



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
SISTEMAS DIGITAIS - PROJETO DA ULA

ESTUDANTE: JEFFERSON PEREIRA DE OLIVEIRA JÚNIOR (jpoj)

ESTUDANTE: JOÃO PEDRO LIMA DE ARAÚJO (jpla)

ESTUDANTE: LUIZ FELIPE PINTO ÁVILA DE BARROS (lfpab)

ESTUDANTE: THÉO MARCOS DO EGITO MOURA (tmem)

CURSO: ENGENHARIA DA COMPUTAÇÃO

DISCIPLINA CURSADA: SISTEMAS DIGITAIS

RELATÓRIO DO PROJETO DA ULA

Recife - PE

2025

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Relatório apresentado como parte do projeto da primeira unidade de Sistemas Digitais, sendo requisito obrigatório para fins avaliativos da referida disciplina.

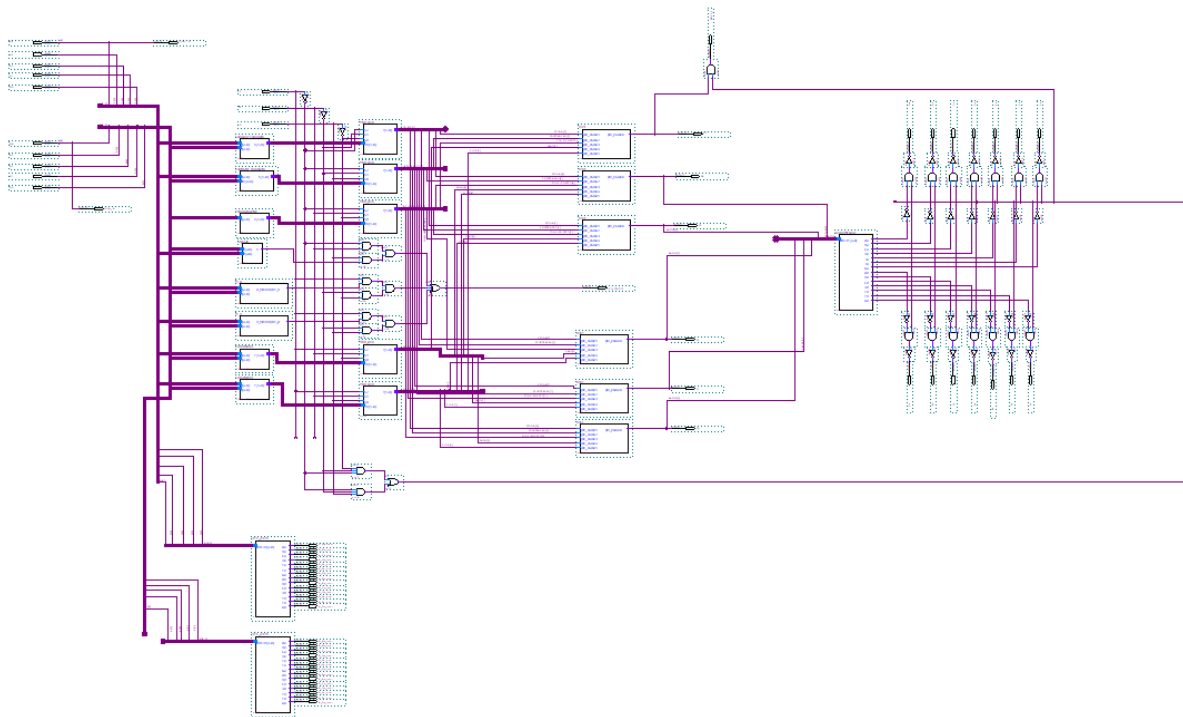
SUMÁRIO

1. APRESENTAÇÃO	3
2. VISÃO GERAL DO PROJETO	4
3. CIRCUITO DE $F = A + B$	8
4. CIRCUITO DE $F = A - B$	10
5. CIRCUITO DE $F = \text{COMPLEMENTO A 2 DE B}$	11
6. CIRCUITO DE $F = A = B$	12
7. CIRCUITO DE $F = A > B$ (CONSIDERANDO O SINAL)	13
8. CIRCUITO DE $F = A < B$ (CONSIDERANDO O SINAL)	15
9. CIRCUITO DE $F = A \text{ AND } B$	19
10. CIRCUITO DE $F = A \text{ XOR } B$	20
11. DECODIFICADOR DO DISPLAY:	21
12. CONCLUSÃO	23

1. APRESENTAÇÃO

Nas páginas seguintes, será relatado o processo de criação do Projeto da ULA, destrinchando seus módulos e visão geral. A atividade utilizou a plataforma Quartus e é resultado dos aprendizados da primeira unidade de Sistemas Digitais utilizando conceitos como tabelas-verdades, mapas de Karnaugh, portas lógicas e multiplexadores.

2. VISÃO GERAL DO PROJETO



Circuito geral do projeto

MÓDULO DA OPERAÇÃO SOMADOR:

No circuito somador, as operações estão divididas em 3 casos:

- Caso em que A e B possuem o mesmo sinal;
- Caso em que A e B possuem sinais diferentes e o módulo de A é maior que o módulo de B;
- Caso em que A e B possuem sinais diferentes e o módulo de B é maior que o módulo de A

No caso em que A e B possuem o mesmo sinal, utilizaremos um cascadeamento de somadores de 1 bit, para os valores de B e de A, que nos resultará em um número de 5 bits, e o sinal será determinado pelo sinal em questão. Por exemplo: se A é -3 e B é -9, somaremos normalmente os 2 números e, após isso, determinaremos o sinal pegando qualquer um dos dois sinais, visto que uma soma de dois números negativos resultará em um número negativo, e o mesmo serve para a soma de dois números positivos.

Para o caso em que A e B possuem sinais diferentes, encontraremos o complemento a 2 do menor valor em módulo, por exemplo: $A = +4$; $B = -6$, apesar de +4 ser maior que -6, o valor

de A é o menor em módulo e portanto o sinal será determinado pelo valor de B que é o menor em módulo. A partir disso, pegaremos o complemento a 2 do menor número e, então somaremos o valor do maior número ao complemento, resultando em um número de 4 bits. O quinto bit sempre será 0, visto que não tem como você somar dois números de 4 bits de sinais diferentes e resultar em um número de 5 bits, e o sexto bit será determinado pelo comparador de magnitude. O mesmo serve para o caso em que A é maior que B.

No caso em que os dois números tem magnitude igual e sinais diferentes, não entrará em nenhum dos casos e a saída será 0.

MÓDULO DA OPERAÇÃO SUBTRATOR:

Tendo o somador em mãos, basta negar o sinal de B, que as operações feitas pra a soma serão feitas com o sinal negado, e conseqüentemente representarão uma subtração.

MÓDULO DA OPERAÇÃO COMPLEMENTO A 2 DE B:

Para realizar a operação do complemento a 2, barraremos todas os bits de B e somaremos 1 ao resultado do barramento. Para realizar a soma, concatenaremos uma espécie de somador, que na prática não é um somador, visto que não teremos uma entrada B, será apenas um somador com +1. Para realizá-lo, utilizamos um VCC ligado à primeira entrada e armazena o carry para as próximas entrada e assim por diante.

MÓDULO DA OPERAÇÃO A = B:

Para realizar a operação de $A=B$, basta concatenar 5 XNOR, que resultarão em 1 quando os bits forem igual. Juntando as saídas dessas 5 XNOR num AND5, caso todas as entradas sejam iguais, a saída do AND5 será 1, caso haja algum dos 5 bits diferentes, a saída será 0. A porta XNOR irá resultar em 1 sempre que $A=B$, portanto a concatenação dessas 5 portas se deve a isso.

MÓDULO DA OPERAÇÃO A > B:

Para as operações de maior que, foi pensado em fazer um módulo para verificar se a magnitude (sem considerar o sinal) de A é maior que B - M_A, e outro módulo para verificar se a magnitude de B é maior que A - M_B.

A partir desses 2 módulos, podemos separar os casos em que os dois números A e B têm sinais contrários, sendo assim fácil verificar se um é maior que o outro, e o caso em que eles

têm mesmo sinal, se ele for negativo, o maior número é o de menor magnitude, se o sinal for positivo, o maior número será o de maior magnitude.

MÓDULO DA OPERAÇÃO $B > A$:

Usando uma lógica idêntica à apresentada no módulo anterior, apenas invertendo as entradas de A para as entradas de B, podemos obter um booleano para $B > A$.

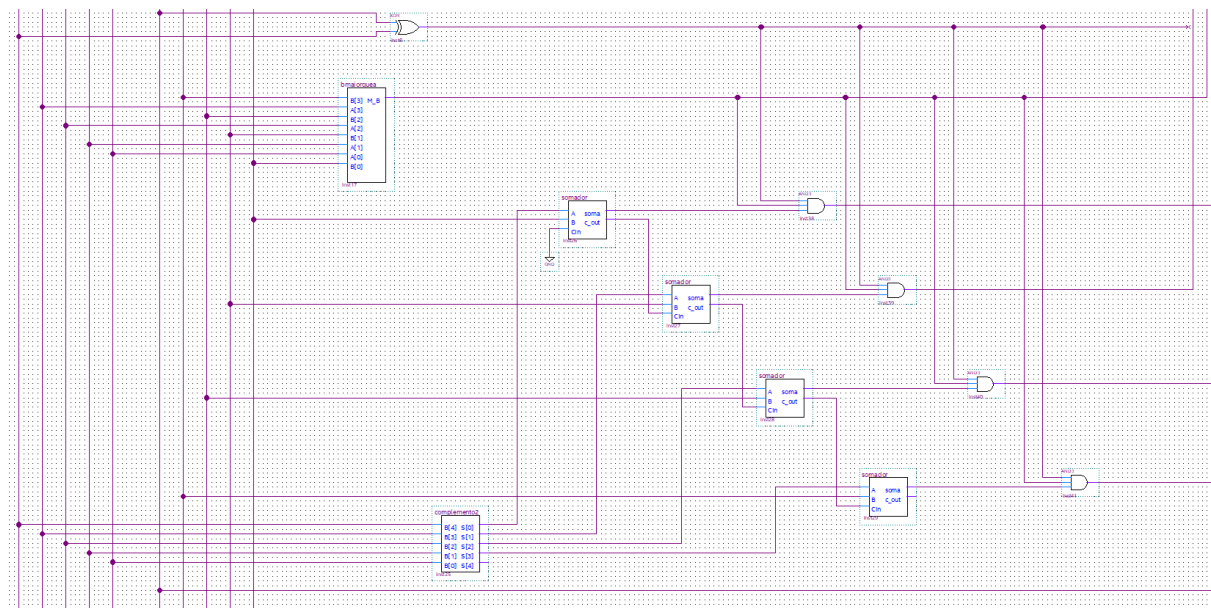
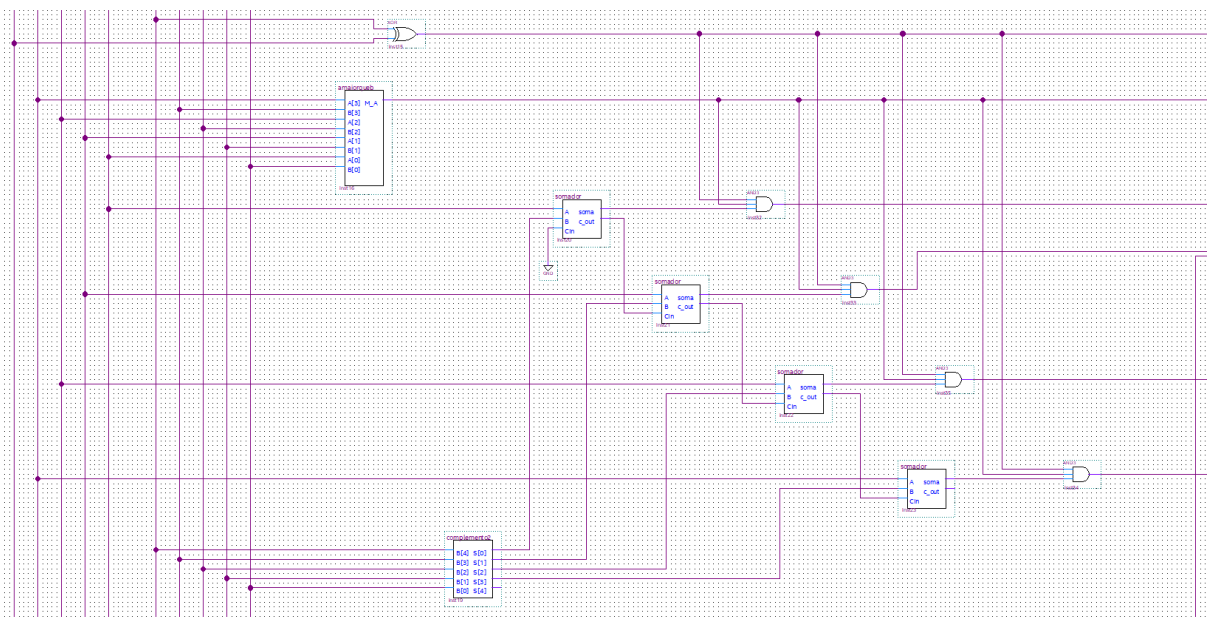
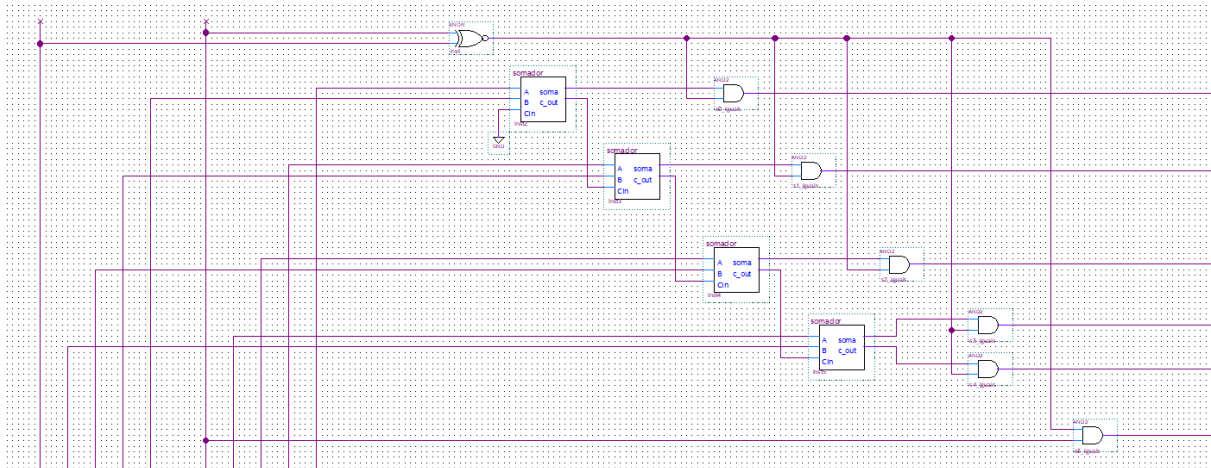
MÓDULO DA OPERAÇÃO AND:

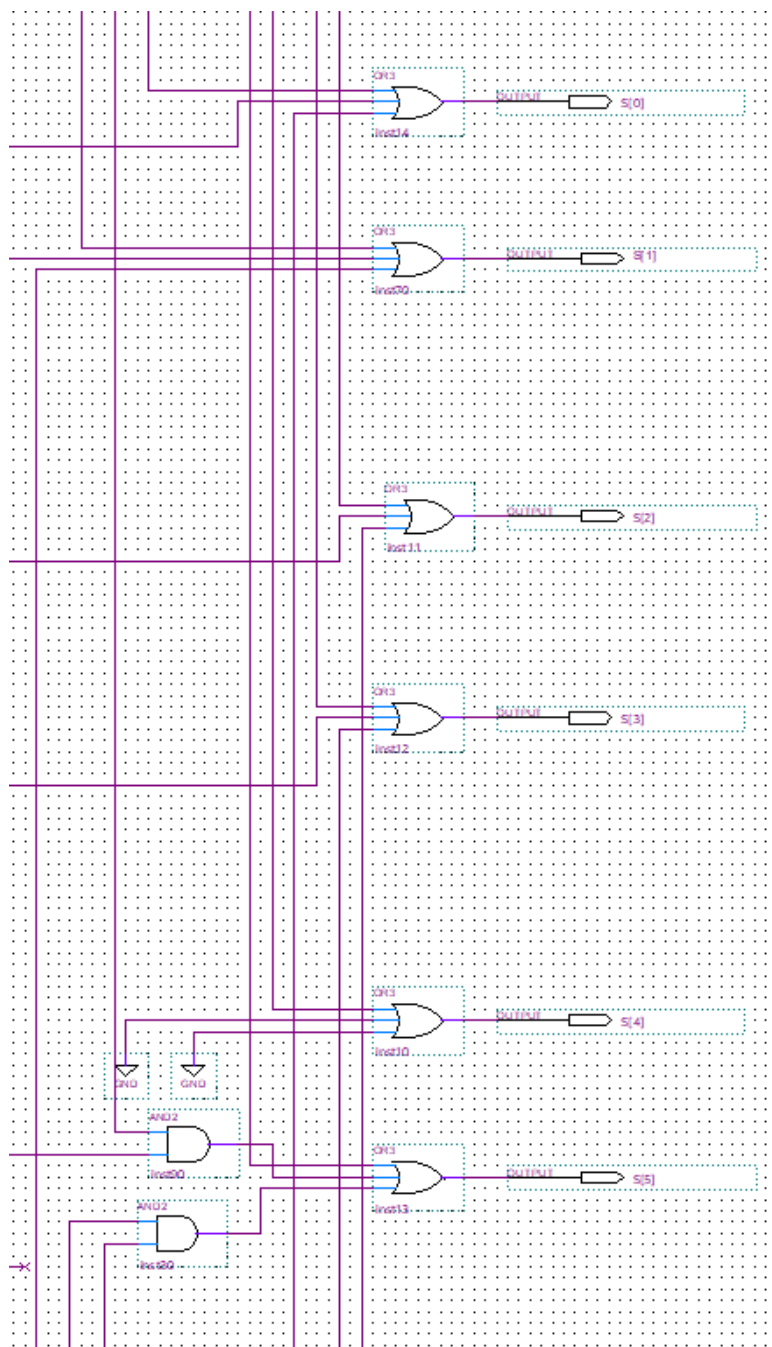
Nesse caso, é feita a operação AND bit a bit, ou seja, o bit 0 do número A realiza a operação com o bit 0 do número B e assim sucessivamente. Além disso, também é feito esse processo no sinal e, como a saída deve ser em 6 bits, o mais à esquerda fica com o sinal e, à sua direita, é colocado um bit sempre em 0.

MÓDULO DA OPERAÇÃO XOR:

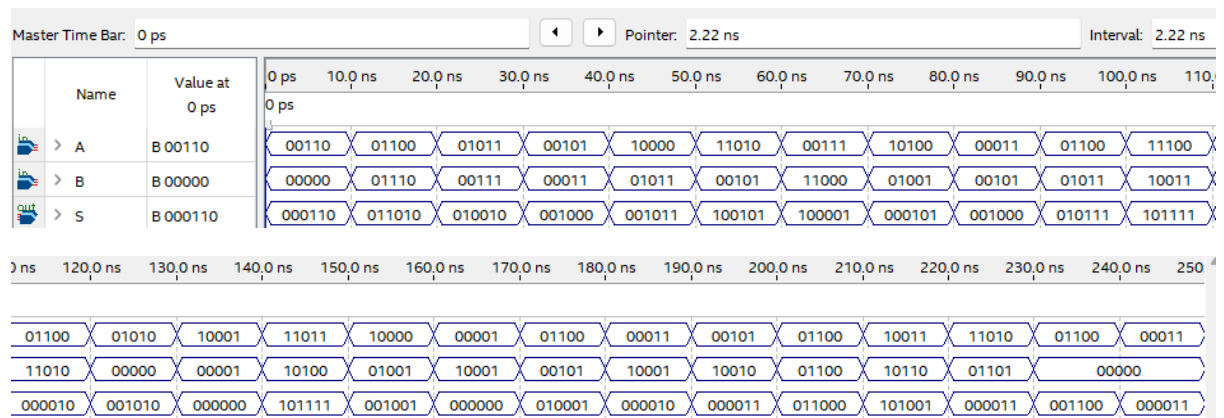
Nesse caso, é feita a operação XOR bit a bit, ou seja, o bit 0 do número A realiza a operação com o bit 0 do número B e assim sucessivamente. Além disso, também é feito esse processo no sinal e, como a saída deve ser em 6 bits, o mais à esquerda fica com o sinal e, à sua direita, é colocado um bit sempre em 0.

3. CIRCUITO DE $F = A + B$

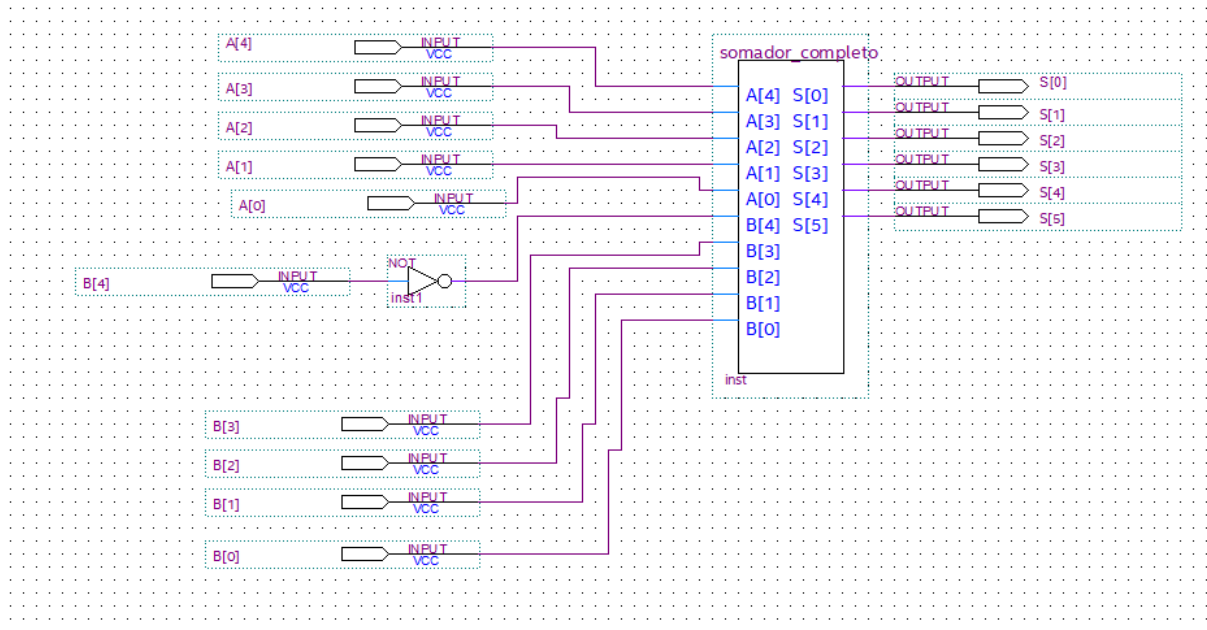





Waveform do circuito somador



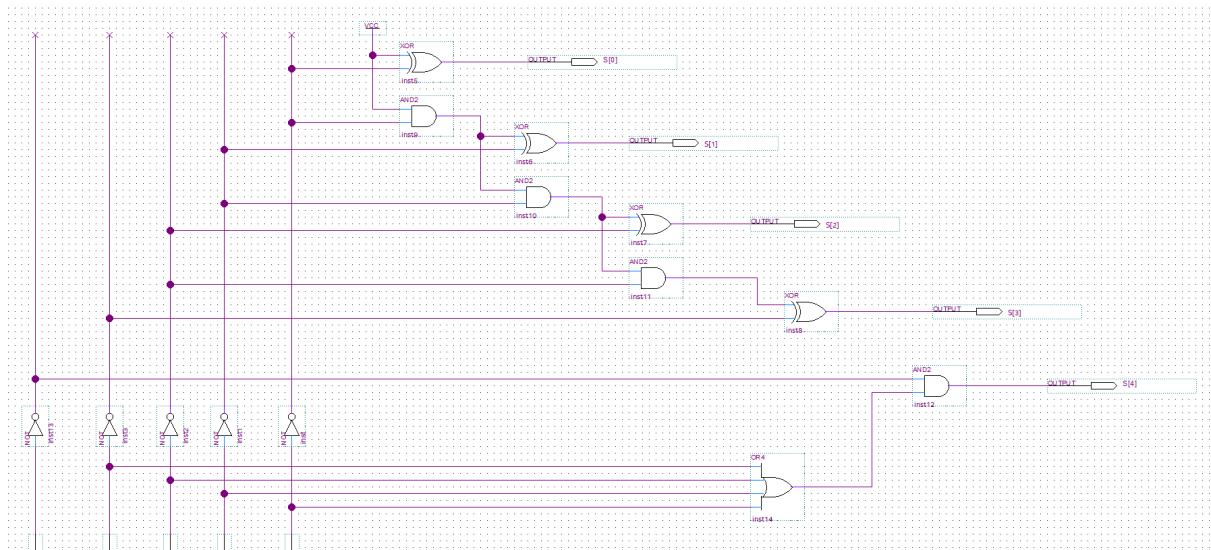
4. CIRCUITO DE $F = A - B$



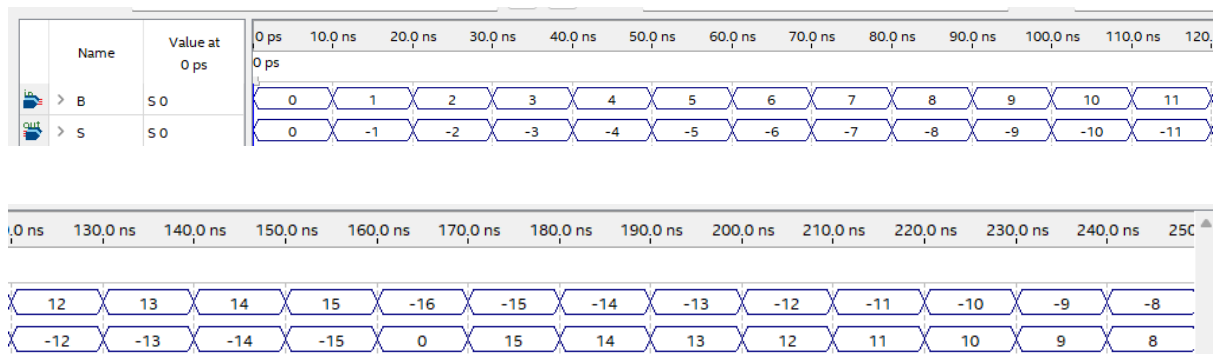
Waveform do circuito

	Name	Value at 0 ps	0 ps	10,0 ns	20,0 ns	30,0 ns	40,0 ns	50,0 ns	60,0 ns	70,0 ns	80,0 ns	90,0 ns	100,0 ns	110,0 ns
	> A	B 01100	01100	10111	00111	10001	11010	10010	11100	11011	10010	11001	01111	
	> B	B 00010	00010	01010	01001	00111	11010	11111	11000	11111	01000	11010		
	> S	B 001010	001010	110001	100010	101010	110001	001000	000011	100011	001101	110001	011001	
			10100	10001	01100	00011	01001	11100	00011	01011	11010	01000	11010	00011
			01110	10000	10110	00010	01010	10101	11001	10110	11000	10111	10010	10000
			110010	100001	001100	001001	000111	110110	001000	010100	100100	010000	100011	000101
			000000	000010	000000	000010	000000	000010	000000	000010	000000	000010	000000	000010

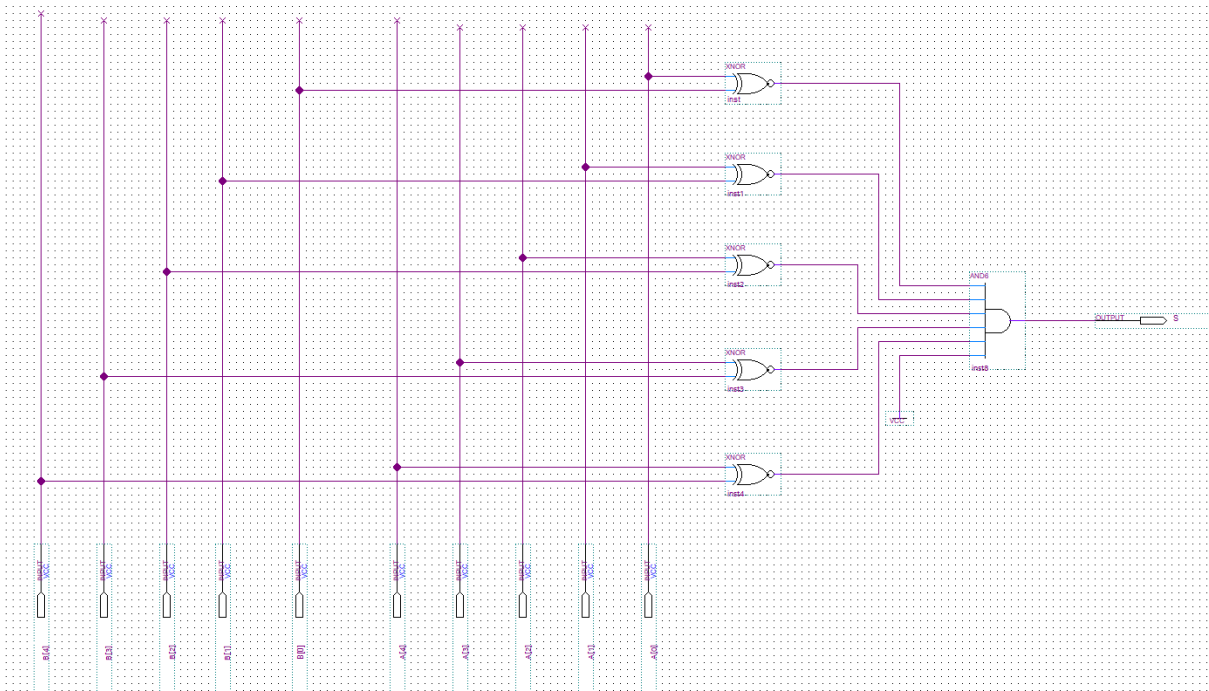
5. CIRCUITO DE $F = \text{COMPLEMENTO A 2 DE B}$



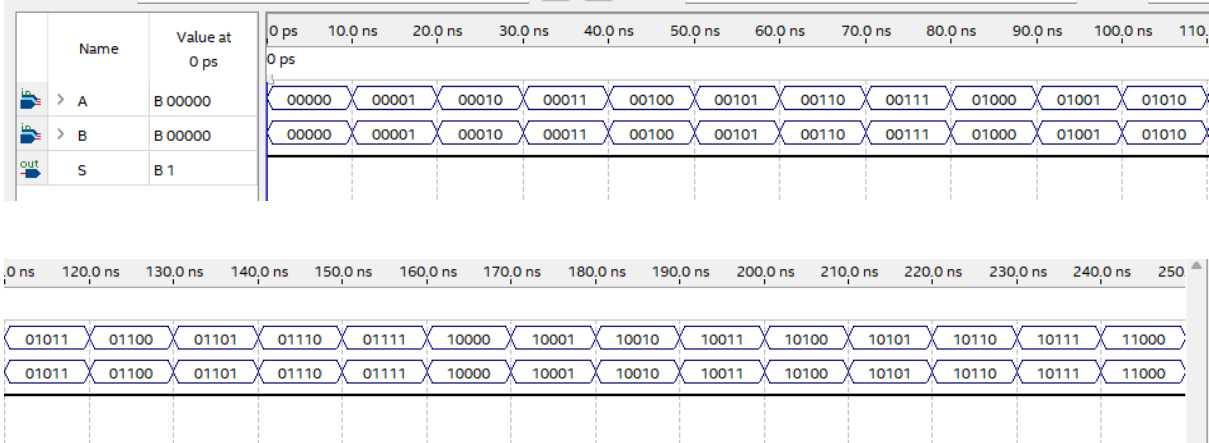
Waveform do Complemento a 2 de B:



6. CIRCUITO DE $F = A = B$



Waveform do circuito de $A = B$



7. CIRCUITO DE F = A > B (CONSIDERANDO O SINAL)

Waveform:

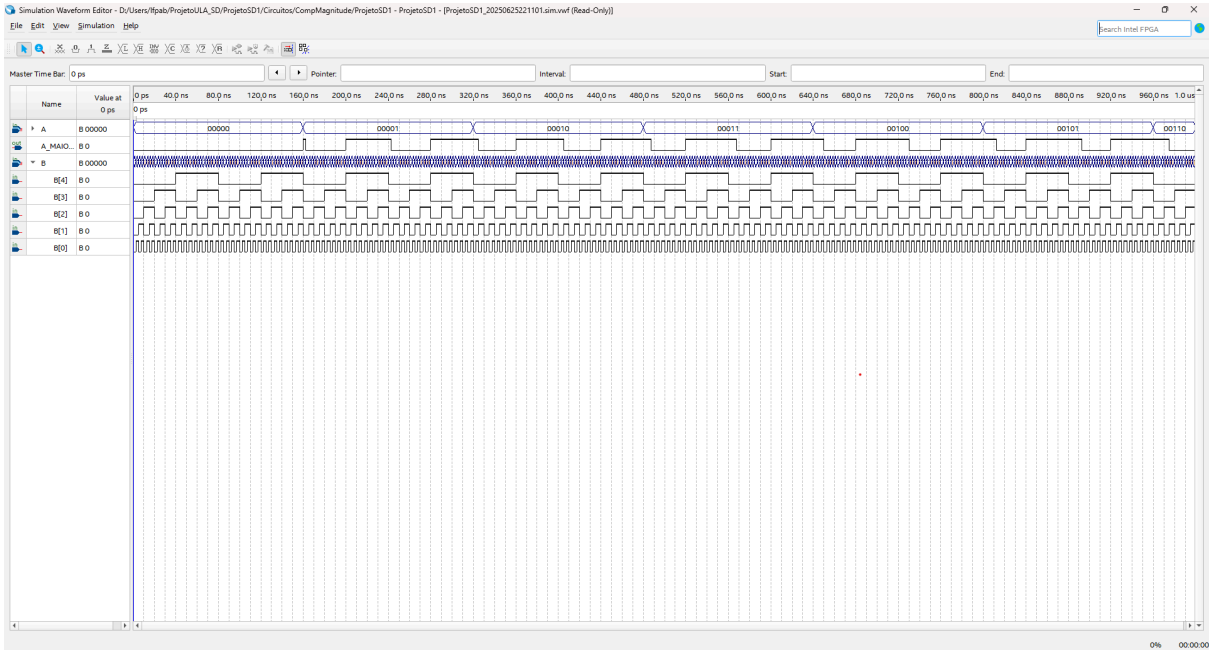


Tabela verdade:

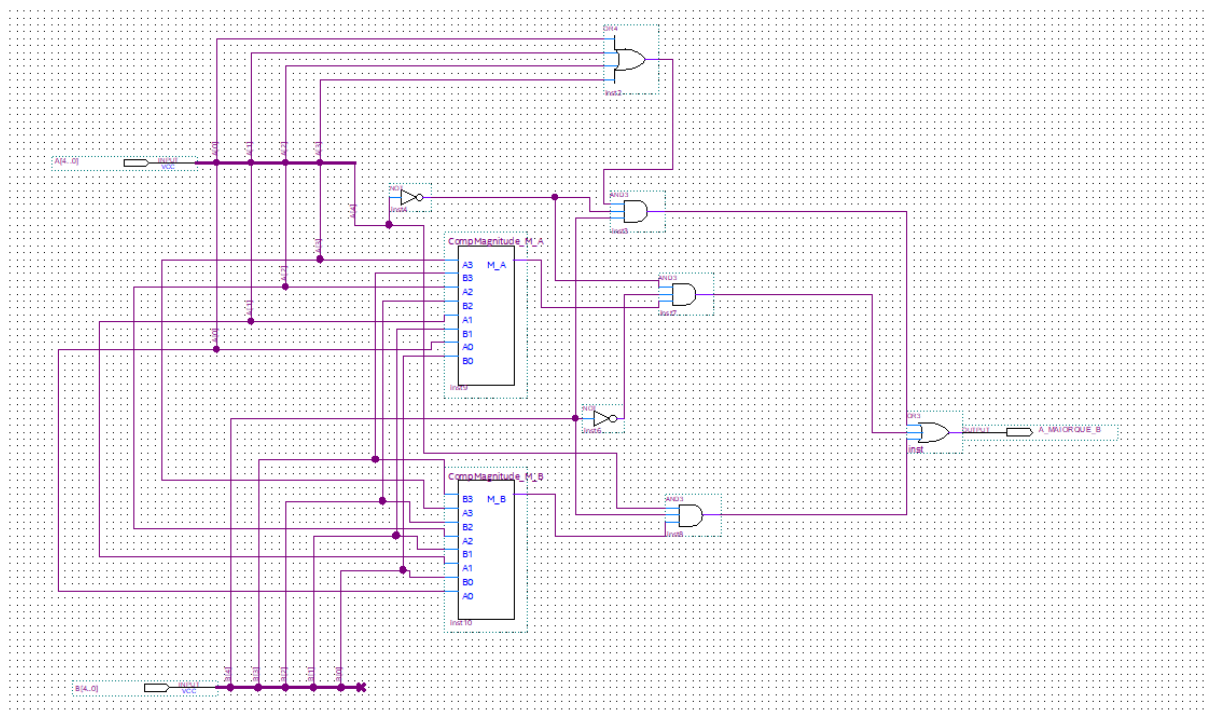
A	BIT_4	BIT_3	BIT_2	BIT_1	BIT_0	B	BIT_4	BIT_3	BIT_2	BIT_1	BIT_0	F
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	1
-2	1	0	0	1	0	2	0	0	0	1	0	0
3	0	0	0	1	1	-3	1	0	0	1	1	1
4	0	0	1	0	0	-5	1	0	1	0	1	1
-7	1	0	1	1	1	5	0	0	1	0	1	0
...	1024 combinações possíveis											

A partir do mapa-K, podemos chegar na seguinte expressão booleana:

$$F = (A_0 + A_1 + A_2 + A_3).A_4'.B_4 + A_4'.B_4'.M_A + A_4.B_4.M_B$$

Sendo M_A um comparador de magnitude que verifica se o módulo de A é maior que o módulo de B, e M_B se o módulo de B é maior que o módulo de A. Os circuitos de M_A e M_B serão explicitados mais à frente.

Circuito:



8. CIRCUITO DE $F = A < B$ (CONSIDERANDO O SINAL)

Waveform:

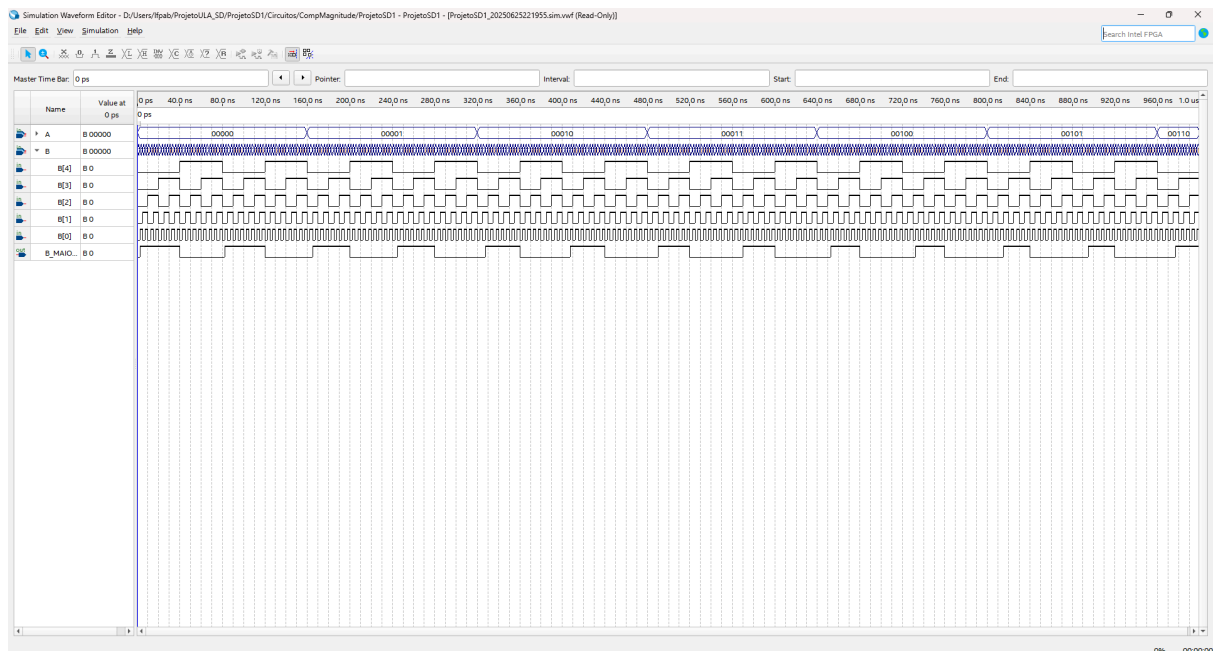


Tabela verdade:

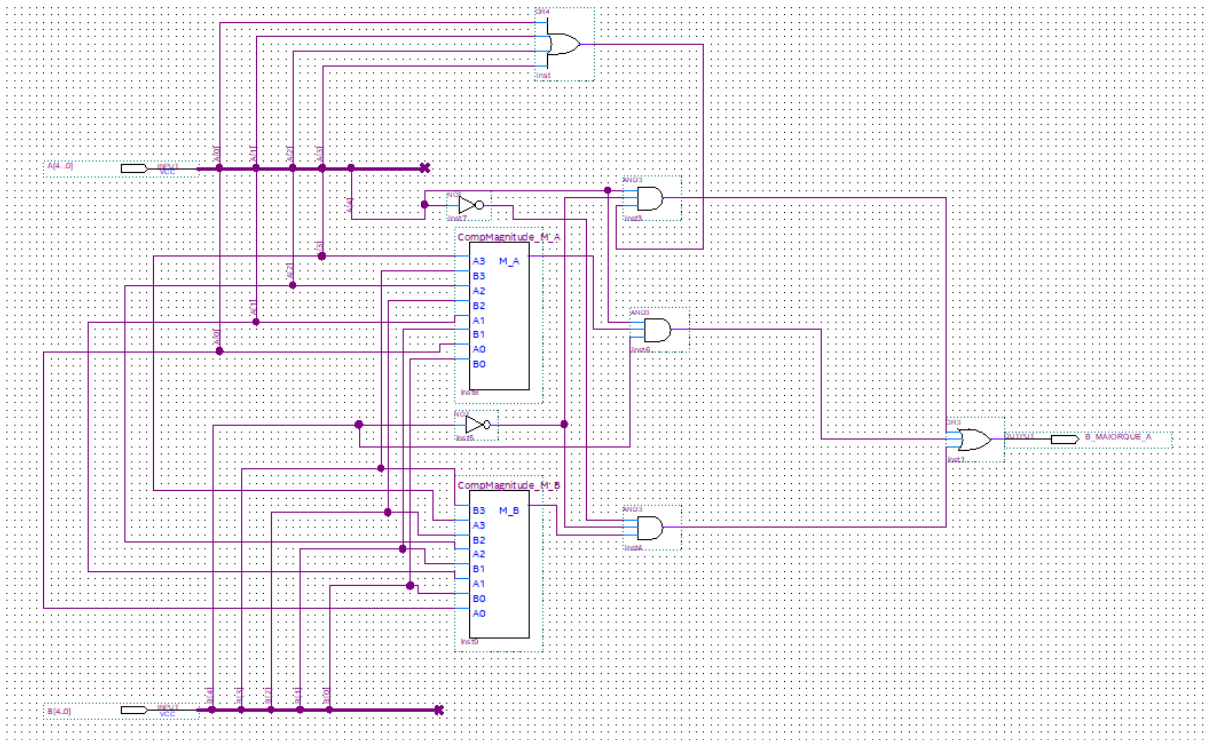
A	BIT_4	BIT_3	BIT_2	BIT_1	BIT_0	B	BIT_4	BIT_3	BIT_2	BIT_1	BIT_0	F
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0
-2	1	0	0	1	0	2	0	0	0	1	0	1
3	0	0	0	1	1	-3	1	0	0	1	1	0
4	0	0	1	0	0	-5	1	0	1	0	1	0
-7	1	0	1	1	1	5	0	0	1	0	1	1
...	1024 combinações possíveis											

A partir do mapa-K, podemos chegar na seguinte expressão booleana:

$$F = (A_0 + A_1 + A_2 + A_3) \cdot A_4 \cdot B_4' + A_4 \cdot B_4 \cdot M_A + A_4' \cdot B_4' \cdot M_B$$

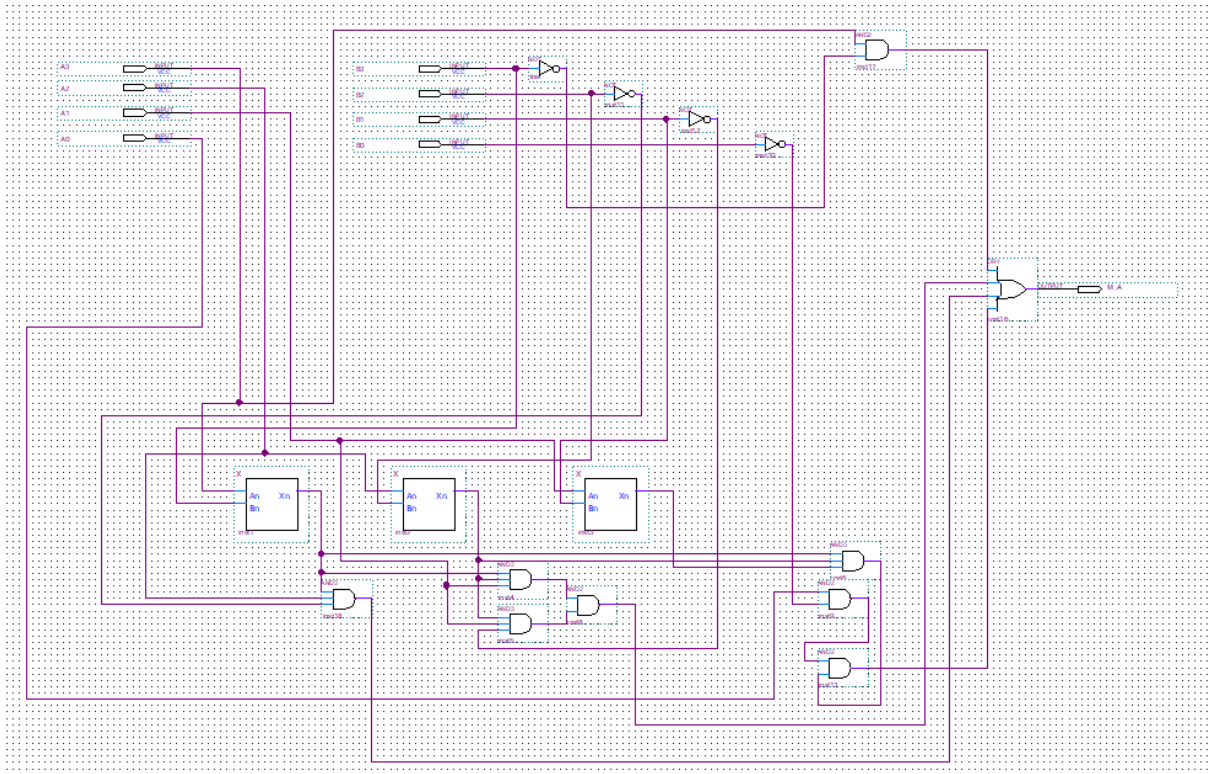
Sendo M_A um comparador de magnitude que verifica se o módulo de A é maior que o módulo de B, e M_B se o módulo de B é maior que o módulo de A. Os circuitos de M_A e M_B serão explicitados mais à frente.

Circuito:

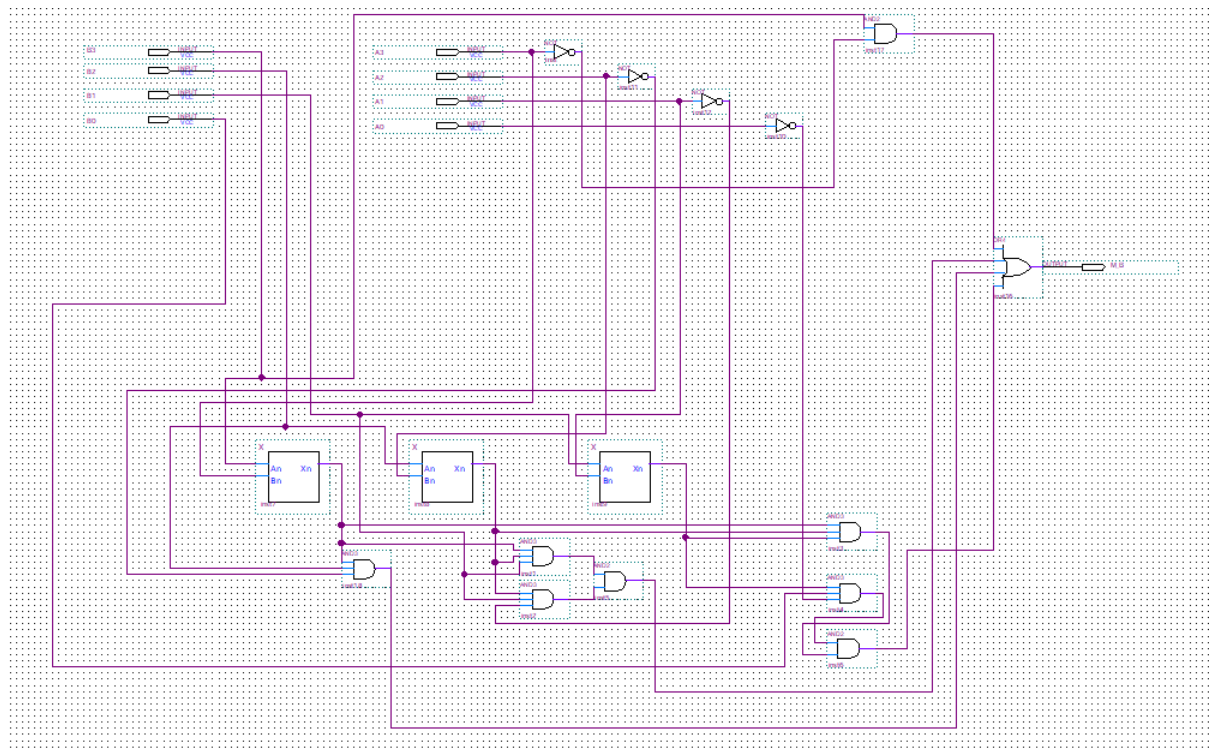


CIRCUITO DO COMPARADOR DE MAGNITUDE

CompMagnitude_M_A:



CompMagnitude_M_B:



OBS: O módulo X retorna a operação $(A \text{ XOR } B)'$.

9. CIRCUITO DE $F = A \text{ AND } B$

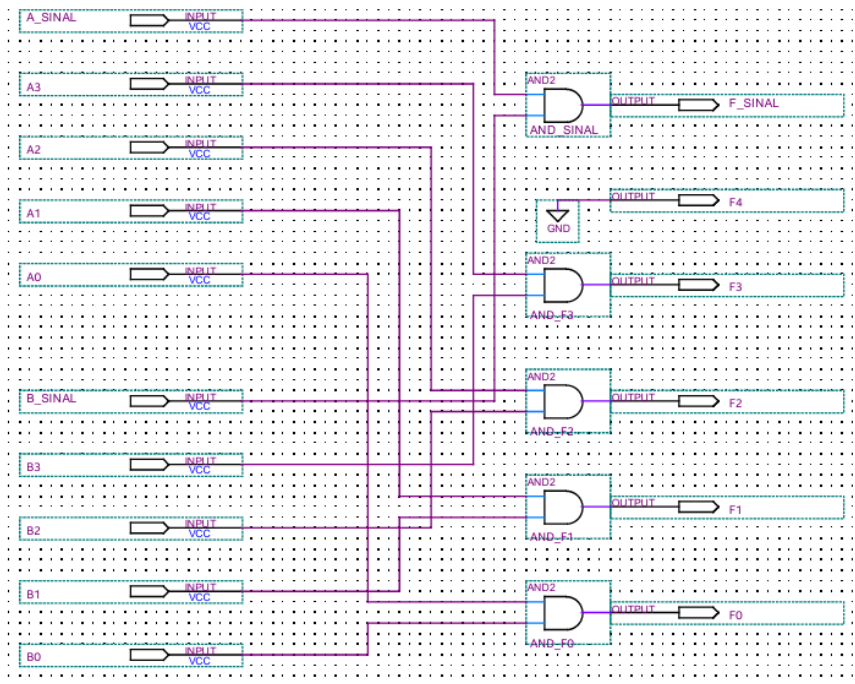


Imagem do circuito da operação “AND”.

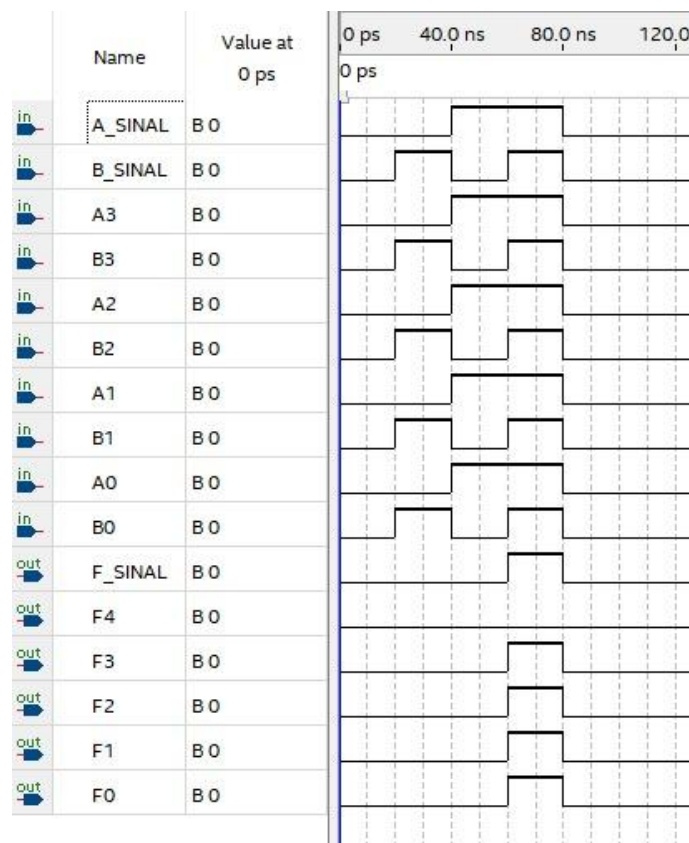


Imagem do waveform da operação “AND”.

10. CIRCUITO DE $F = A \text{ XOR } B$

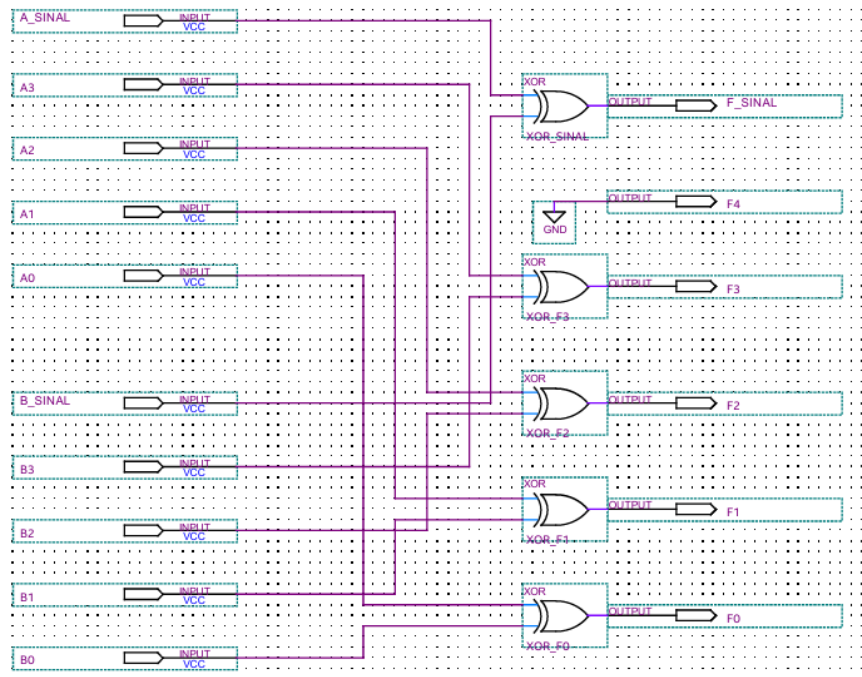


Imagem do circuito da operação “XOR”.

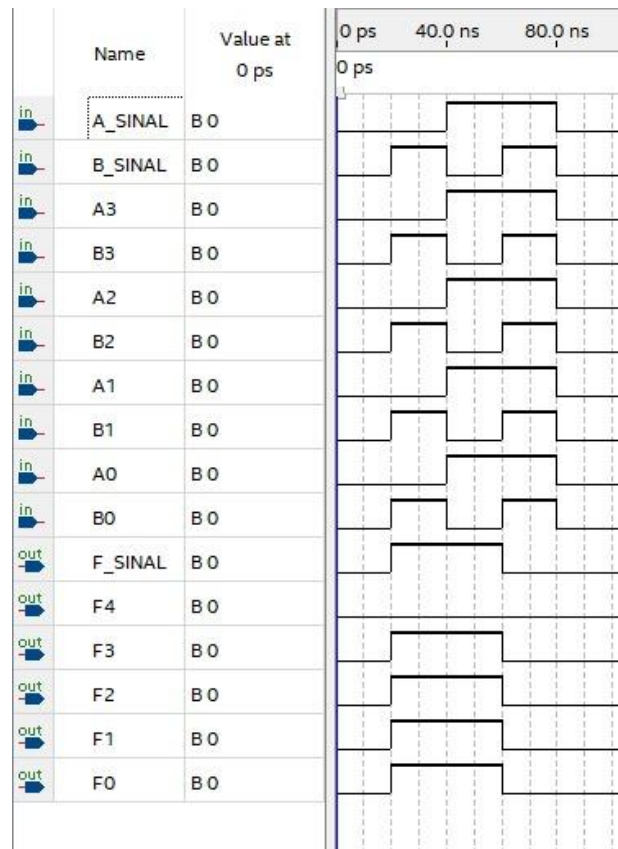


Imagem do waveform da operação “XOR”.

11. DECODIFICADOR DO DISPLAY:

Waveform:

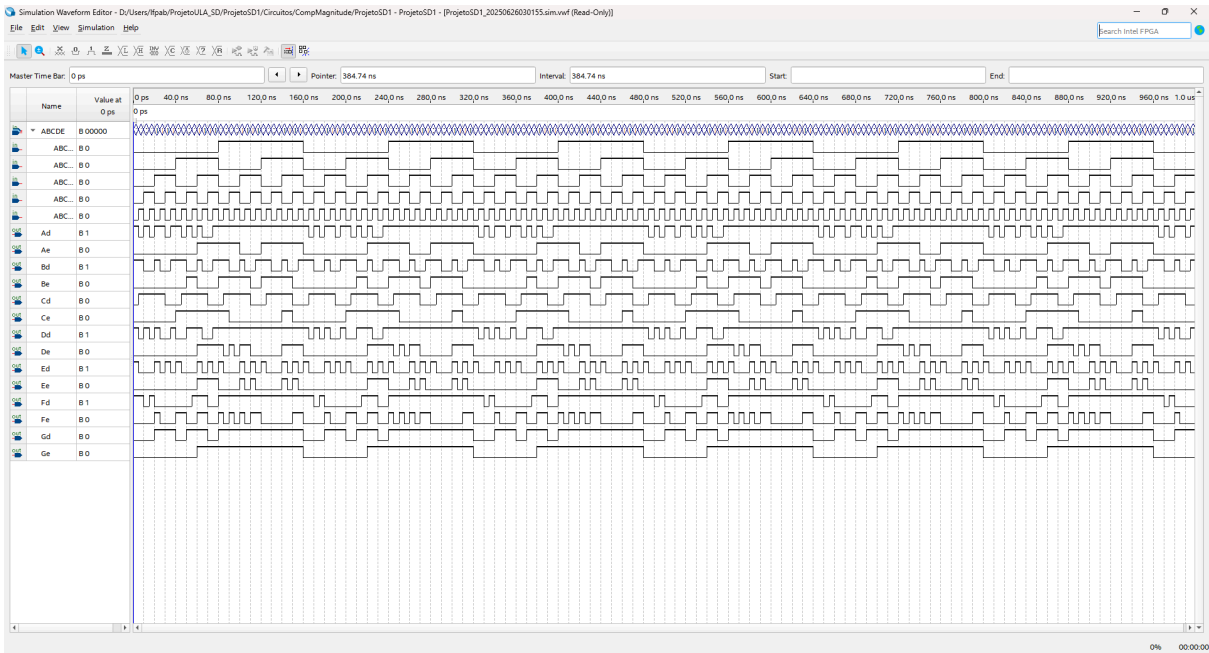


Tabela verdade:

NÚMERO DECIMAL	BIT_4	BIT_3	BIT_2	BIT_1	BIT_0	DISPLAY DA ESQUERDA						DISPLAY DA DIREITA							
						Ae	Be	Ce	De	Ee	Fe	Ge	Ad	Bd	Cd	Dd	Ed	Fd	Gd
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	1	1	1
2	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
3	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1	1	0
4	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0
5	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	1	0	0
6	0	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
7	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	1	1	1	1
8	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
9	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0
10	0	1	0	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	1
11	0	1	0	1	1	1	1	0	0	1	1	1	1	0	0	1	1	1	1
12	0	1	1	0	0	1	0	0	1	1	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	0
14	0	1	1	1	0	0	1	0	1	1	1	1	1	0	0	1	1	0	0
15	0	1	1	1	1	1	0	0	1	1	1	1	0	1	0	0	1	0	0
16	1	0	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	0
17	1	0	0	0	1	1	0	0	1	1	1	1	0	0	0	1	1	1	1
18	1	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0
19	1	0	0	1	1	1	1	0	0	1	1	1	0	0	0	0	1	0	0
20	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1
21	1	0	1	0	1	0	1	0	0	0	1	0	1	0	0	1	1	1	1
22	1	0	1	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0
23	1	0	1	1	1	1	0	1	0	0	1	0	0	0	0	0	1	1	0
24	1	1	0	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	0
25	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
26	1	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0
27	1	1	0	1	1	1	0	1	0	0	1	0	0	0	0	1	1	1	1
28	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
29	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0
30	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1

A partir do mapa-K, podemos chegar nas seguintes expressões booleanas para os segmentos:

$Ae=AB'C' + A'BD + A'BC$

$$B_e = 0$$

$$C_e = AB'C + ABC' + ABD'$$

$$D_e = AB'C' + A'BD + A'BC$$

$$E_e = AB'C' + A'BD + A'BC + BCD$$

$$F_e = A + BD + BC$$

$$G_e = A' + B'C'$$

$$A_d = A'B'C'D'E + A'B'CD'E' + ABC'D'E' + A'BC'DE + A'BCDE' + AB'CD'E$$

$$B_d = BCDE + AB'C'D'E' + A'B'CD'E + A'B'CDE' + ABC'D'E + ABC'DE'$$

$$C_d = A'B'C'DE' + A'BCD'E' + AB'CDE'$$

$$D_d = B'C'D'E + AB'D'E + BC'DE + A'B'CD'E' + ABC'D'E' + A'B'CDE + A'BCDE'$$

$$E_d = E + A'B'CD' + ABC'D' + A'BCD$$

$$F_d = B'C'D'E + A'B'C'D + A'BCD' + B'CDE + BC'DE + AB'CE + AB'CD$$

$$G_d = A'B'C'D' + A'BC'D + AB'D'E + AB'CD' + BC'DE + ABCD + A'B'CDE$$

12. CONCLUSÃO

Por fim, conseguimos realizar uma implementação bem-sucedida de uma Unidade Lógica e Aritmética (ULA) funcional, que executa operações essenciais como soma e subtração com sinal, comparações de magnitude e funções lógicas AND e XOR. Através da aplicação prática de conceitos da disciplina de Sistemas Digitais, como mapas de Karnaugh e portas lógicas, foi possível transformar o conhecimento teórico em um circuito integrado e funcional na plataforma Quartus. Dessa forma, o trabalho atendeu aos requisitos avaliativos e consolidou o aprendizado da equipe sobre o projeto e a arquitetura de componentes digitais centrais para a computação.