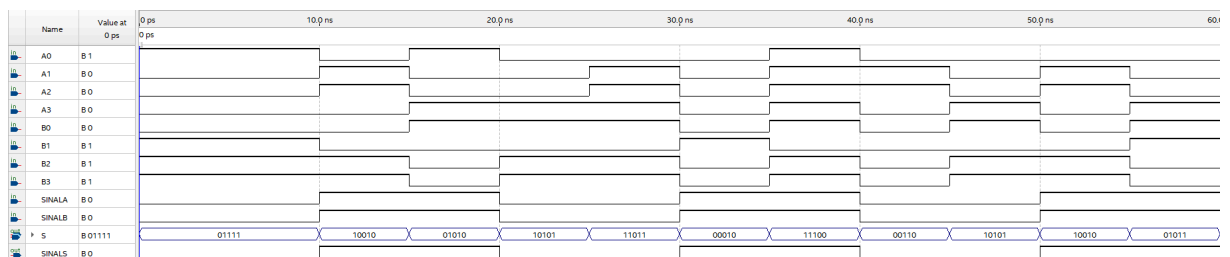
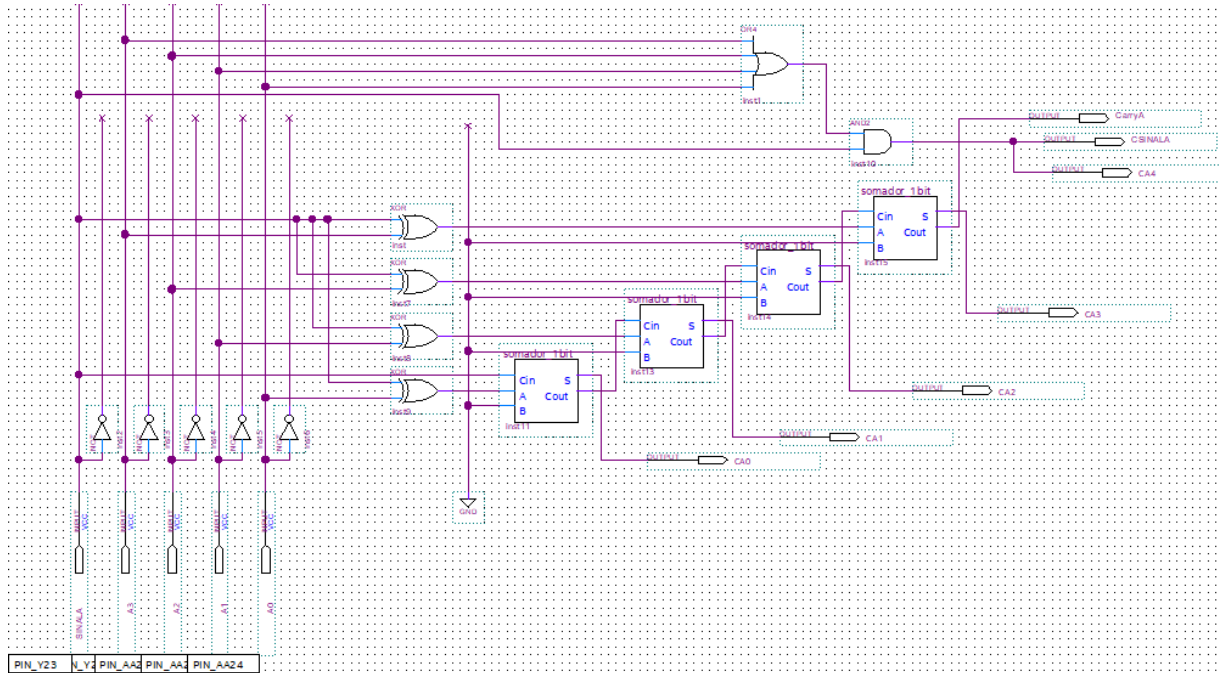
 [PRINTS\\_PROJETO](#)

### 3 CIRCUITO PROJETADO DE CADA MÓDULO E SIMULAÇÃO

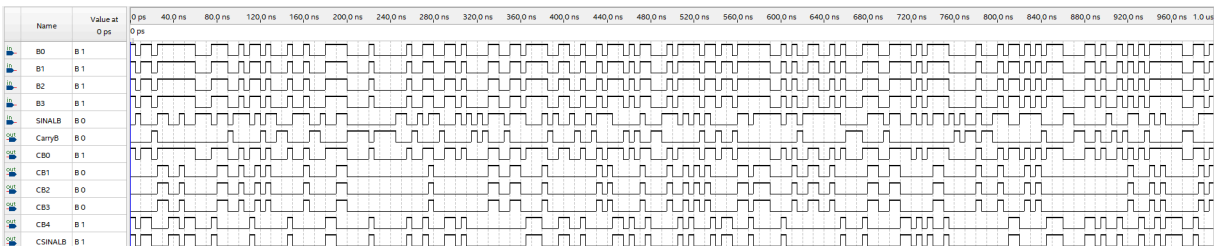
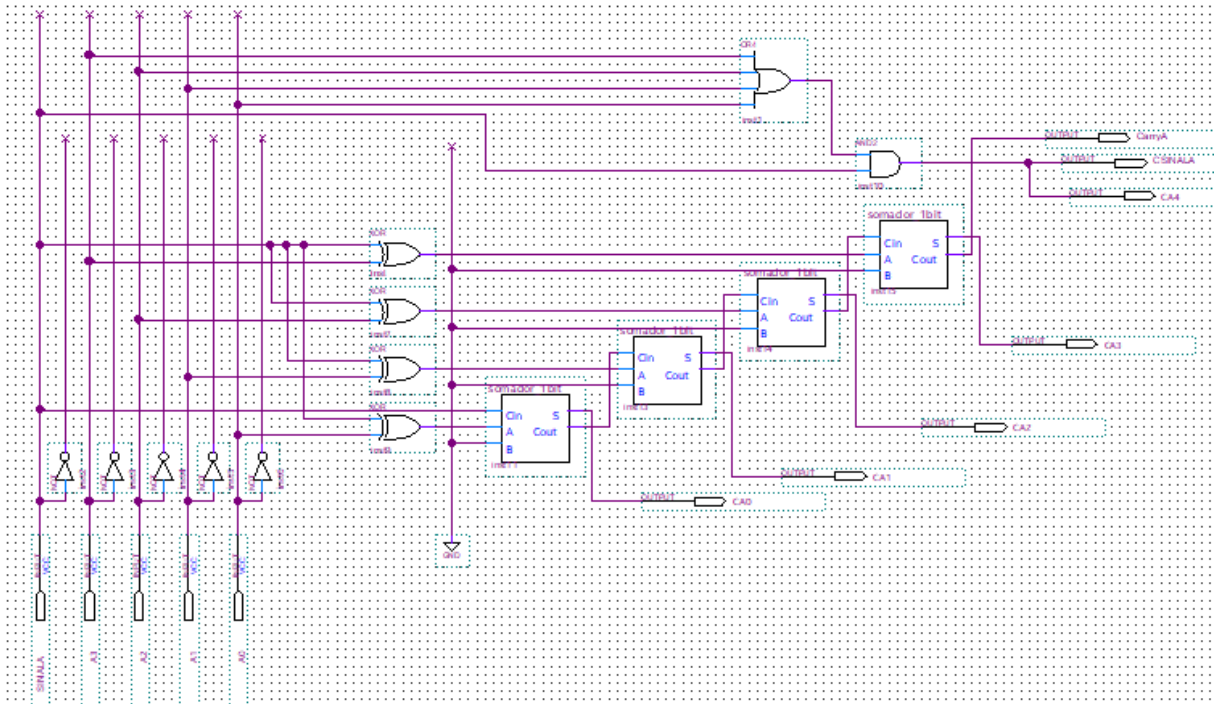
Todos os Circuitos, estão nesse link:

 Modulos

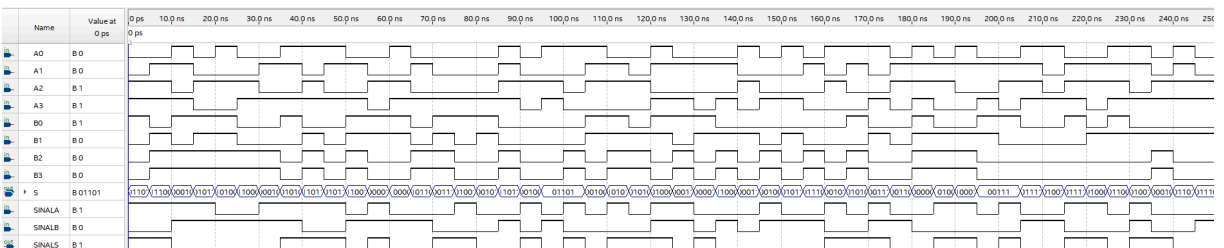
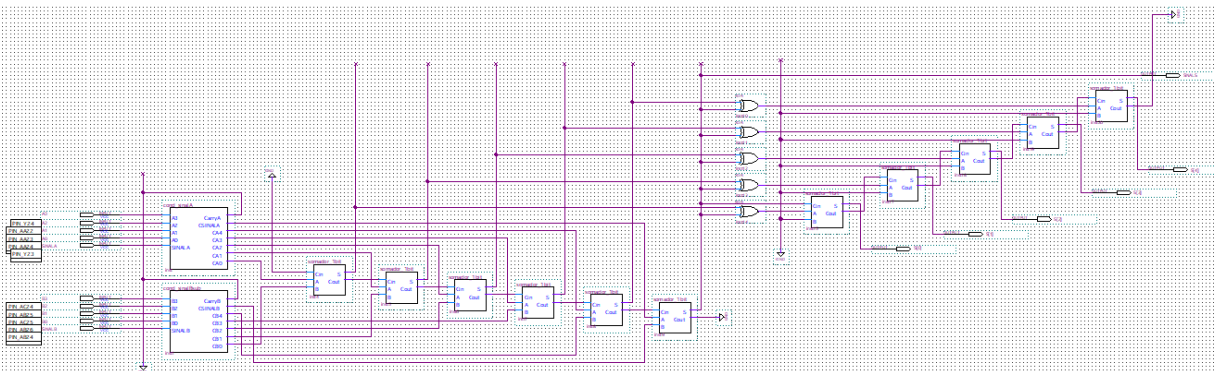
#### Somador:



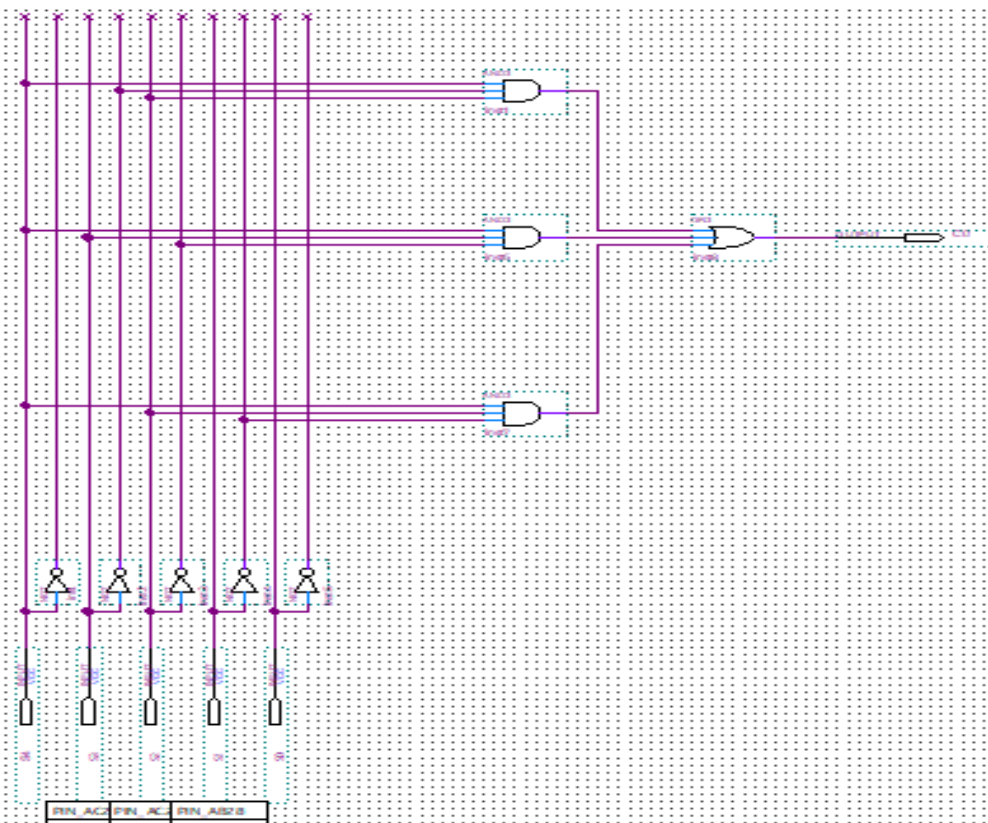
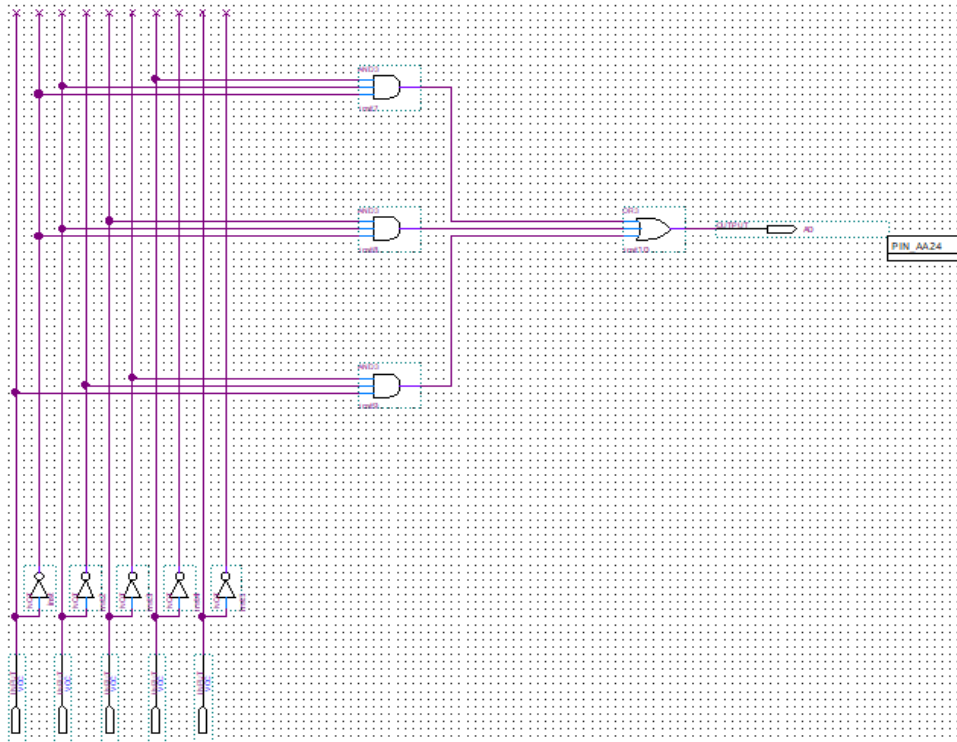
#### Condição de Sinal A-B

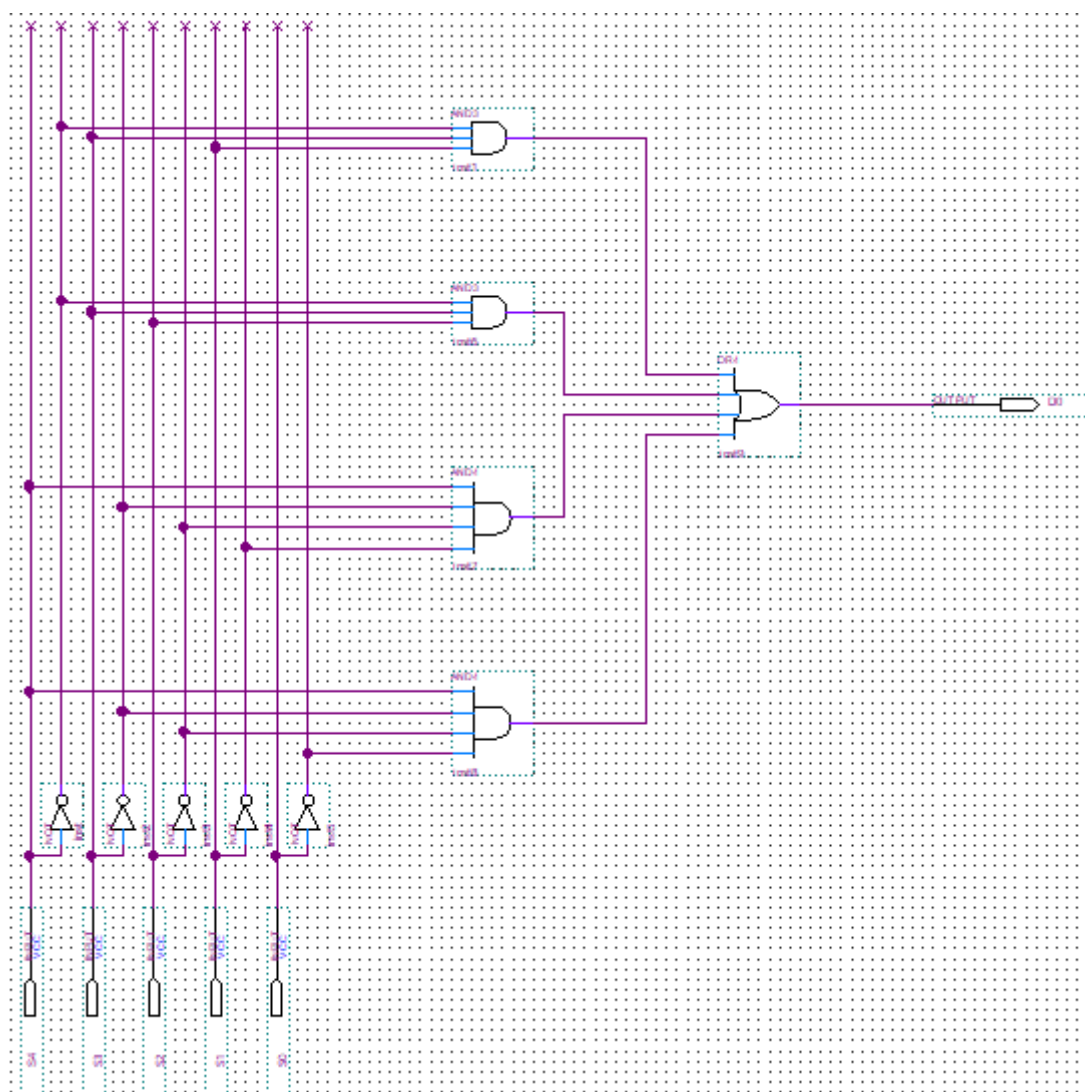


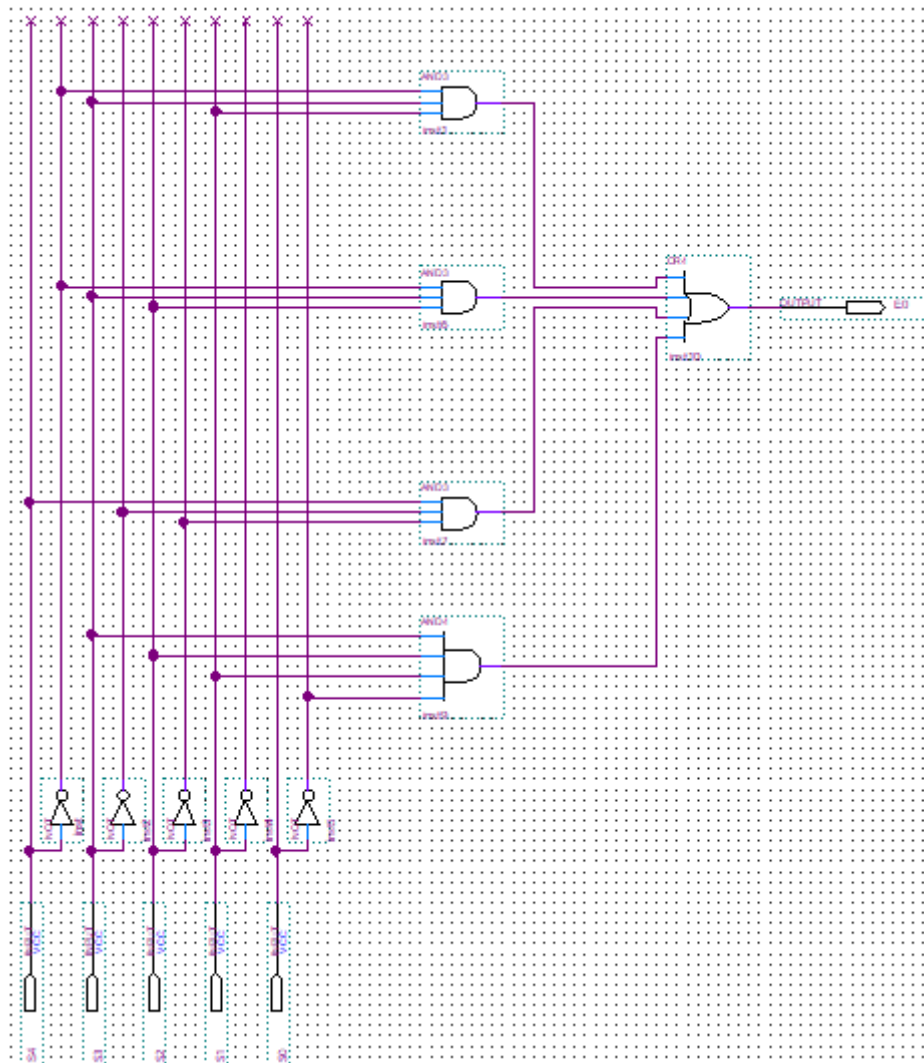
## Subtrator

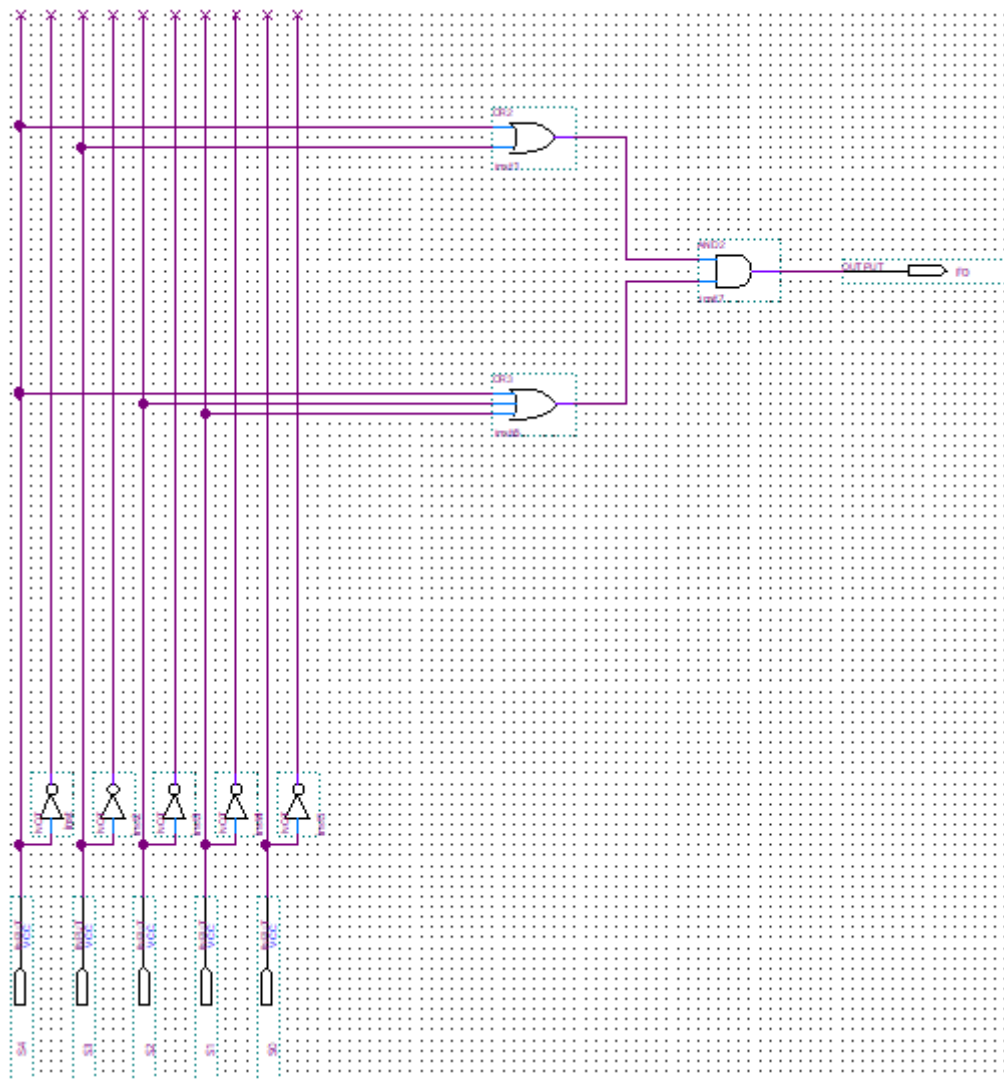


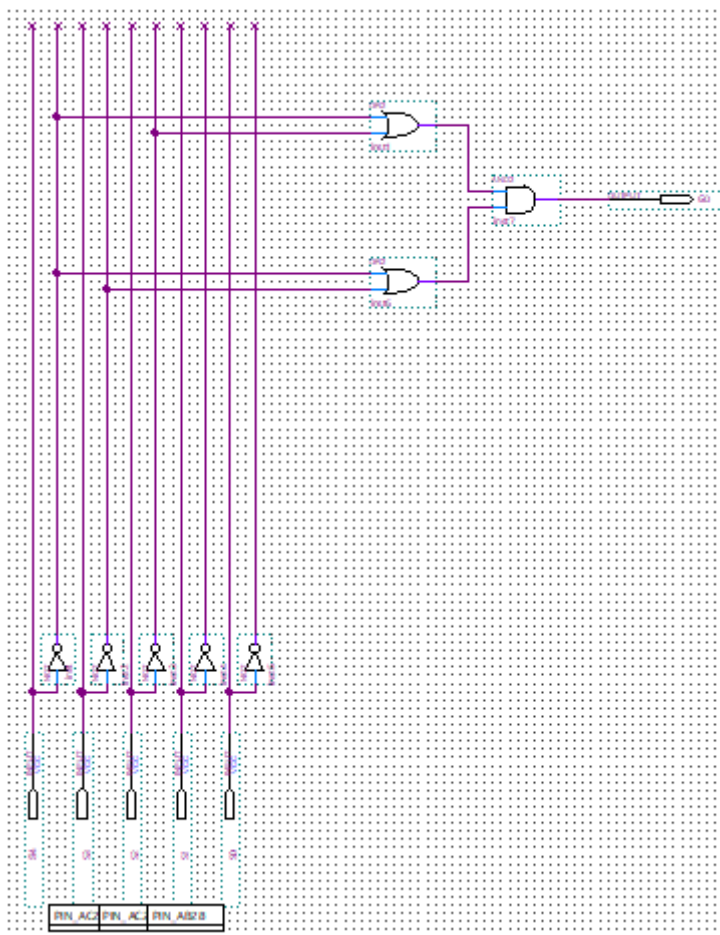
## Segmentos A0-G0





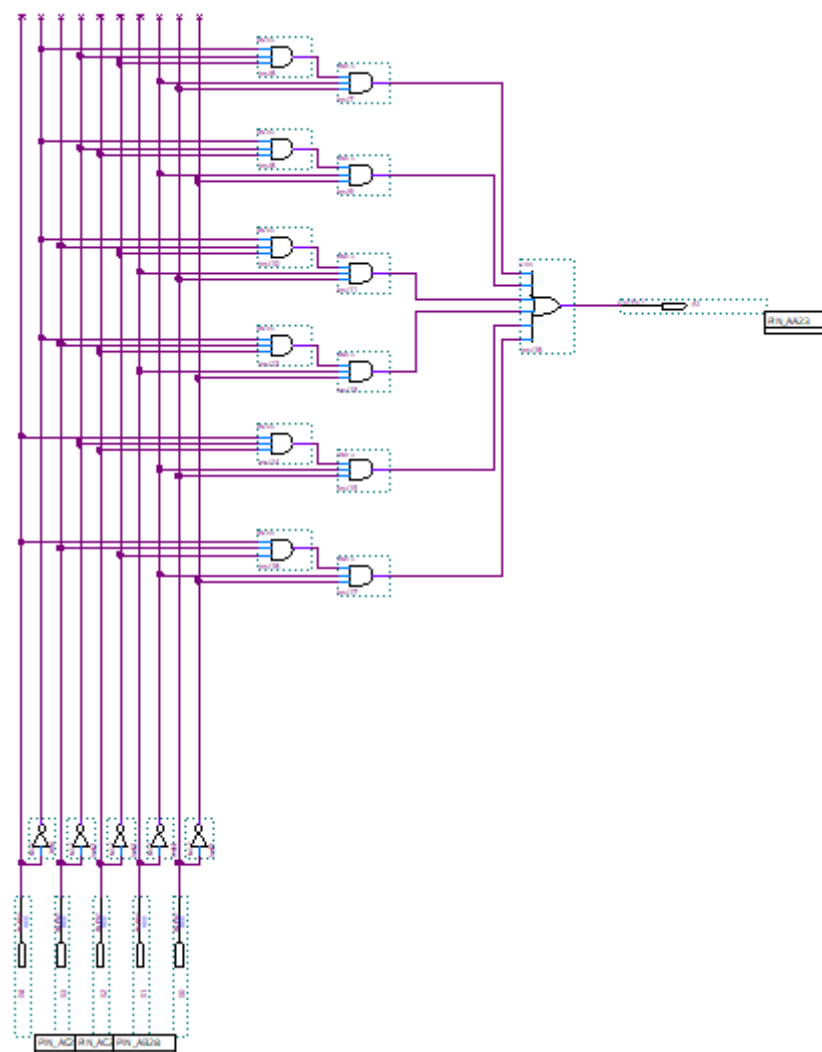


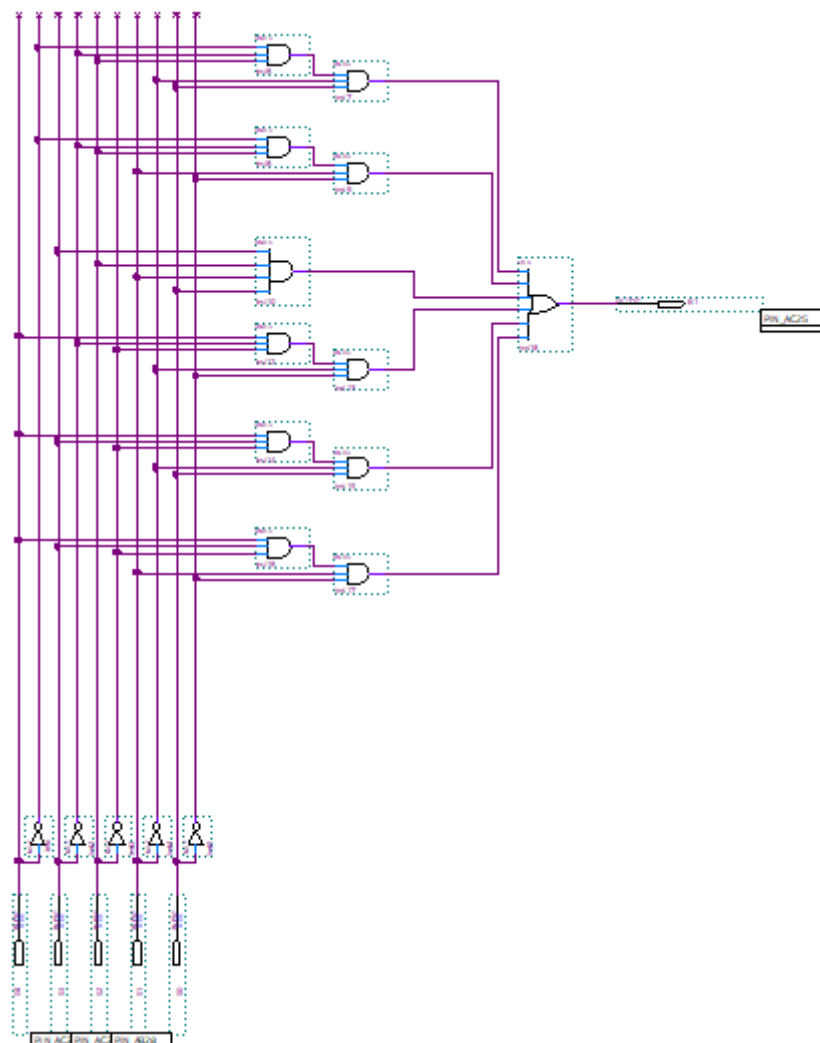


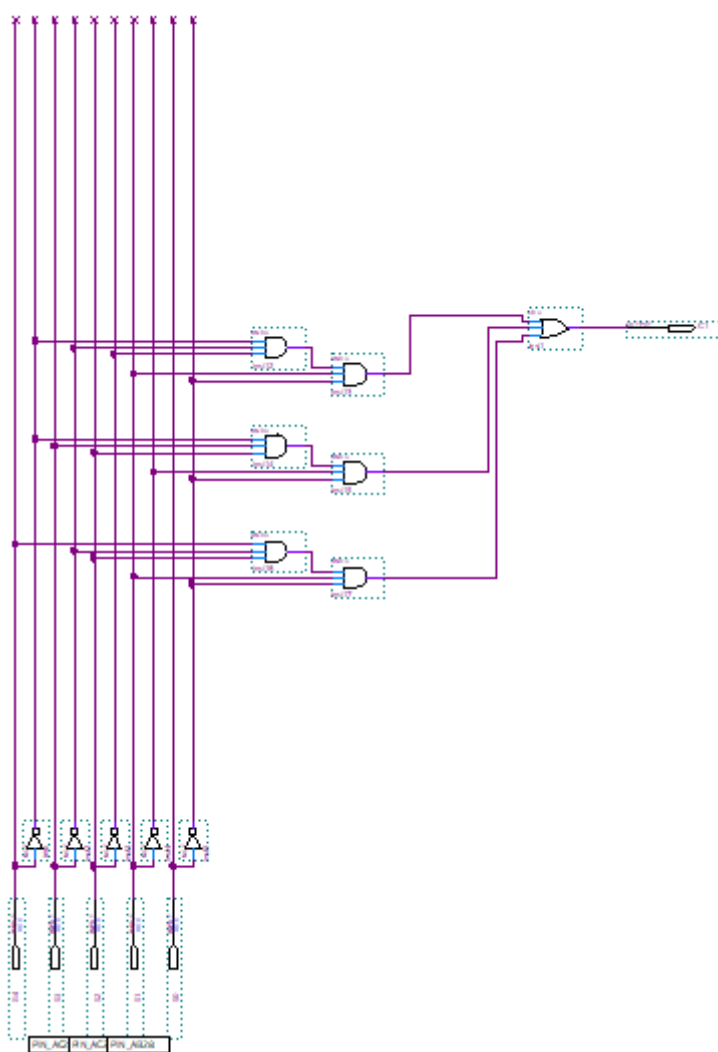


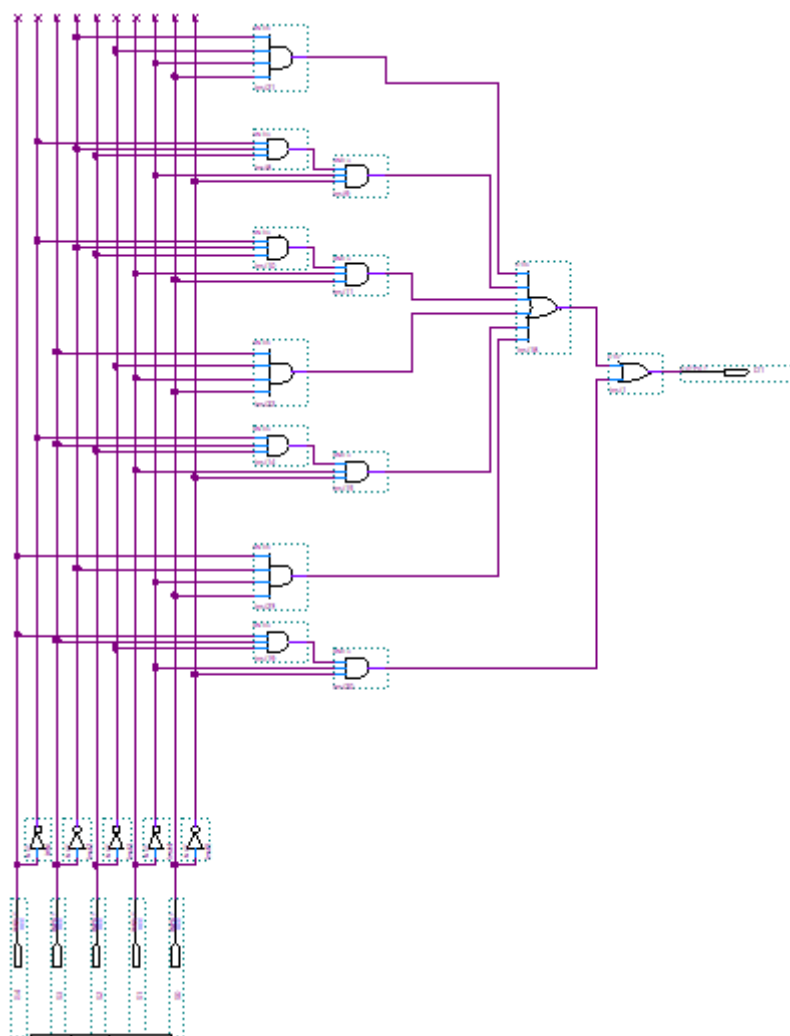
Segmentos A1-G1

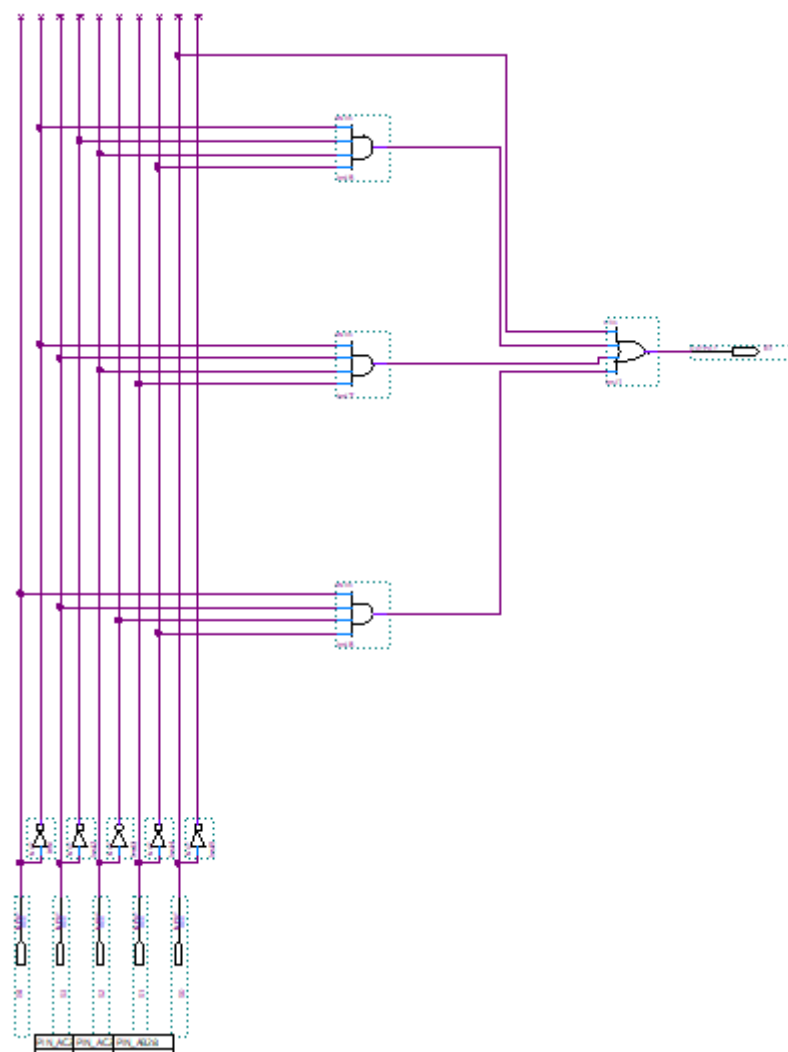


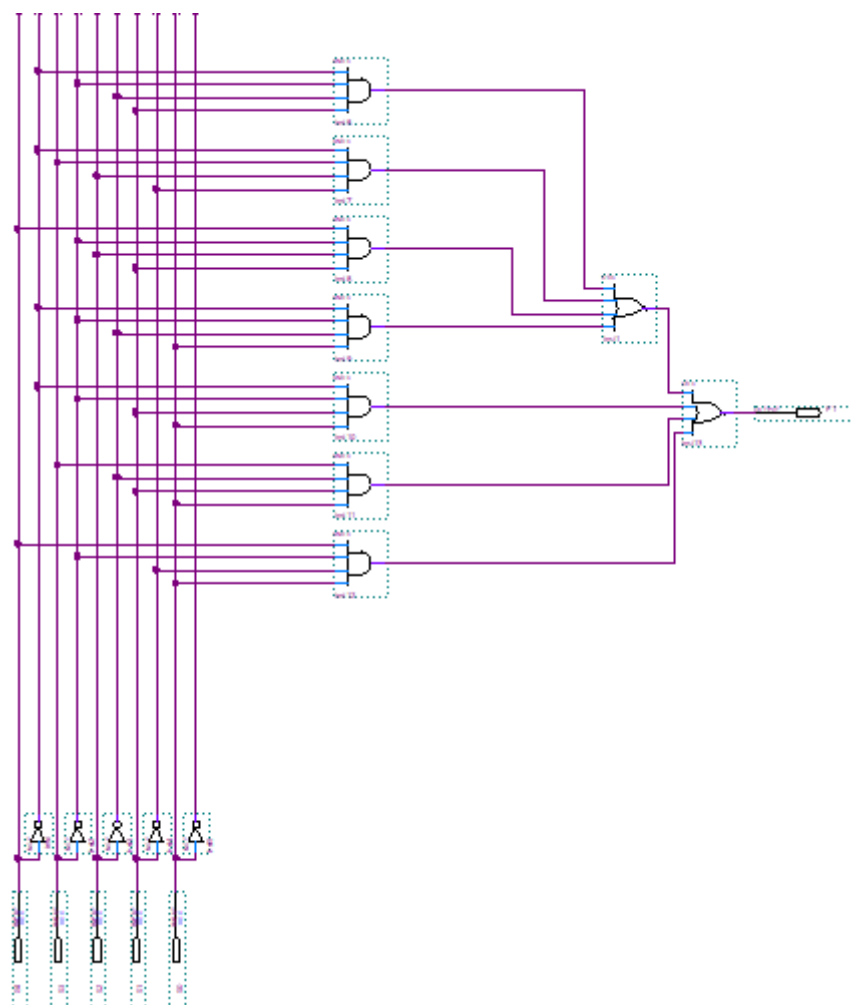


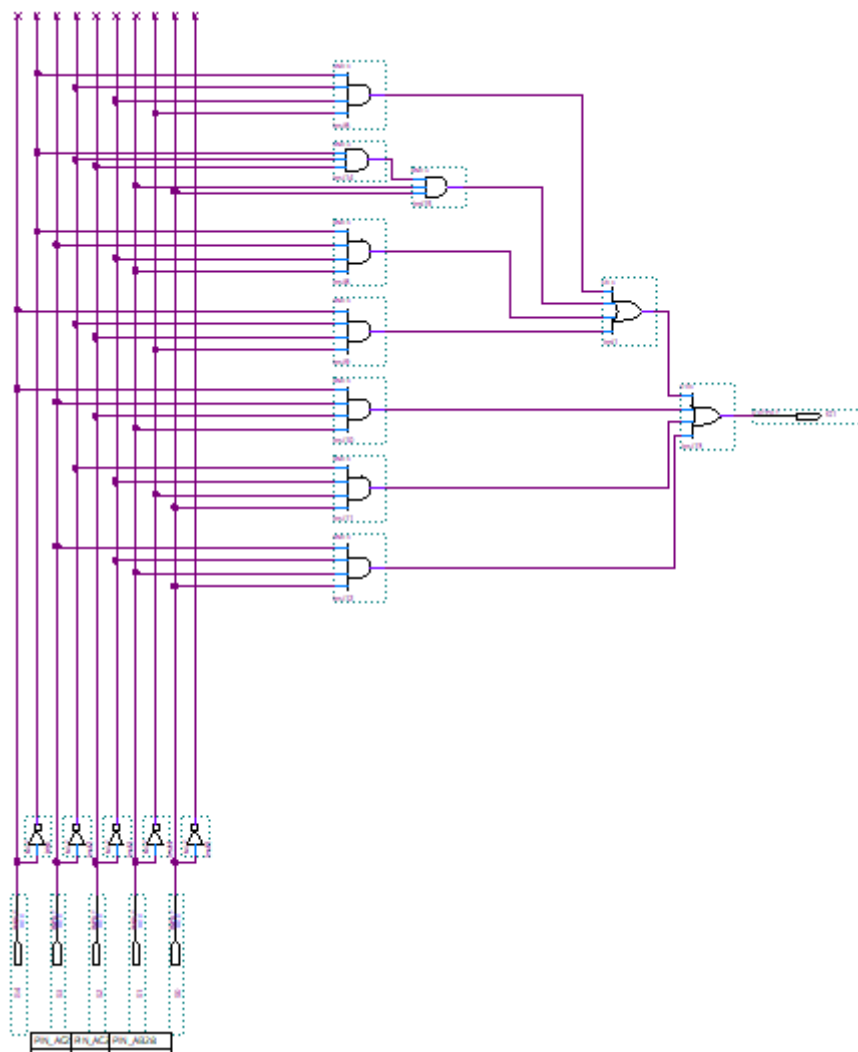




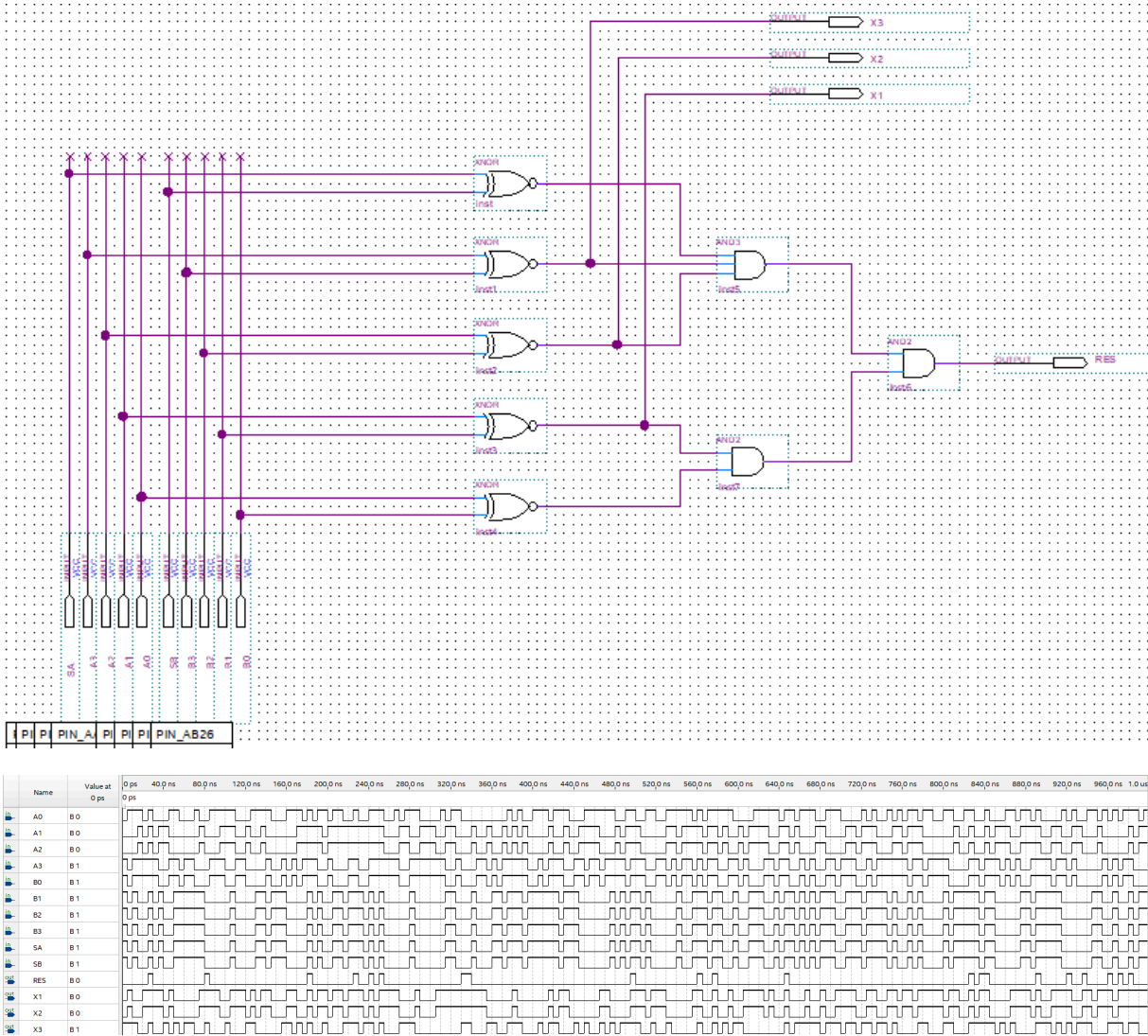






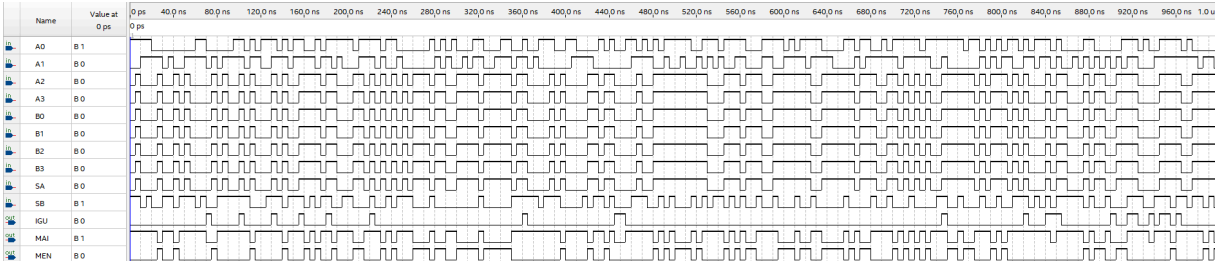
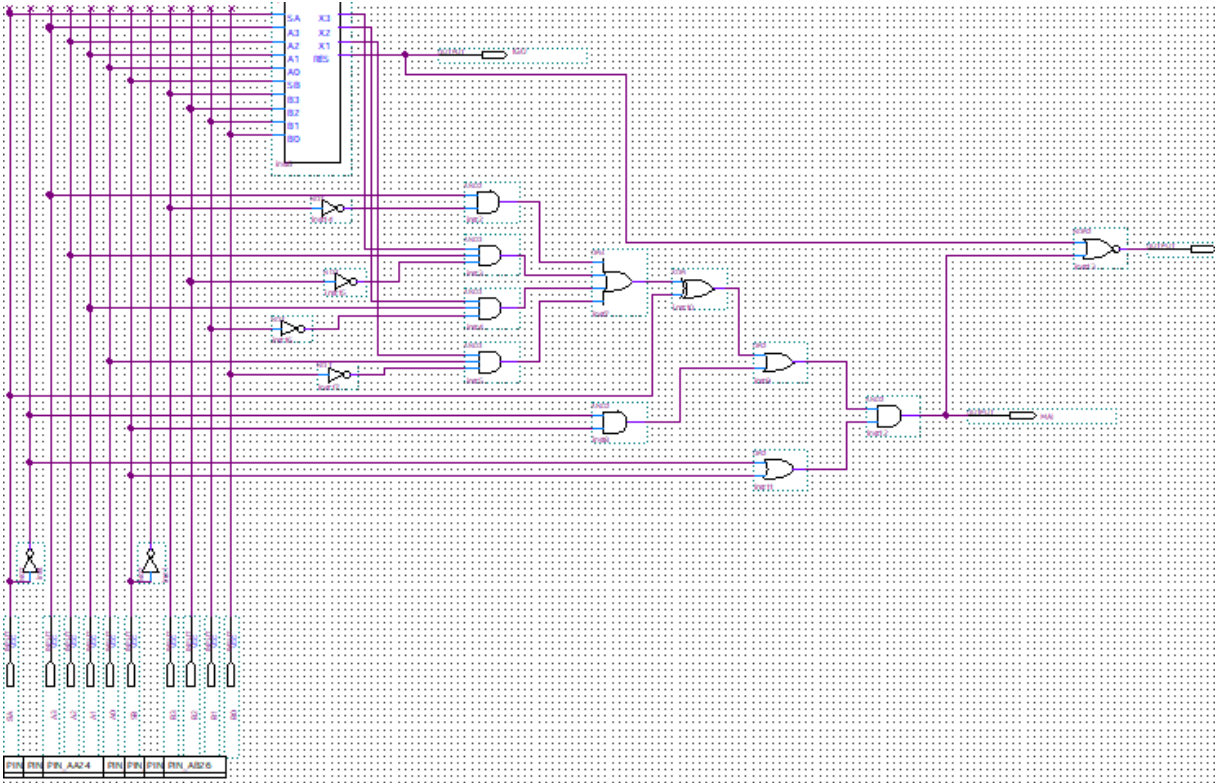


Igualdade

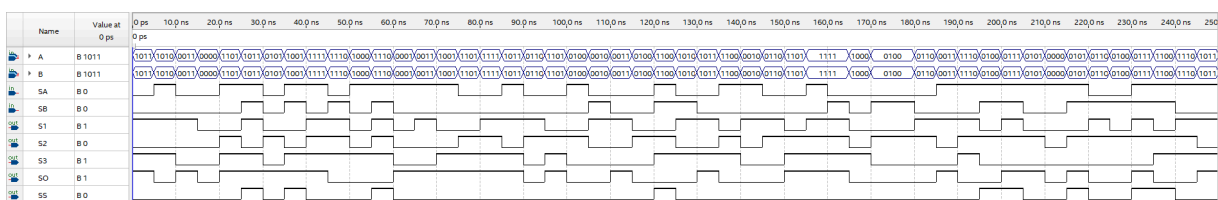
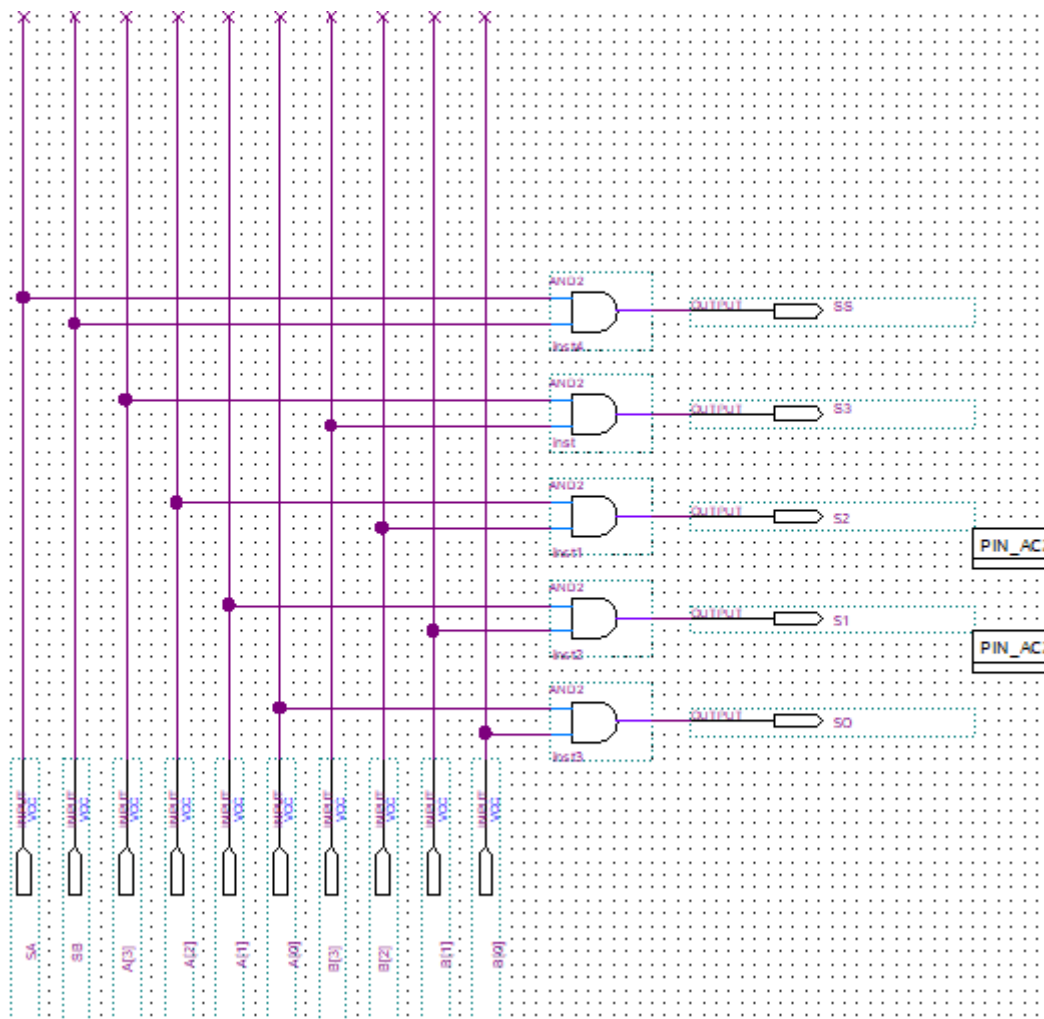




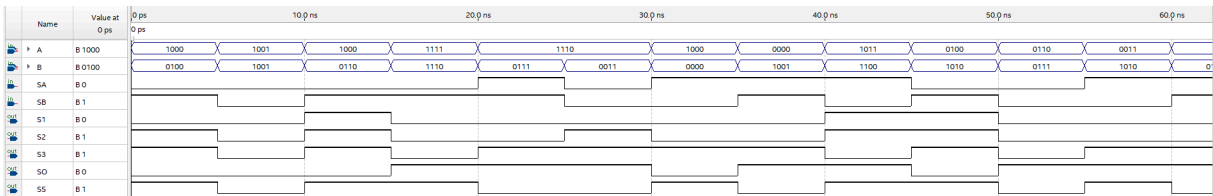
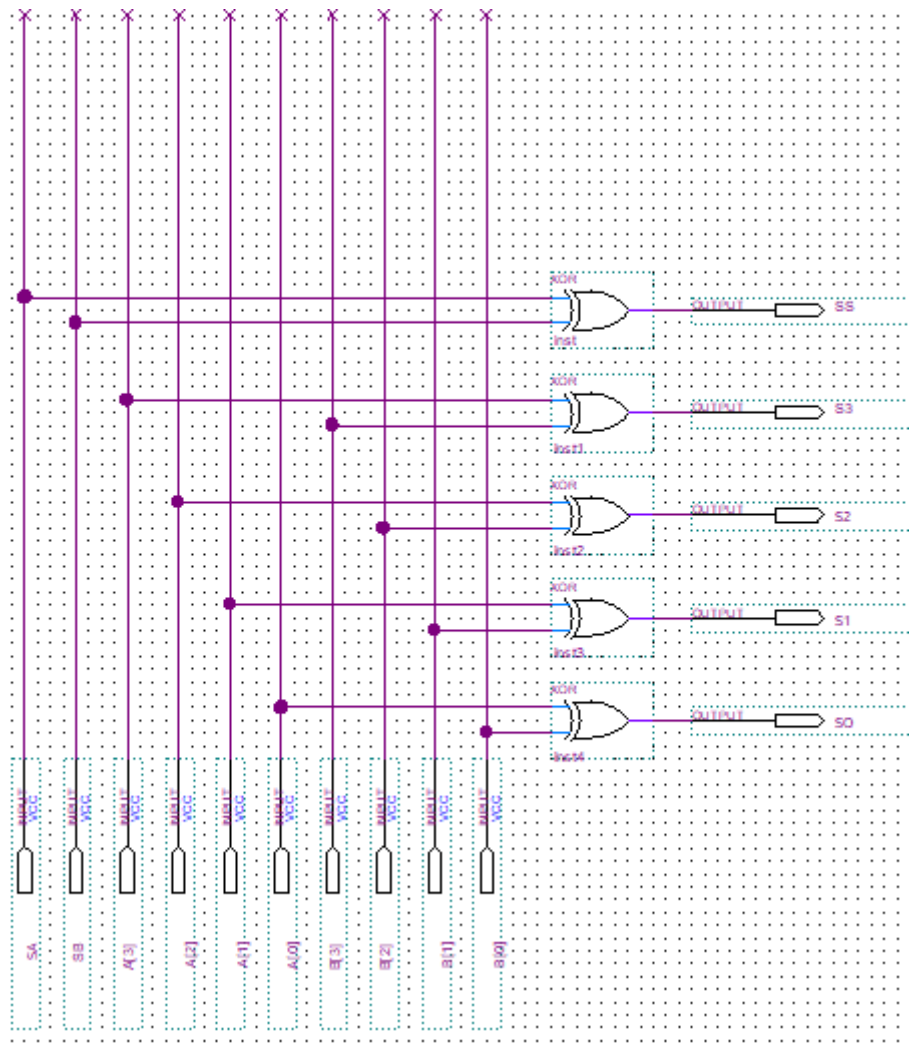
Comparador de Magnitude



## Operação AND

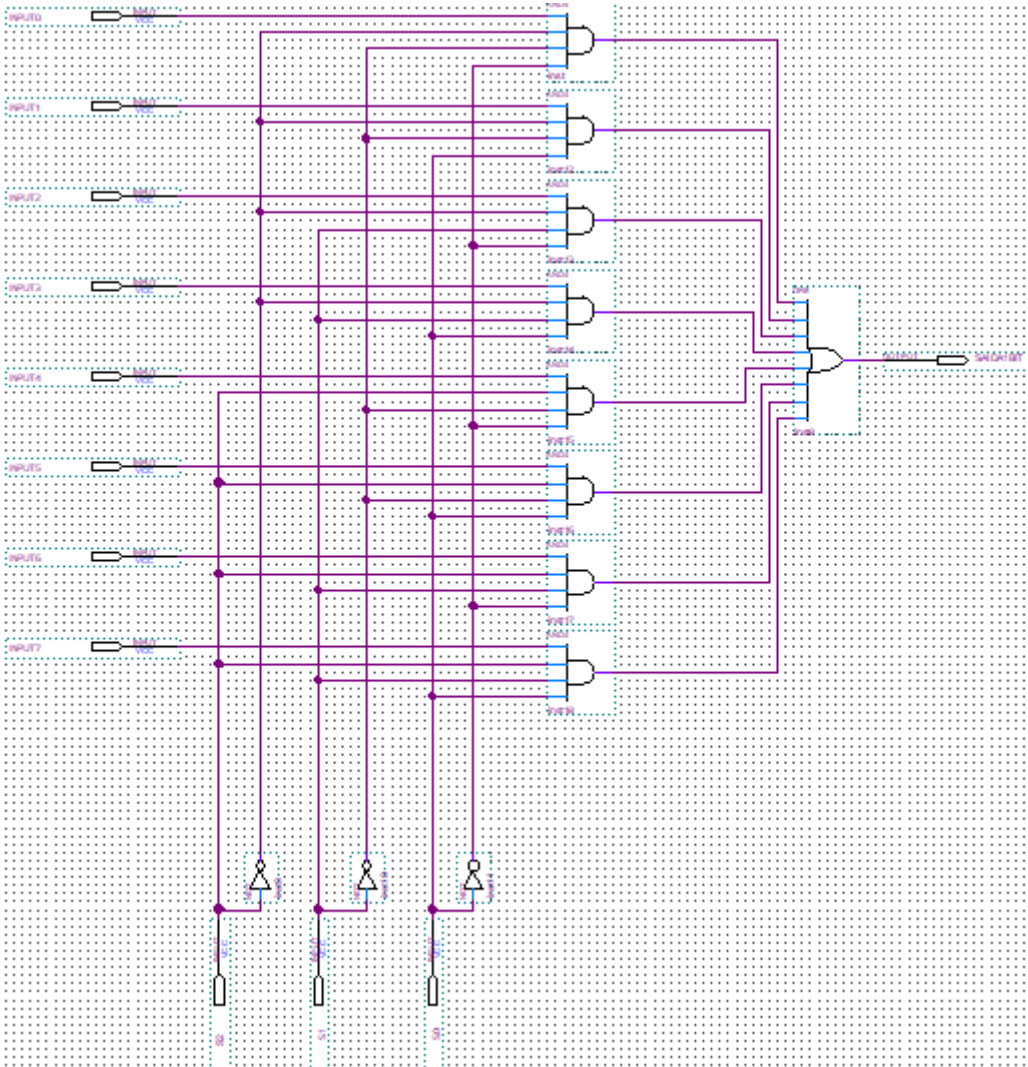


Operação XOR

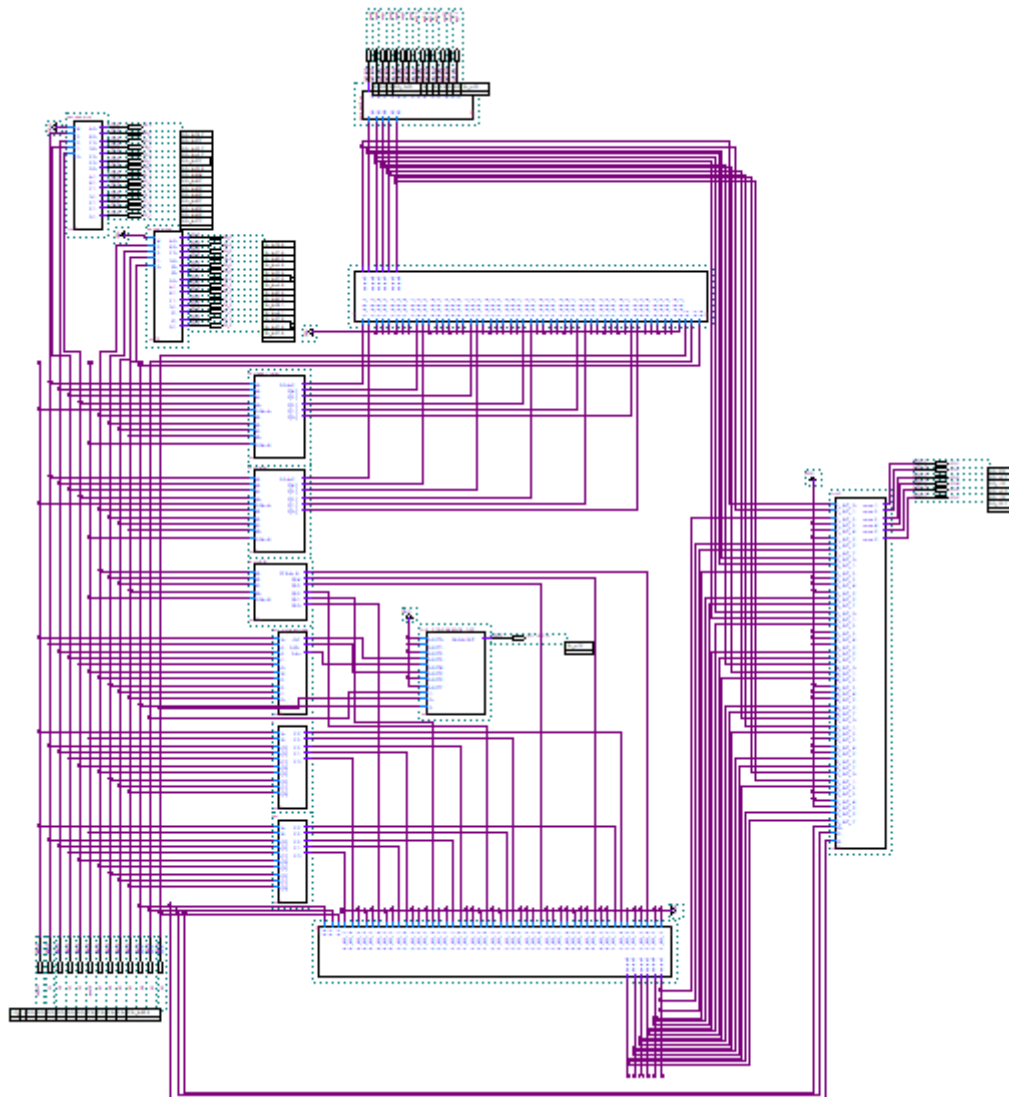


MUX 6x1

MUX Comparador



#### 4 CIRCUITO COM TODO SISTEMA CONECTADO E SIMULAÇÃO





## 5 CONCLUSÃO

Nossa ULA foi bem desenvolvida conforme a solicitação do professor, nós tivemos alguns erros e dificuldades durante o processo, mas conseguimos concluir com êxito! Nossa ULA, realiza todas operações solicitadas, são elas: Soma, Subtração, Complemento a 2 de B, Igualdade, Maior que, Menor que, AND, XOR. As operações de soma e subtração, são visíveis no display de 7 segmentos e o restante das operações são visíveis através dos LEDs.