

UNIVERSIDADE DE SÃO PAULO (USP)
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO (ICMC)
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO (SCC)

Isabella Arão - 9265732
Marina Fagundes - 9265405
Robson Silva - 7233062

Entrega 2

São Carlos
2023

Isabella Arão - 9265732
Marina Fagundes - 9265405
Robson Silva - 7233062

Entrega 2

Relatório apresentado à disciplina Elementos de Lógica Digital, como parte dos requisitos para aprovação na matéria oferecida pela Universidade de São Paulo, na área de Computação.

Prof.: Danilo Hernane Spatti
Disciplina: Elementos de Lógica Digital
Turma: Turma SCC 0512

São Carlos
2023

1. Introdução	1
2. Unidade de Lógica Aritmética (ULA)	1
2.1. Somador e subtrator de 1 bit	1
2.2. Somador e subtrator de 4 bits	4
2.3. Número negativo	5
3. Memória	5
3.1 Número 08 (1000)	6
3.2 Número 7 (0111)	6
3.3 Número 6 (0110)	7
3.4 Número 5 (0101)	7
3.6 Memória completa	7
4. Contador síncrono de 2 bits	9
5. Chaves binárias	10
6. Quartus	10
7. Displays de 7 segmentos	14
7.1. Primeiro e segundo display	14
7.2. Terceiro display	15
7.3. Quarto display	15
7.4. Quinto display	16
8. Calculadora	17
9. Referências bibliográficas	19

1. Introdução

O presente trabalho tem como objetivo a construção de uma calculadora a ser implementada na FPGA, a partir da construção de uma unidade lógico aritmética, memória de 4x4 (que contém os números 8, 7, 6 e 5), um contador síncrono de 2 *bits*, chaves binárias e *displays* de 7 segmentos (a exemplo do que foi desenvolvido na entrega 01). Os componentes da calculadora serão apresentados a seguir, como parte do processo de confecção.

2. Unidade de Lógica Aritmética (ULA)

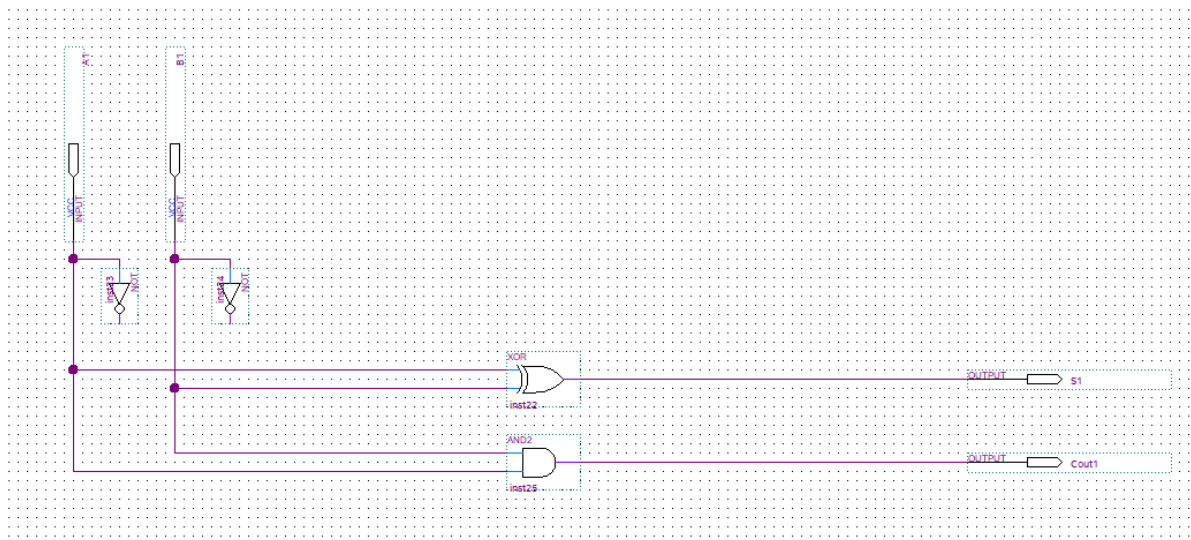
A Unidade Lógico Aritmética deve conter um somador de 4 *bits* e um subtrator de 4 *bits*. A operação deve ser selecionada pela chave SW8.

2.1. Somador e subtrator de 1 *bit*

O primeiro passo para a construção da Unidade Lógica Aritmética (ULA) é a construção de meio somador, meio subtrator, somador completo e subtrator completo de 1 *bit*.

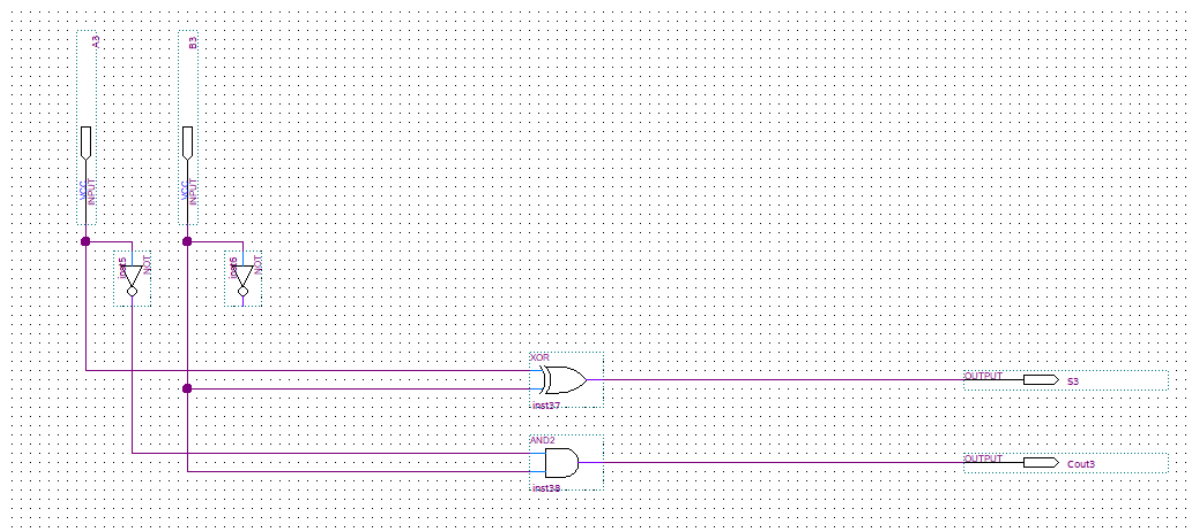
A seguir, apresentamos os circuitos de cada uma dessas operações, respectivamente.

Figura 1 - Meio Somador.



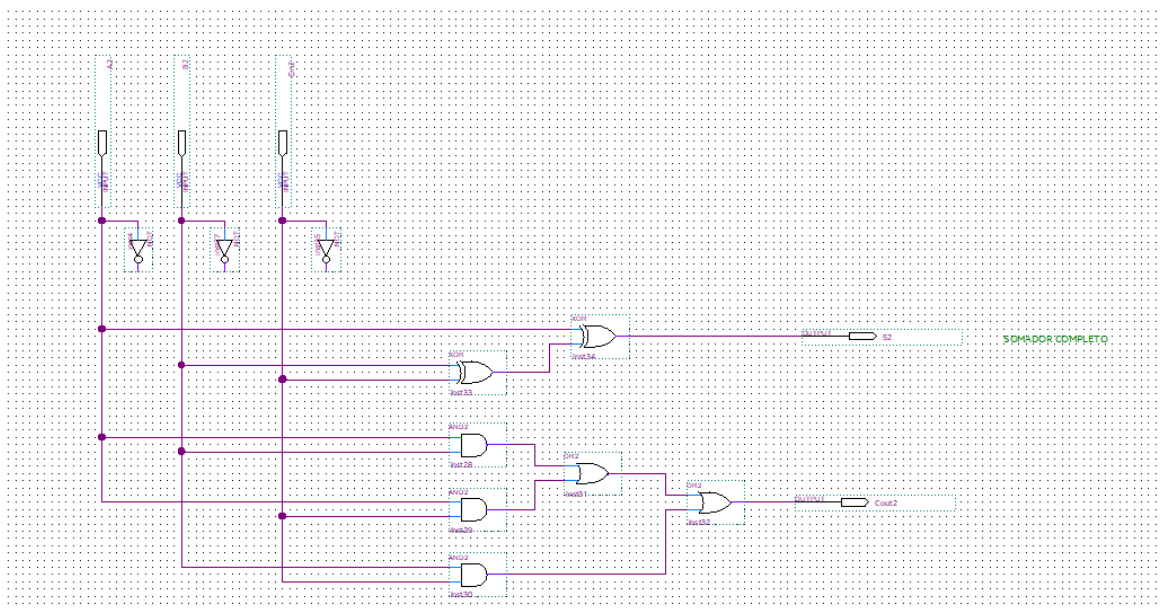
Fonte: elaborado pelos autores.

Figura 2 - Meio Subtrator.



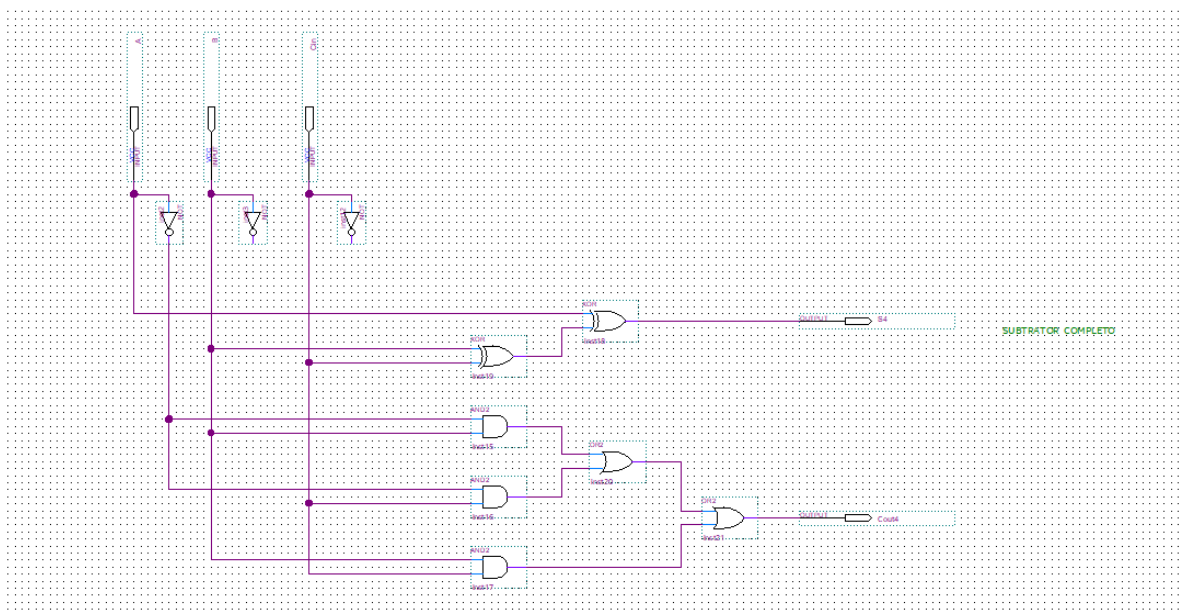
Fonte: elaborado pelos autores.

Figura 3 - Somador Completo.



Fonte: elaborado pelos autores.

Figura 4 - Subtrator Completo.

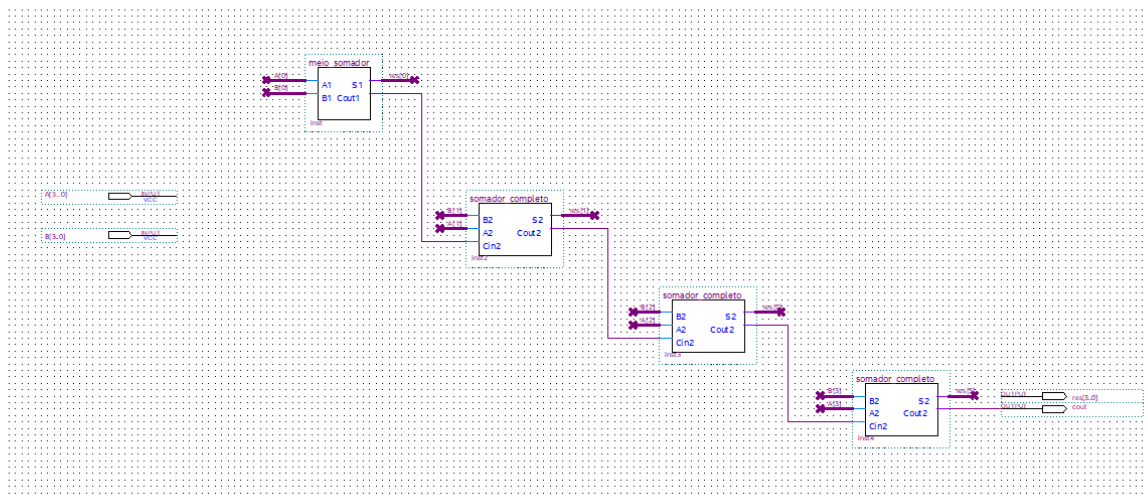


Fonte: elaborado pelos autores.

2.2. Somador e subtrator de 4 bits

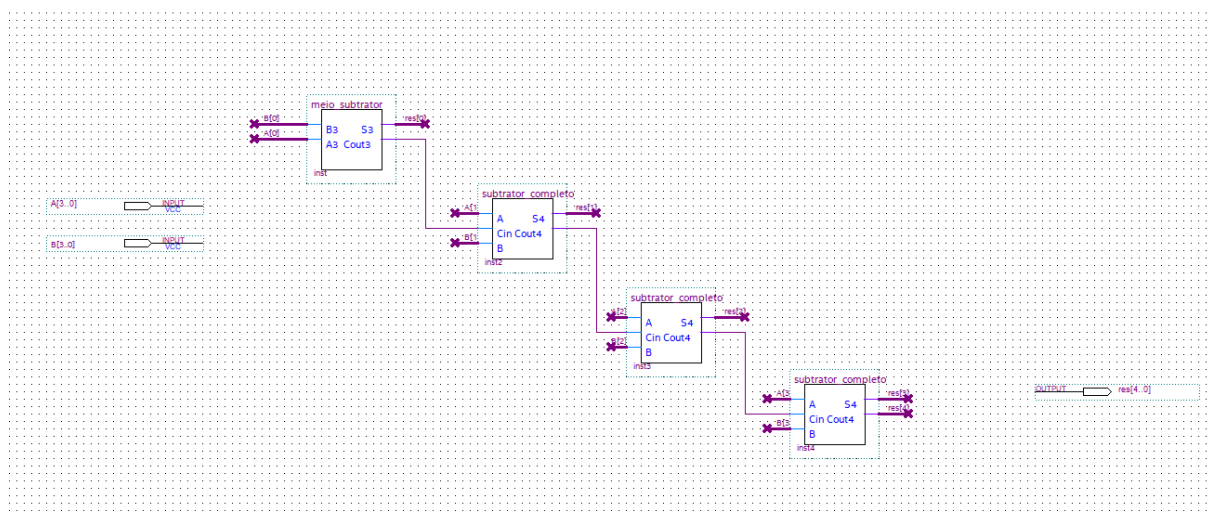
O somador (e o subtrator) de quatro *bits* são resultado da combinação de um meio somador (ou subtrator) e três somadores (ou subtratores) completos, em que a saída *Cout* de cada bloco vira a entrada *Cin* do próximo, conforme vemos a seguir.

Figura 5 - Somador de 4 bits.



Fonte: elaborado pelos autores.

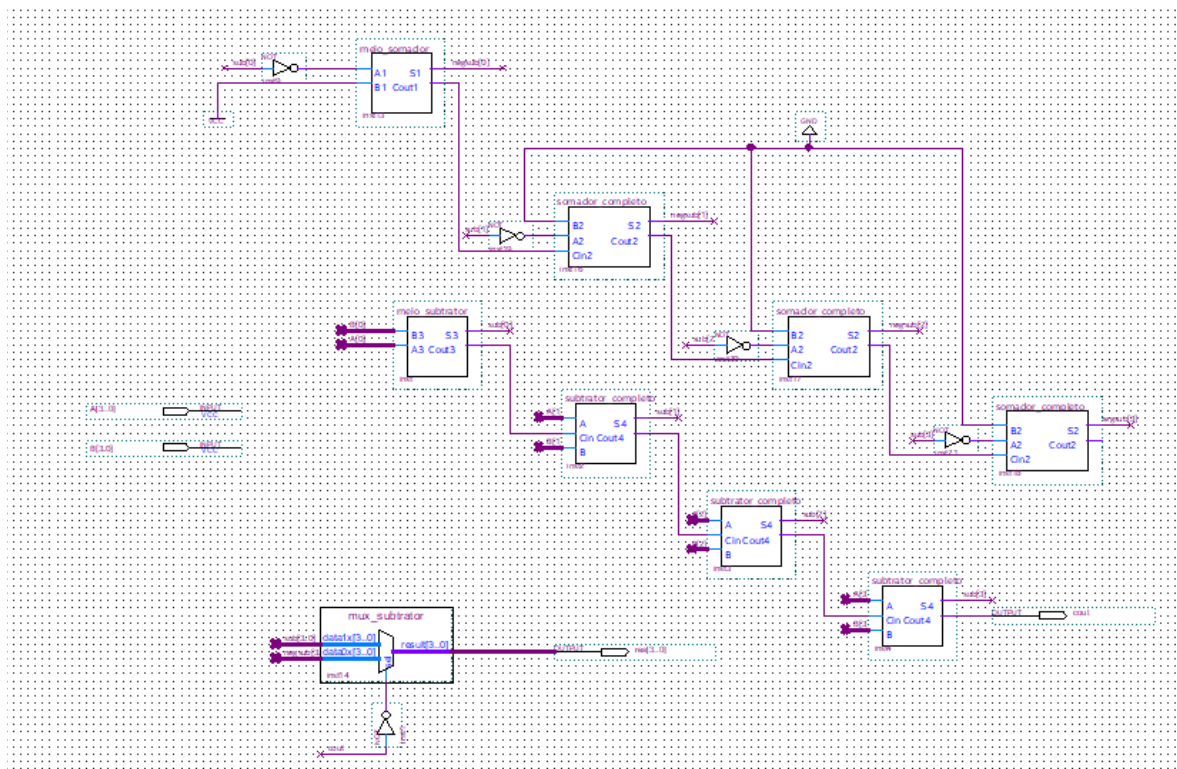
Figura 6 - Subtrator de 4 *bits* (incompleto).



2.3. Número negativo

No caso de subtração, enfatizamos que há a possibilidade de a operação resultar negativa e, nesse caso, o circuito precisa de uma lógica extra, o complemento de 2. Para incluir tais situações, alteramos o nosso bloco de subtrator de 4 *bits*, conforme a imagem abaixo.

Figura 7 - Subtrator de 4 *bits* (completo).



Fonte: elaborado pelos autores.

Assim, adicionamos o complemento de 2 dentro do bloco do subtrator de 4 *bits* (parte superior). Dessa forma, as contas já saem com os resultados ajustados para todos os casos, inclusive os em que o resultado é negativo.

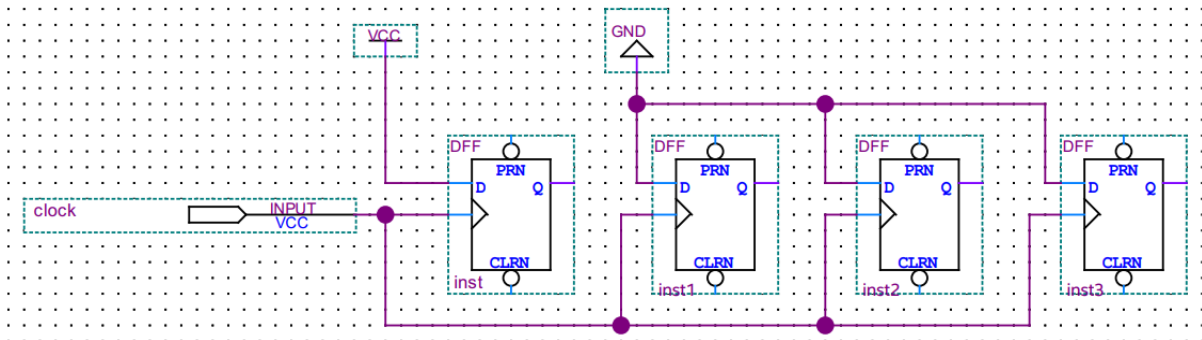
3. Memória

Começamos a confecção da memória usando os *flip-flops* tipo D (DFF) da biblioteca do *Quartus*. Cada linha corresponde a um número binário, indo de 8 a 5. Conectamos a entrada D no modo VCC, quando o número correspondente é 1, e

em GND, quando o número correspondente é 0. A seguir, estão as imagens de cada um dos números, antes de conectar as saídas ao MUX.

3.1 Número 08 (1000)

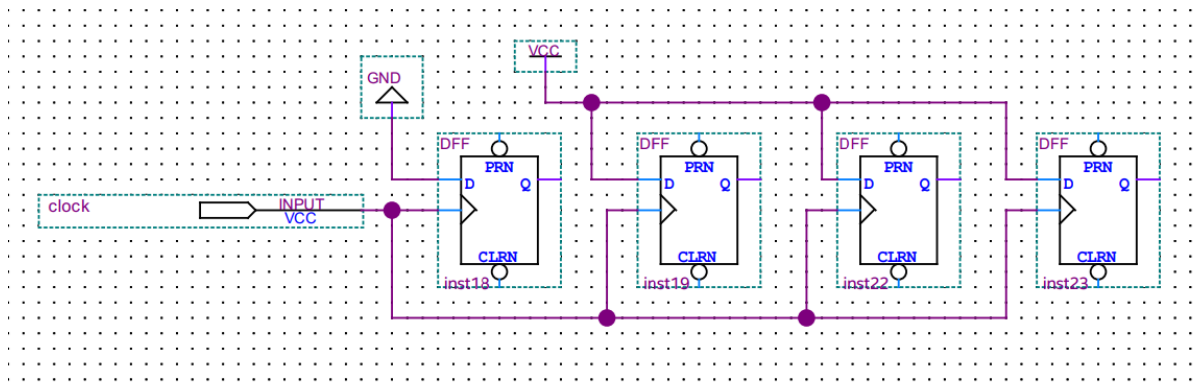
Figura 8 - Número 08.



Fonte: elaborado pelos autores.

3.2 Número 7 (0111)

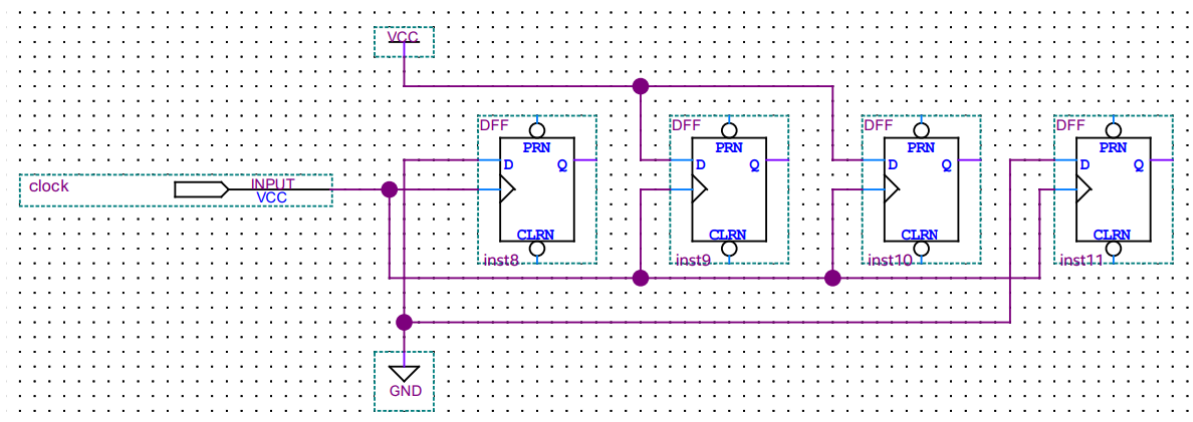
Figura 9 - Número 07.



Fonte: elaborado pelos autores.

3.3 Número 6 (0110)

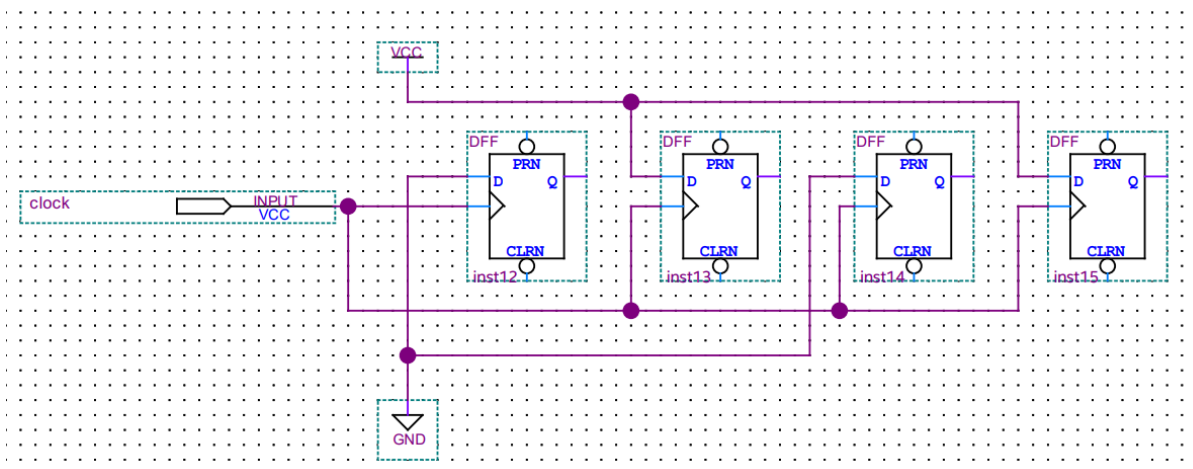
Figura 10 - Número 06.



Fonte: elaborado pelos autores.

3.4 Número 5 (0101)

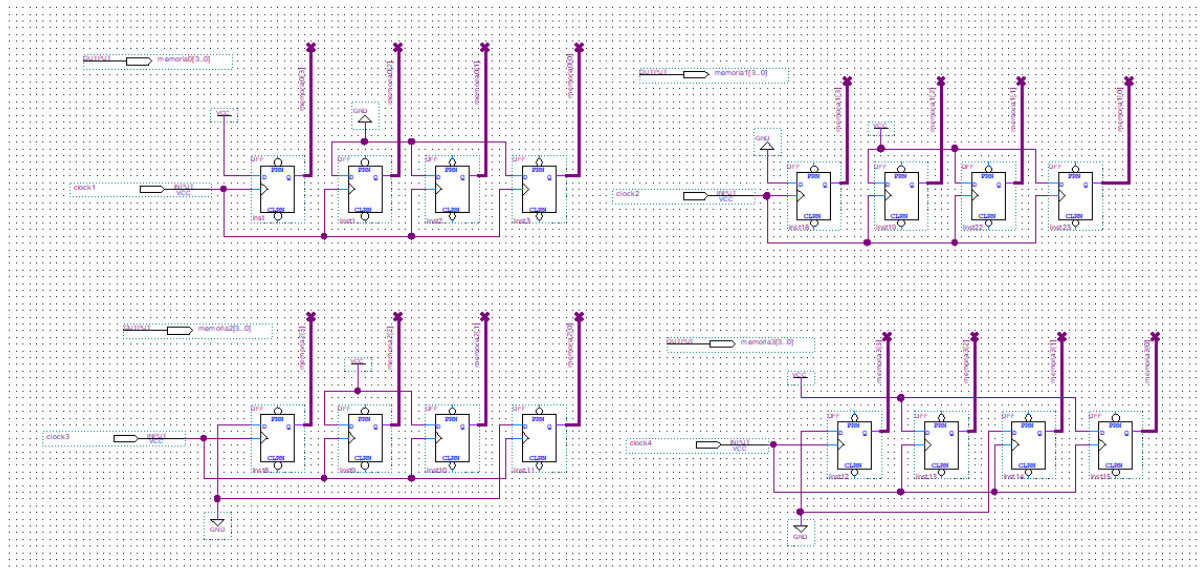
Figura 11 - Número 05.



Fonte: elaborado pelos autores.

3.6 Memória completa

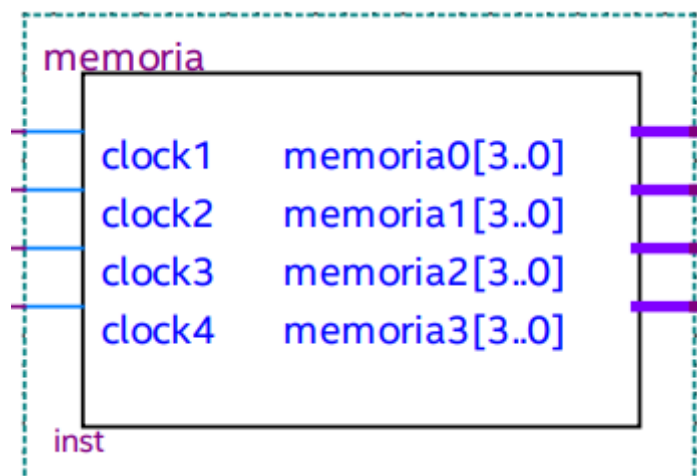
A seguir, está a imagem da memória completa. Cada número, de 8 a 5, teve suas saídas conectadas a um vetor, que está ligado ao MUX de 4 *bits* no arquivo final, por meio de ligações em vetor.

Figura 12 - Memória Completa.

Fonte: elaborado pelos autores.

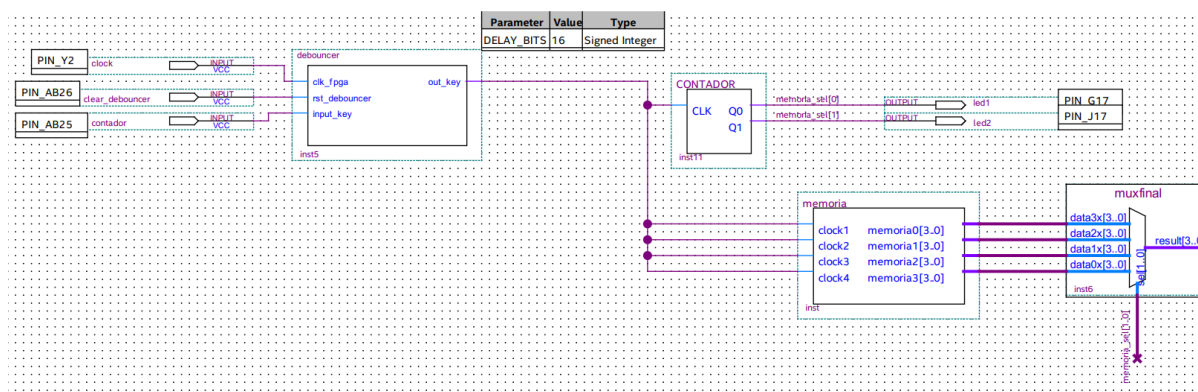
Após a confecção da memória, foi criado um bloco para ser inserido no arquivo final.

Para a limpeza da memória, inserimos o *Debouncer*, com um pino *Clear*, conectado à memória.

Figura 13 - Memória 4x4 (bloco).

Fonte: elaborado pelos autores.

Figura 14 - Debouncer, contador e memória.

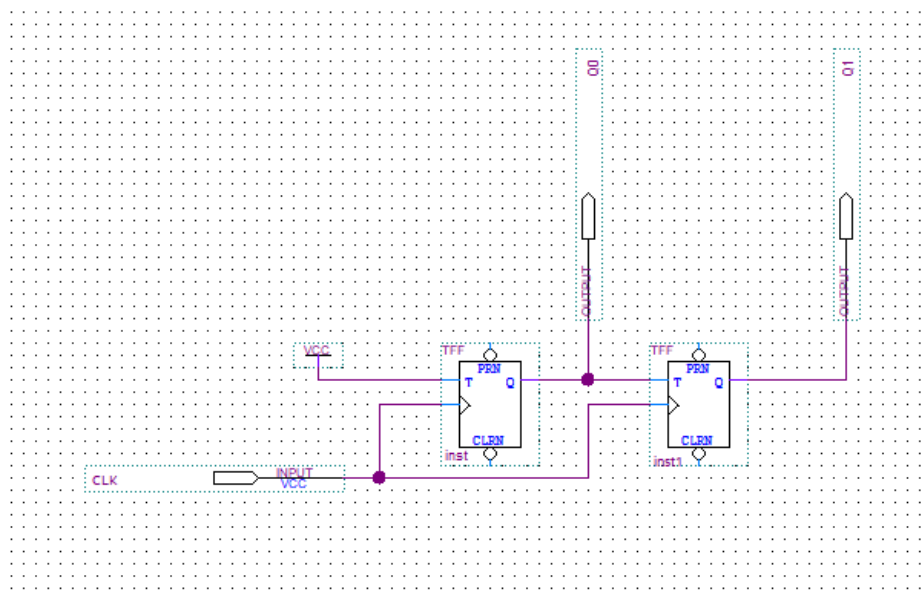


Fonte: elaborado pelos autores.

4. Contador síncrono de 2 bits

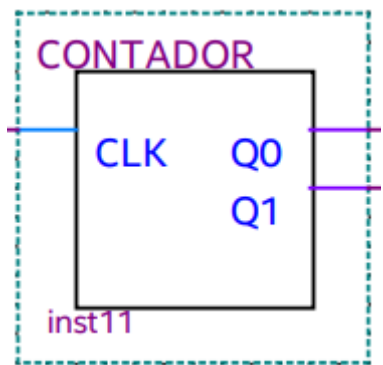
Contadores síncronos são circuitos digitais que geram determinada sequência de estados, sob o comando de um sinal de *clock*. Para esse projeto, foi necessária a elaboração de um contador síncrono de 2 bits, utilizando *flip-flops* tipo T.

Figura 15 - Contador de 2 bits (circuito).



Fonte: elaborado pelos autores.

Figura 16 - Contador de 2 bits (bloco).

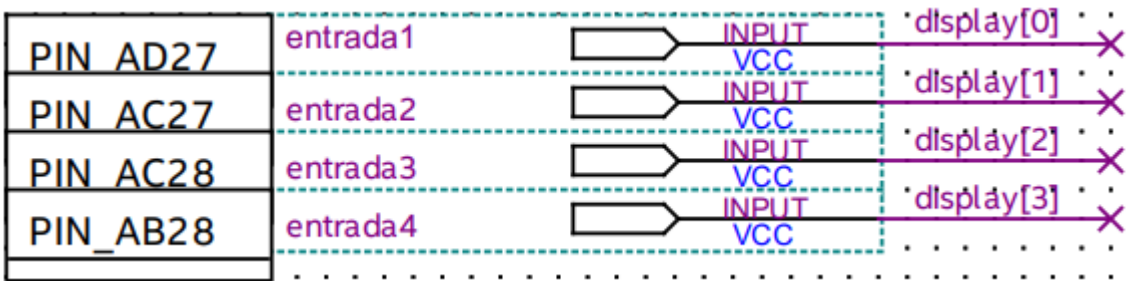


Fonte: elaborado pelos autores.

5. Chaves binárias

As chaves binárias usadas no projeto foram as SW0, SW1, SW2, e SW3. Abaixo está a imagem das chaves com seus respectivos pinos.

Figura 17 - Chaves binárias.

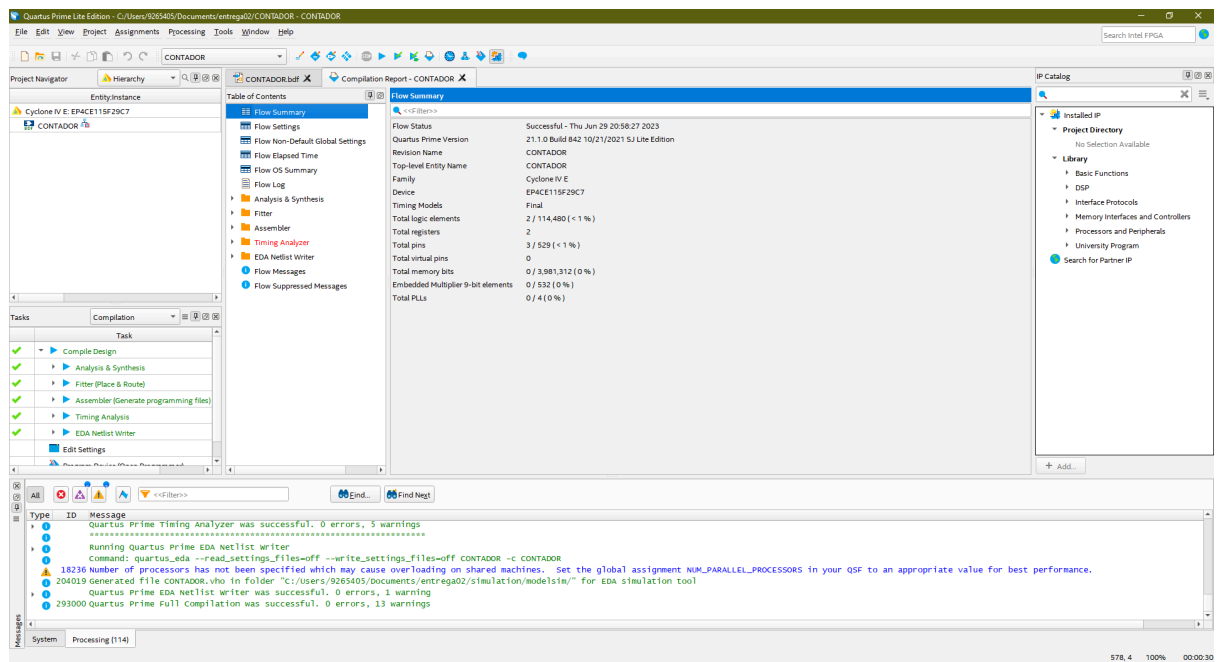


Fonte: elaborado pelos autores.

6. Quartus

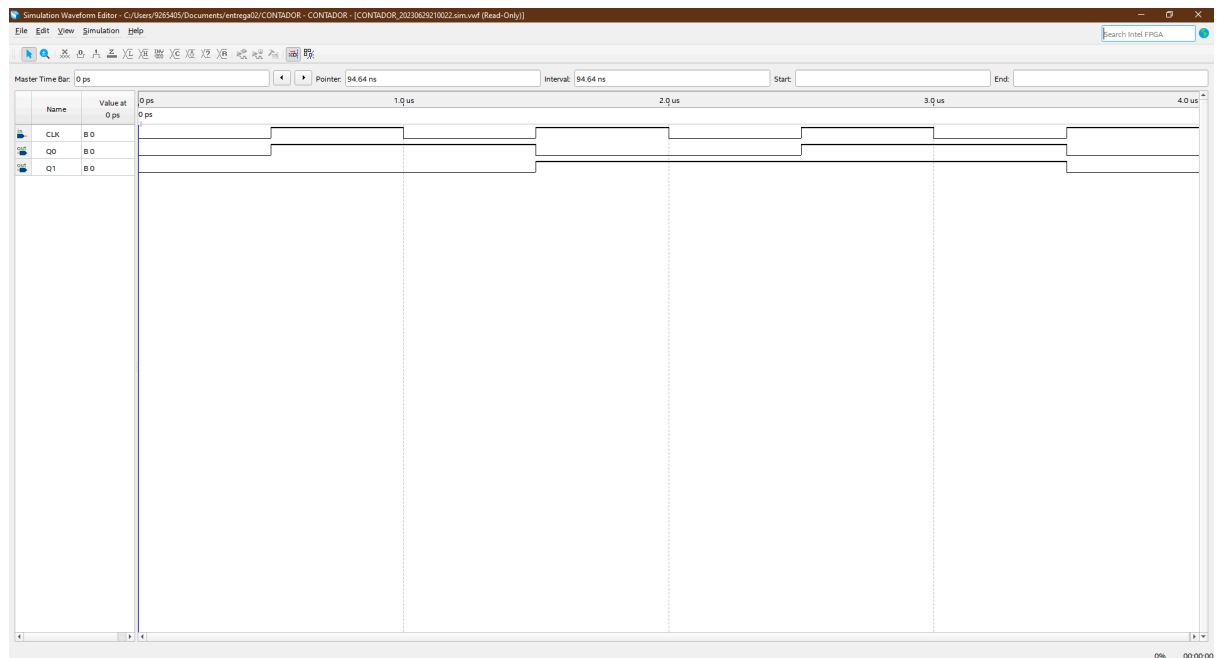
Nesse tópico, apresentamos as compilações e simulação do contador e da memória, além da compilação do circuito completo. Não realizamos as compilações do subtrator e do somador, uma vez que são elementos assíncronos.

Figura 18 - Compilação do contador.



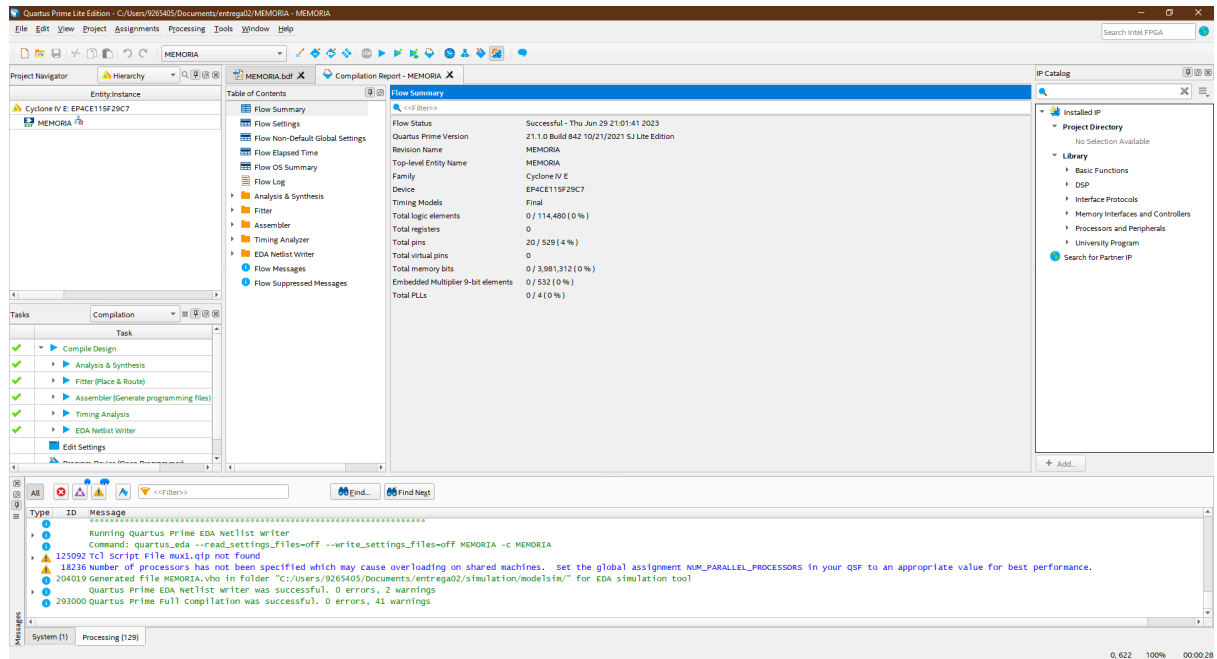
Fonte: elaborado pelos autores.

Figura 19 - Simulação do contador.



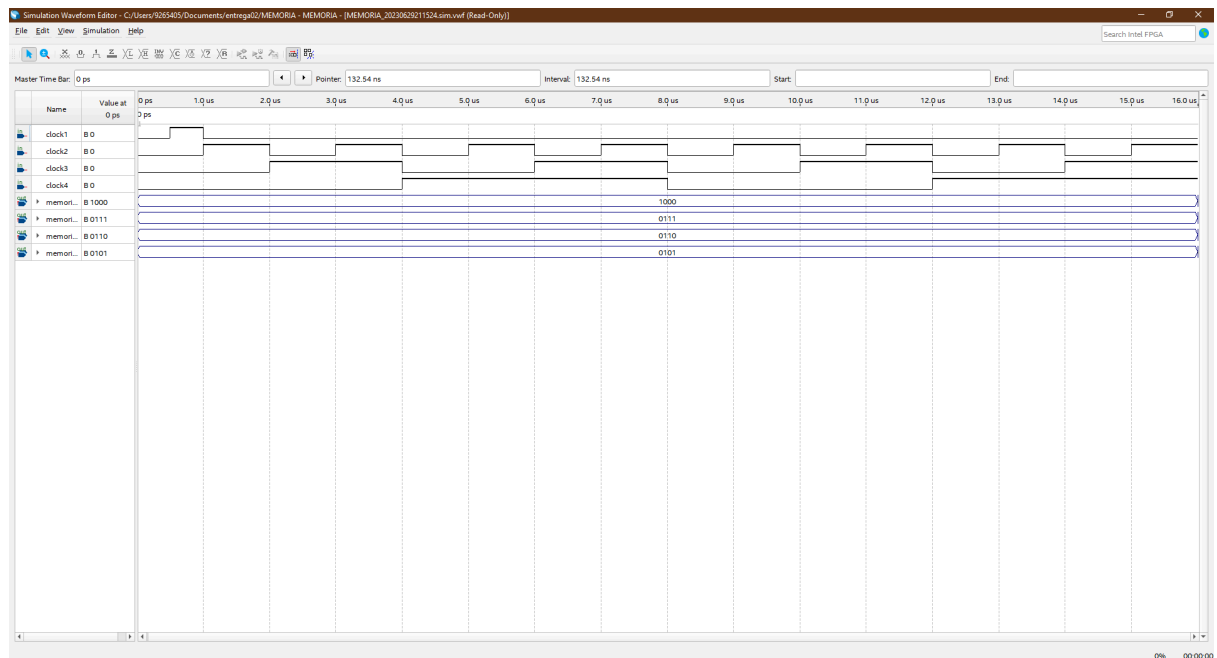
Fonte: elaborado pelos autores.

Figura 20 - Compilação da memória.



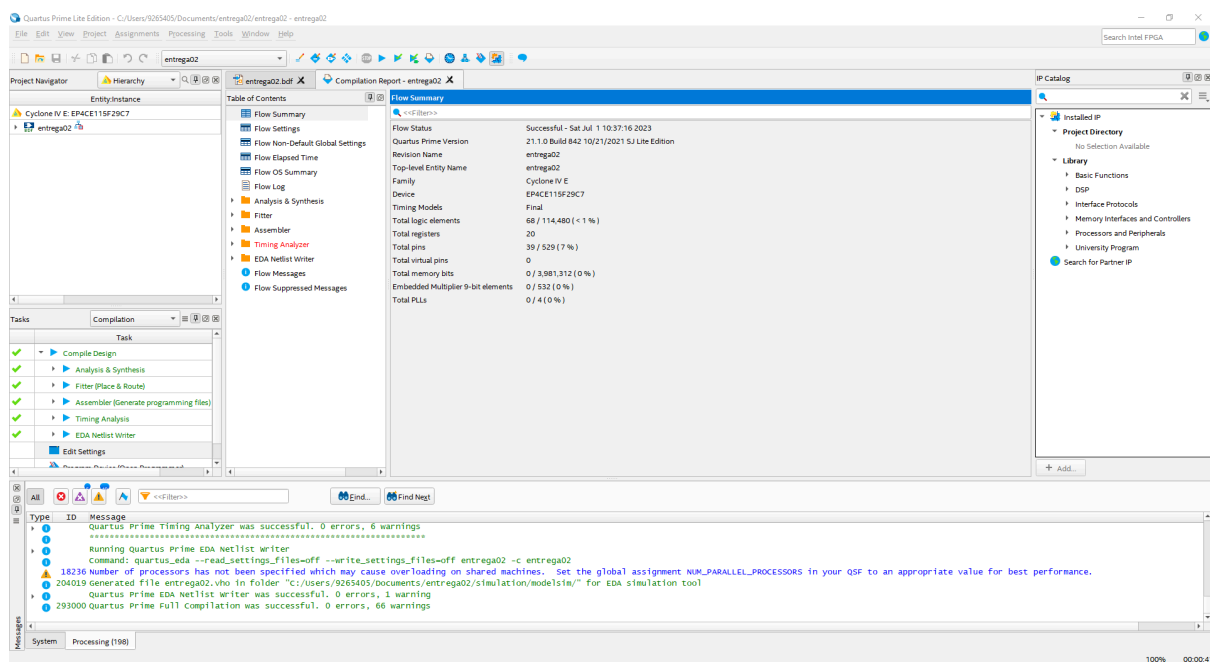
Fonte: elaborado pelos autores.

Figura 21 - Simulação da memória.



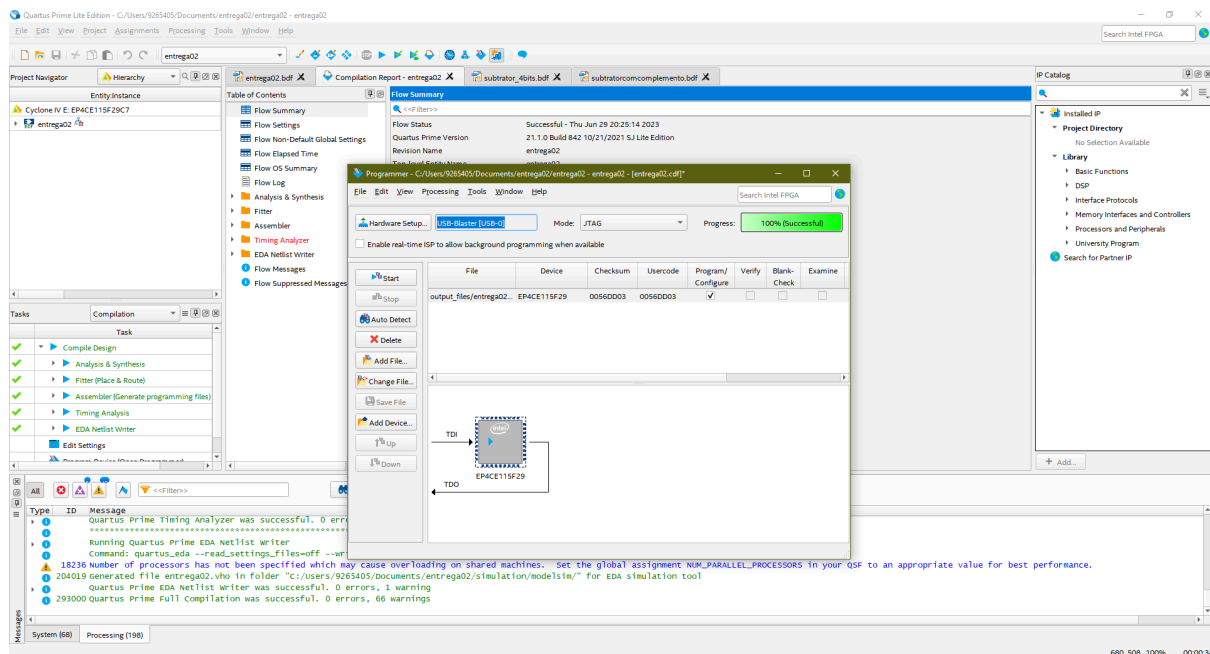
Fonte: elaborado pelos autores.

Figura 22 - Compilação do circuito.



Fonte: elaborado pelos autores.

Figura 23 - Passagem do circuito para a FPGA.



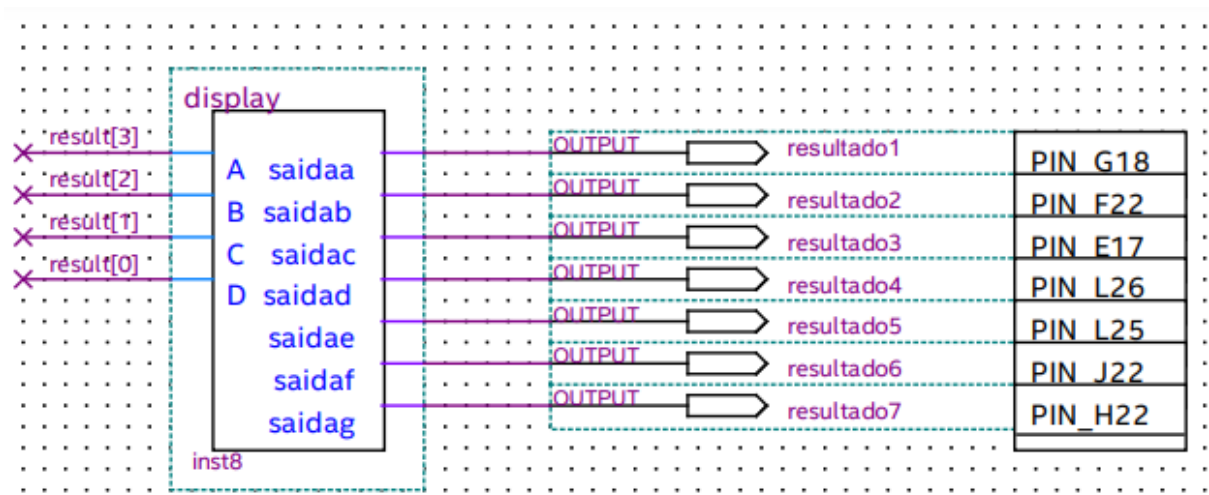
Fonte: elaborado pelos autores.

7. *Displays* de 7 segmentos

Foram empregados 5 displays de 7 segmentos: HEX[0] e HEX[1] mostrando o resultado da operação, HEX[2] os números das chaves binárias, HEX[3] os dados da memória e, por fim, o HEX[4], que mostra a operação que está sendo realizada (sinal de menos aparece quando é feita subtração e nenhum sinal aparece quando é feita soma).

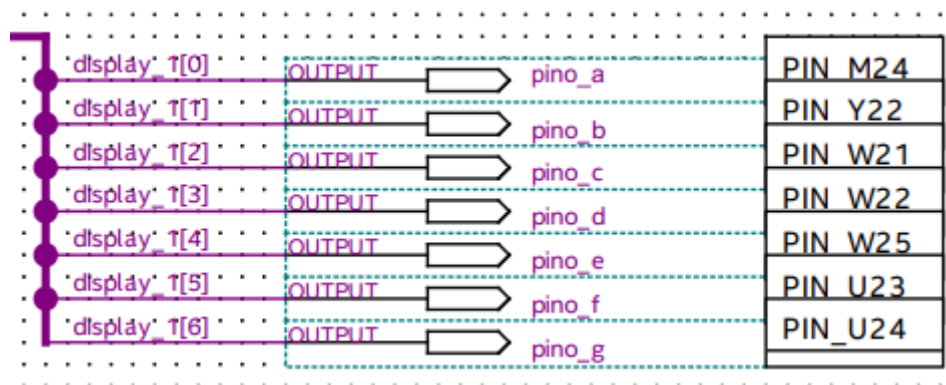
7.1. Primeiro e segundo *display*

Figura 24 - Display HEX[0].



Fonte: elaborado pelos autores.

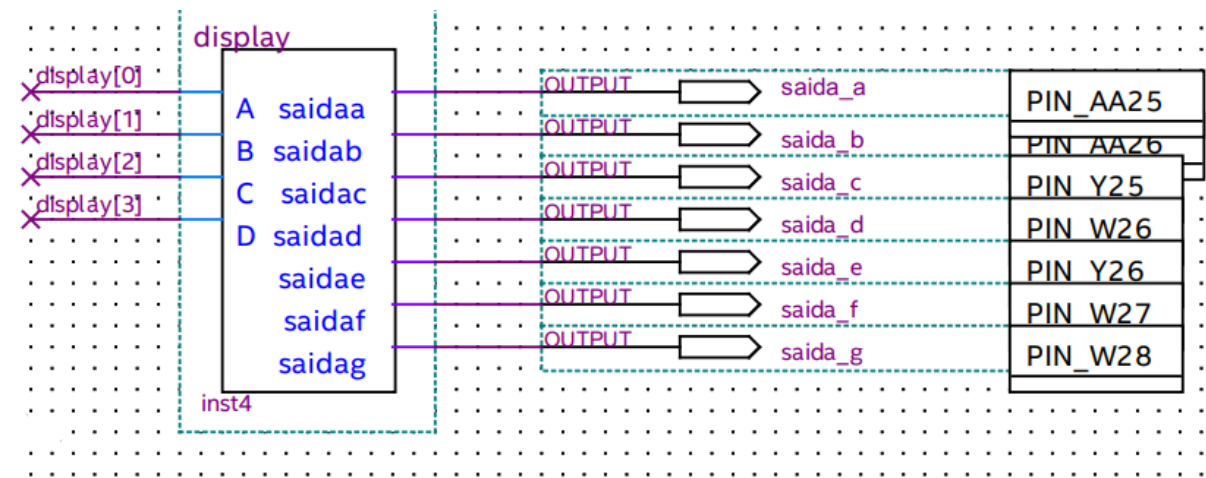
Figura 25 - Display HEX[1].



Fonte: elaborado pelos autores.

7.2. Terceiro *display*

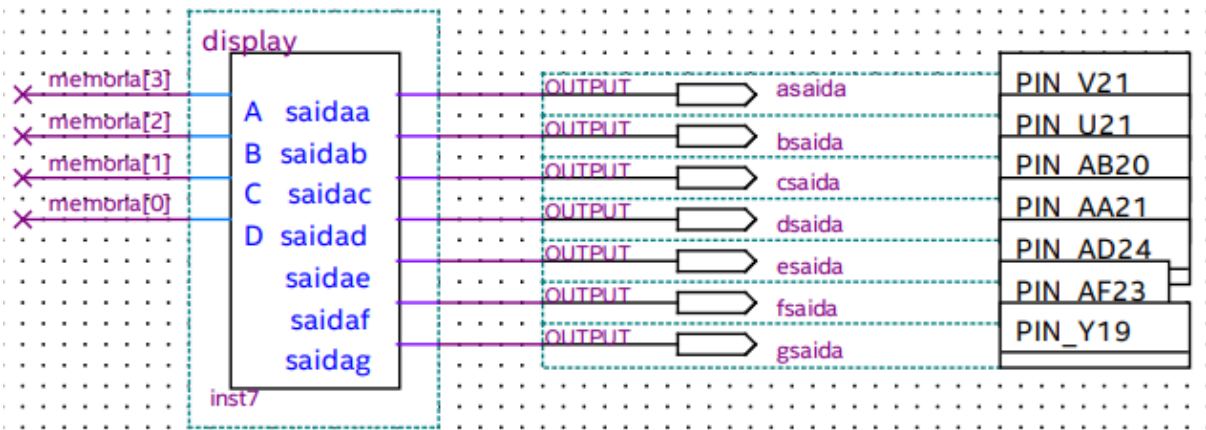
Figura 26 - Display HEX[2].



Fonte: elaborado pelos autores.

7.3. Quarto *display*

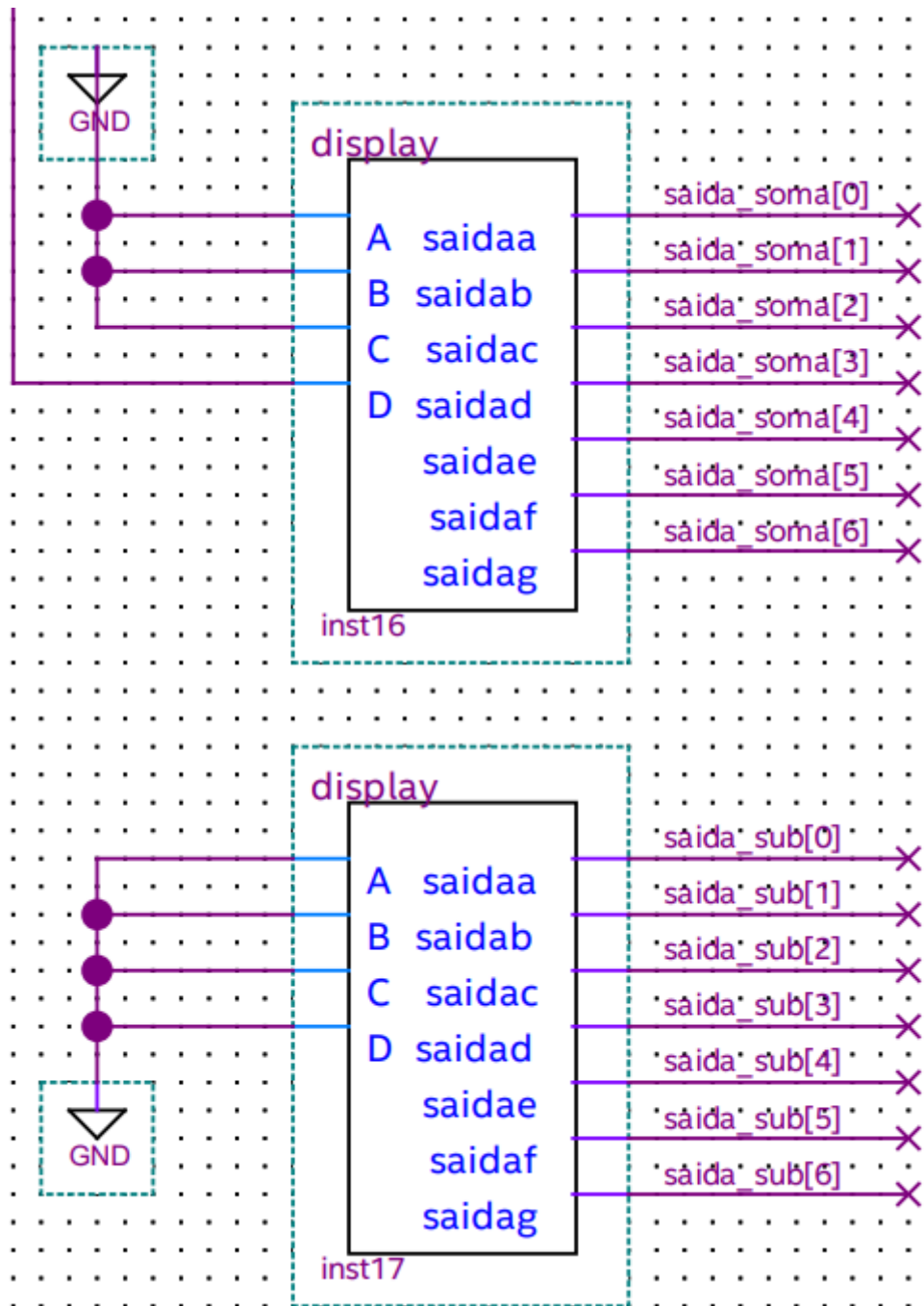
Figura 27 - Display HEX[3].



Fonte: elaborado pelos autores.

7.4. Quinto *display*

Figura 28 - HEX[4].

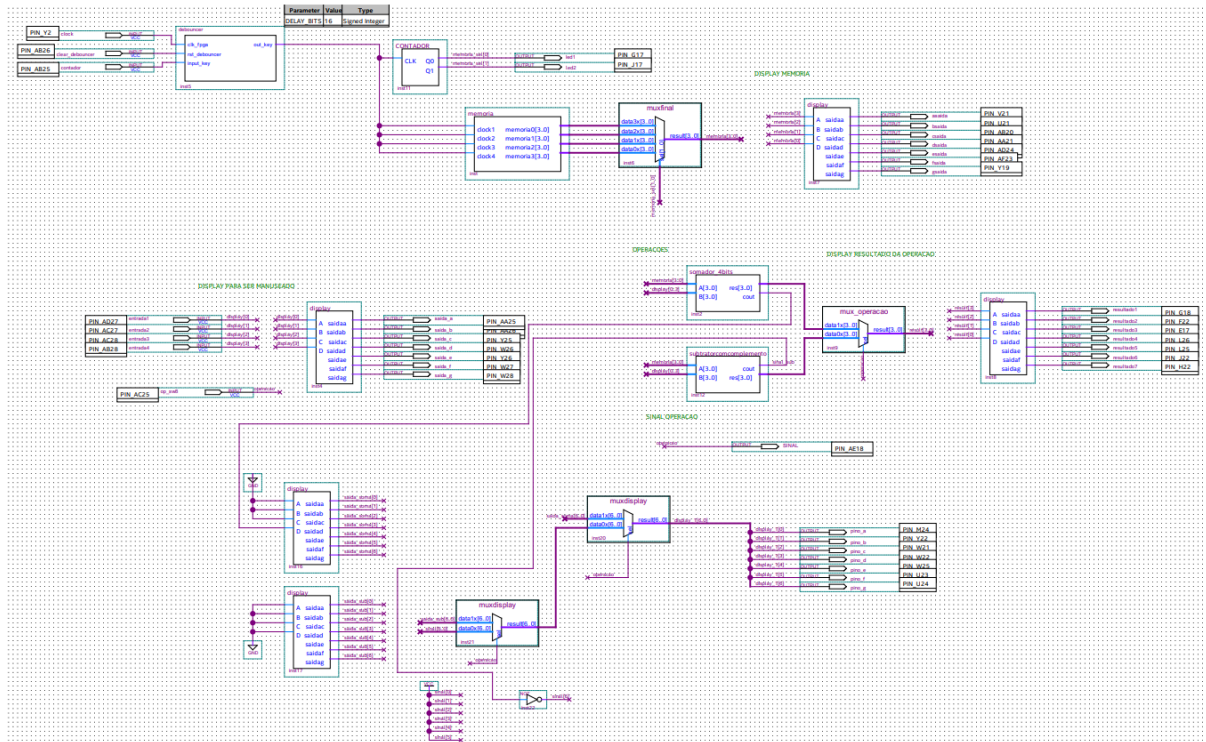


Fonte: elaborado pelos autores.

8. Calculadora

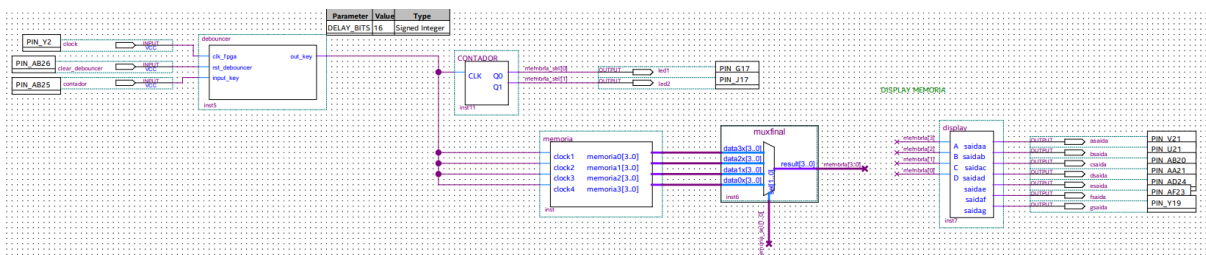
Abaixo, segue imagem do circuito completo da calculadora.

Figura 29 - Circuito completo.



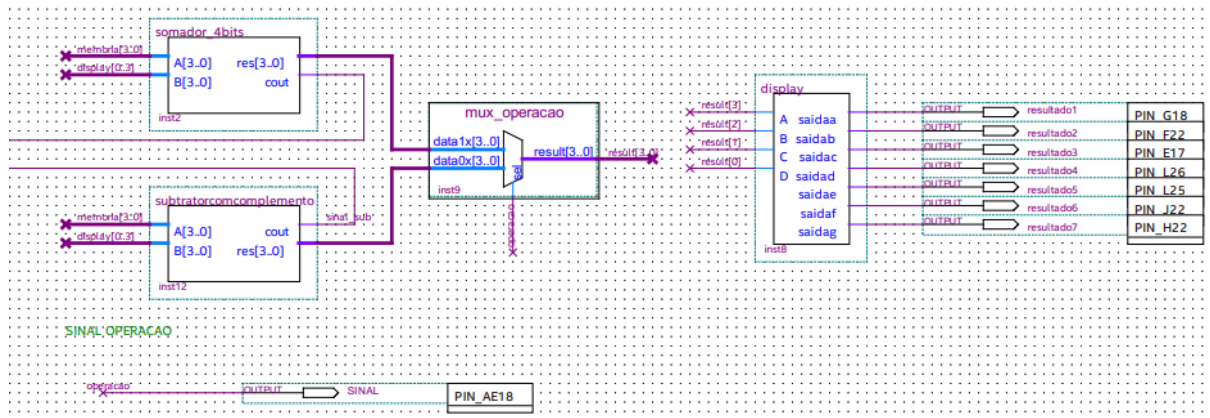
Fonte: elaborado pelos autores.

Figura 30 - Debouncer, contador e memória.



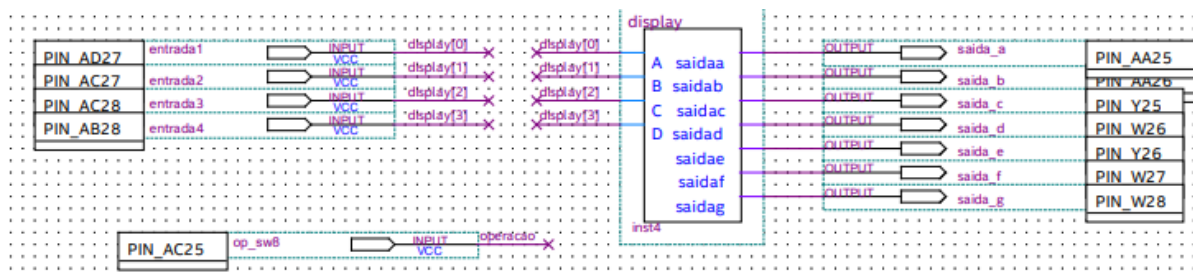
Fonte: elaborado pelos autores.

Figura 31 - Operações e resultado da operação.



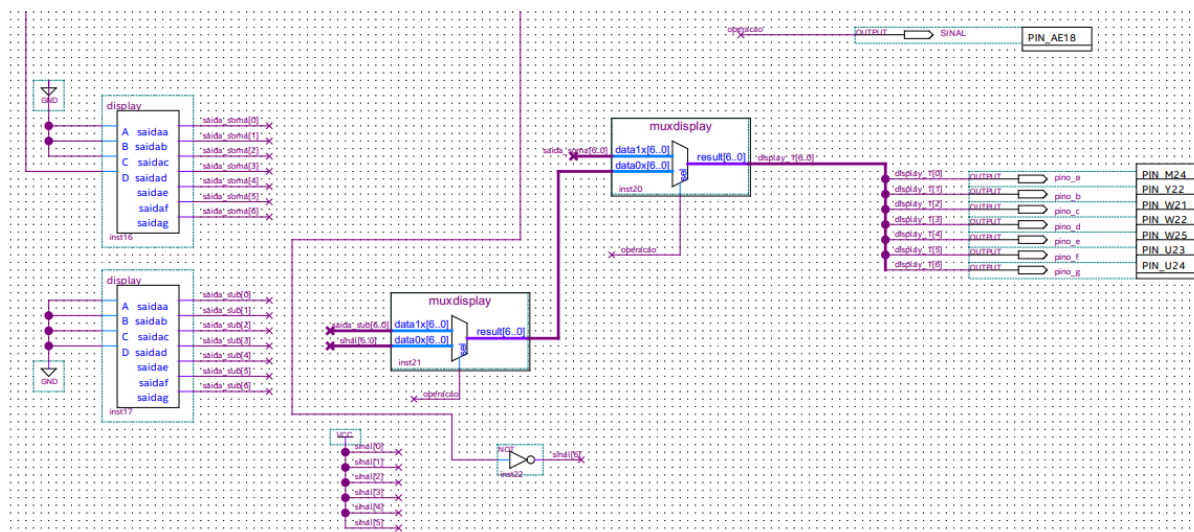
Fonte: elaborado pelos autores.

Figura 32 - Display das chaves binárias e pino da escolha da operação.



Fonte: elaborado pelos autores.

Figura 33 - Sinal e display de resultado da operação, display da escolha da operação.



Fonte: elaborado pelos autores.

9. Referências bibliográficas

SPATTI, D. H. **Material didático apresentado na disciplina SCC0512 - Elementos de Lógica Digital**. São Carlos/SP. s.d.

TERASIC TECHNOLOGIES. **DE2-115 User Manual**. 2010.