



CALCULADORA DIGITAL

Projeto Final

GRUPO C6:

Akin Sangiacomo Bazila, 221002002
221002002@aluno.unb.br

Guilherme Fornari Leonel, 221017023
221017023@aluno.unb.br

Gustavo Borges Caldas, 221030016
221030016@aluno.unb.br

Pedro Avila Beneveli, 221001972
221001972@aluno.unb.br

Dep. Ciência da Computação -
Universidade de Brasília (UnB)

CIC0231 - Laboratório de Circuitos
Lógicos



UnB | CIC

RESUMO

O Projeto Final consiste em implementar uma calculadora digital simples baseada em FPGA. A calculadora contém uma ULA (Unidade Lógica Aritmética) que realiza as operações de adição, subtração e multiplicação de números inteiros de 0 a 9. Os dados de entrada, os quais são fornecidos por intermédio de um teclado matricial 4x4 de membrana, devem seguir a seguinte ordem para seu correto funcionamento: receber um primeiro número, receber um dos três operadores aritméticos, receber o segundo número e, por fim, receber o operador '=', para calcular e mostrar o resultado, em base decimal e base binária, num conjunto de displays de 7 segmentos (HEX6, HEX4, HEX1 e HEX0) e LEDs (LEDR[17:14], LEDG[1:0] e LEDR[12:9]), respectivamente, do kit de desenvolvimento DE2. Além disso, o LEDG[8] foi utilizado para indicar que o sistema está pronto para iniciar uma nova operação, caso o resultado da operação fosse negativo, o sinal do mesmo era mostrado no display HEX2 e o botão KEY[0] foi utilizado como o reset do sistema.

ABSTRACT

The Final Project consists of implementing a simple digital calculator based on FPGA. The calculator contains an ALU (Arithmetic Logic Unit) that performs the addition, subtraction and multiplication operations of whole numbers from 0 to 9. The input data, which are provided through a 4x4 membrane matrix keyboard, must follow the following order for its correct operation: receive a first number, receive one of the three arithmetic operators, receive the second number and, finally, receive the '=' operator, to calculate and display the result, in decimal base and binary base, on a set of 7-segment displays (HEX6, HEX4, HEX1 and HEX0) and LEDs, respectively, from the DE2 development kit. In addition, the LEDG[8] was used to indicate that the system is ready to begin a new calculation, if the result of the operation was negative, the same sign was shown on the HEX2 display and the KEY[0] button was used like the system reset.

1. INTRODUÇÃO

No trabalho, foi projetado e montado um circuito digital síncrono que consiste numa calculadora a qual permite a leitura de um número, de 0 a 9, do teclado, seguido de uma operação (A = multiplicação, B = subtração e C = adição), a leitura de um segundo número e, por fim, com o acionamento a tecla "D" (=), o resultado da operação em questão será verificado em dois displays de 7 segmentos, com um terceiro representando o sinal do mesmo.

Na configuração das entradas e saídas na placa FPGA, um teclado matricial 4x4 foi utilizado como fornecedor de dados da calculadora. Inicialmente, é necessário ressaltar que um LED verde foi utilizado como um indicador que o sistema está pronto para iniciar uma nova operação. Após a entrada do primeiro número, com valor em decimal, o mesmo é mostrado em um display de 7 segmentos e sua representação em binário, nos LEDs vermelhos logo abaixo do display. Em seguida, tem-se a entrada da operação, que pode ser visualizada em 2 LEDs verdes, nos quais 01 significa adição, 10 subtração e 11 multiplicação. Logo após, ocorre a entrada do segundo número, que possui as mesmas características do primeiro, mas é mostrado em outro display e sua representação em binário, em outros LEDs vermelhos. Finalmente, a tecla "D" (=) deve ser pressionada para que o resultado da operação seja verificado em outros dois displays.

Por fim, é importante destacar que o sistema conta com um display que indica o sinal do resultado da operação. Além disso, um botão tem como função reiniciar o sistema (reset), o que faz o mesmo voltar ao estado inicial, esperando o primeiro número.

1.1. Objetivos

O objetivo do trabalho era implementar uma calculadora com uma ULA que calcula soma, subtração e multiplicação de dois números inteiros decimais de um só dígito. Para isso, o grupo deveria obter conhecimentos acerca de SystemVerilog e, principalmente, do funcionamento do software Quartus II e do kit de desenvolvimento DE2. Os dados de entrada devem ser obtidos por um mini-teclado matricial 4x4 de membrana e o resultado da operação deve aparecer num display de 7 segmentos do kit.

Apesar do objetivo principal ser a confecção e apresentação da calculadora, sabe-se que há outros objetivos por trás do trabalho. Dessa forma, infere-se que o trabalho fomenta o estudo da linguagem de descrição de hardware utilizada e da absorção do conteúdo visto durante toda a disciplina de Laboratório de Circuitos Lógicos, uma vez que para a implementação do circuito foram necessários conhecimentos adquiridos ao longo do semestre, possibilitando a construção de um sistema digital eficiente e funcional.

2. METODOLOGIA

2.1. Unidade Lógica Aritmética

Inicialmente, o grupo se dedicou a montar a parte central do circuito de calculadora: o circuito responsável por realizar as operações de soma, subtração e multiplicação. Esse circuito é chamado de Unidade Lógica Aritmética, ou ULA. Para a montagem desse circuito no *Quartus II*, foi utilizado SystemVerilog, já que, ao fazer esse circuito em linguagem de descrição de hardware, é possível realizar a descrição comportamental do circuito e assim evitar o projeto de um circuito combinacional que realiza multiplicação, algo possivelmente muito complexo, como pode ser visto na imagem abaixo.

Com isso, determinamos três valores, que recebem os valores resultantes dessas operações aritméticas, e, utilizando um multiplexador que tem como entrada de seleção a operação selecionada pelo usuário, é feita a decisão de qual dessas operações que terá seu resultado como saída do circuito. No caso da ULA receber uma operação '00', a qual não existe na calculadora especificada, o circuito retorna apenas 0, para evitar erros inesperados.

```
module ula(
    input logic unsigned [3:0] num1,
    input logic unsigned [3:0] num2,
    input logic operacao [1:0],
    output logic [7:0] resultado
);

    logic signed [7:0] soma;
    logic signed [7:0] sub;
    logic signed [7:0] mult;

    assign soma = num1 + num2;
    assign sub = num1 - num2;
    assign mult = num1 * num2;

    mux4 muxUla (.d0(7'b0), .d1(soma), .d2(sub), .d3(mult), .selection(operacao), .S(resultado));

endmodule

module mux4(
    input logic [7:0] d0, d1, d2, d3,
    input selection[1:0],
    output [7:0] S
);
    assign S = selection[1] ? (selection[0] ? d3 : d2) : (selection[0] ? d1 : d0);
endmodule
```

Implementação em SystemVerilog da Unidade Lógica Aritmética.

2. METODOLOGIA

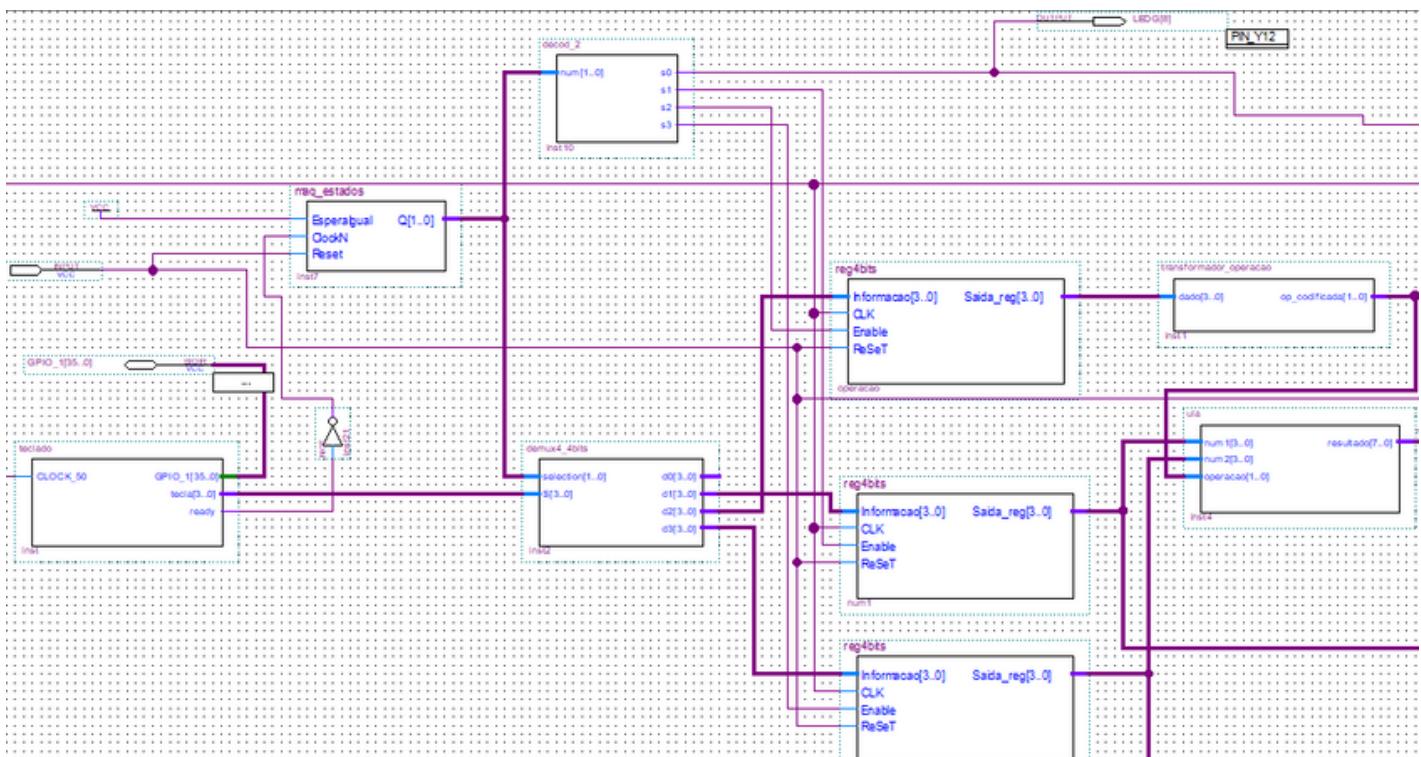
2.2. Máquina de Estados e Registradores

Após a implementação da ULA, o grupo mirou sua atenção em como receber os dados captados do teclado e guardá-los separadamente, para que cada dado seja levado as partes corretas do circuito. Com isso, foi decidido utilizar uma máquina de estados, de forma que um estado representaria a espera pelo 1º número, outro a espera da operação, outro a espera de um 2º número, e por fim a espera pelo sinal de '='. Portanto, utilizando a codificação 00, 01, 10 e 11 para os respectivos estados citados anteriormente, foi montado essa máquina de Moore, com a sequência 00 -> 01 -> 10 -> 11 -> 00 (reinicia), sendo suas entradas um sinal de *clock* e um sinal de *Reset*.

Após determinar essa sequência, determinou-se que o dado esperado em cada estado seria captado no próximo. Portanto, o 1º número seria recebido no estado 01, a operação no estado 10, o 2º número no estado 11 e o sinal de '=' no estado 00. Portanto, concluiu-se que circuitos de memória, com o objetivo de preservar esses dados para exibição e uso na ULA deveria ocorrer nesses estados especificados.

Então, o grupo trabalhou para criar circuitos combinacionais que permitiam que uma certa saída fosse um quando um certo estado estivesse ativo e um que permitia que um certo dado fosse direcionado a uma saída específica nesse estado. Ou seja, foram feitos um decodificador 2x4 e um demux 1x4, que direciona dados de 4 bits, em vez de apenas 1.

Com esses circuitos prontos, foi então criado um registrador de entrada e saída paralela de 4 bits. Utilizando os circuitos citados anteriormente, ligamos a entrada de dado de 3 desses registradores às saídas d1, d2 e d3 do demux, permitindo que os valores do 1º valor, da operação e do 2º valor fossem armazenadas, respectivamente. Já para a entrada de enable desses registradores, ligamos as saídas s1, s2 e s3 do decod, assim permitindo que a informação seja salva apenas quando for o estado adequado.



Implementação em esquemático do circuito descrito acima.

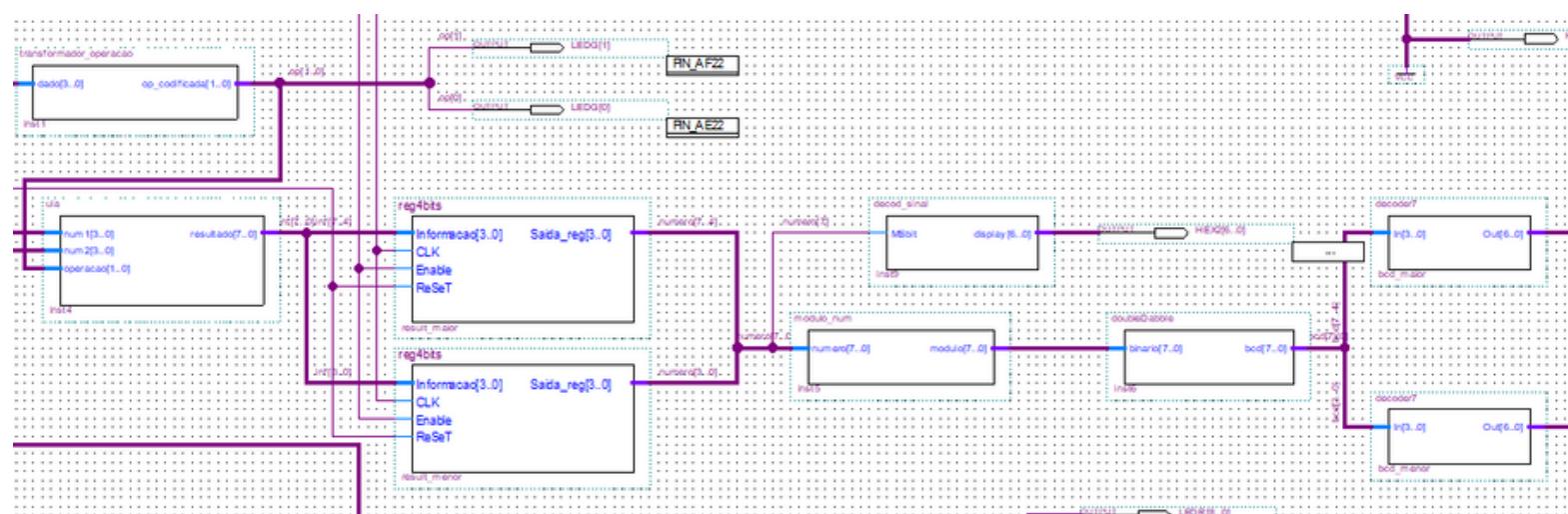
2. METODOLOGIA

2.3. Exibição do Resultado da Operação

Para que o resultado fosse corretamente exibido em 2 displays de 7 segmentos, com o sinal negativo em um terceiro caso necessário, foi necessário pesquisar algum algoritmo de conversão de binário para BCD (binary coded decimal), valor capaz de ser exibido corretamente em decimal nos displays. Tal algoritmo foi descoberto no vídeo "Visualizing Data with 7-Segment Displays", publicado no canal do YouTube "Sebastian Lague", no qual é mencionado o algoritmo *Double Dabble*. Esse algoritmo consiste em realizar o deslocamento de um número para a esquerda, e no fim desse deslocamento, verificar se nos 4 bits de cada casa decimal, é igual ou maior que 5, e, nesse caso, adicionar 3 a esse conjunto de 4 bits. Esse algoritmo funciona pois ele realiza a soma do 3 antes de realizar o deslocamento a esquerda, que dobra o número, e assim, o valor final da seção em BCD seria $2^*(\text{valor anterior} + 3)$, o que seria o +6 realizado para transformar valores binários em BCD, assim lidando com o overflow dessa operação antes dele aparecer na conta.

Com o entendimento do funcionamento desse algoritmo, o grupo foi pesquisar formas de implementá-lo em SystemVerilog, pois, devido à nossa inesperiência com essa linguagem, não sabíamos como poderíamos realizar essa implementação. Com sorte, foi encontrado a matéria "Binary to BCD and BCD to Binary" no site Real Digital, que sugeria uma forma de implementar o Double Dabble para um número binário de 14 bits em Verilog. Adaptando esse código, foi possível criar um circuito que realiza essa transformação de um número binário de 8 bits para seu valor correspondente em BCD.

Com esse problema resolvido, surgiu o problema da representação de números negativos. Como o software Quartus II utiliza complemento de 2 para armazenar números com sinal, seria necessário detectar esse número negativo para exibir o sinal '-', e também transformar esse número negativo em seu módulo, para que não ocorresse erros ao passar o número pelo circuito DoubleDabble. Para isso, captamos o MSbit do resultado e passamos em um decodificador que, quando esse valor era 1, a saída era os valores do display de 7 segmentos que ligava apenas o segmento central, caso contrário, nenhum segmento será ligado. Além disso, para obtermos o módulo do número, caso o MSbit seja 1, ocorre há inversão de todos os bits do número e soma-se 1 a ele.



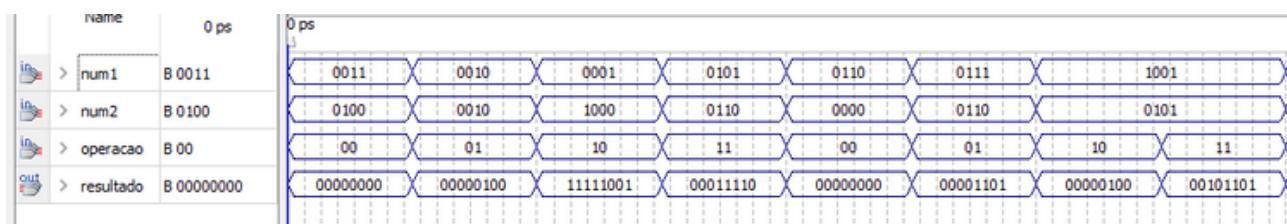
Implementação em esquemático do circuito de exibição do resultado determinado na ULA.

3. RESULTADOS OBTIDOS

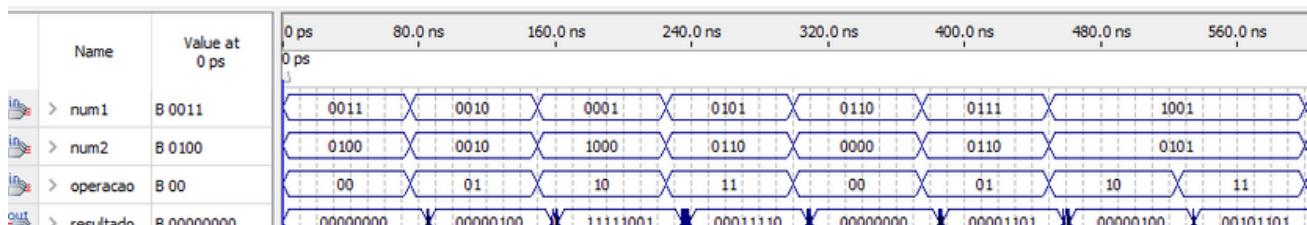
Como resultado final, conseguimos implementar a calculadora digital com sucesso, realizando as operações de soma, subtração e multiplicação de decimais de 1 bit. Com isso, pudemos aprofundar de forma prática os conceitos de hardware utilizando a placa FPGA DE2, com os displays de 7 segmentos e LEDs pudemos observar os resultados, tanto em decimal quanto em binário, implementados através do Quartus II.

Além disso, o trabalho também evidencia a necessidade de se seguir uma sequência correta na entrada de dados, para o funcionamento adequado da calculadora. A utilização do botão KEY[0] como reset do sistema e do LEDG[8] para indicar que o sistema está pronto para iniciar uma nova operação, auxiliando o uso e evitando possíveis erros de entrada.

Até o momento da escrita deste relatório, a transição da quarta para a primeira fase da máquina de estados, a qual controla o recebimento do número resultante da operação na ULA do circuito, não está funcionando conforme o planejado. Haja visto que, idealmente, somente ocorreria tal transição caso o sinal enviado fosse 1011, tecla "D"(=). Entretanto, tal mudança está ocorrendo com o recebimento de qualquer sinal, não apenas ao sinal 1011 como a chave para a transição entre os estados especificados anteriormente. É possível perceber que esse problema não ocorre na ULA, já que podemos observar seu comportamento normal nas simulações abaixo.



Simulação funcional da ULA, utilizando valores válidos aleatórios como as entradas num1 e num2.



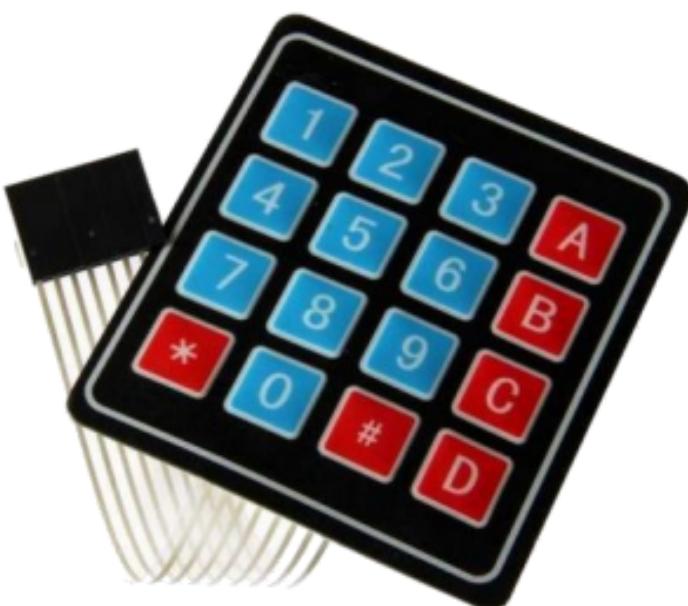
Simulação temporal da ULA, utilizando valores válidos aleatórios como as entradas num1 e num2.

Ademais, o grupo enfrentou algumas dificuldades no processo de desenvolvimento do trabalho. O principal deles foi a utilização de alguns kits DE2 com funcionamentos indesejados, os quais atrapalharam na sintetização e avaliação do circuito, uma vez que alguns problemas eram dos kits e não do circuito em si. Em relação a possíveis limitações que poderiam ocorrer, não foram encontradas pelo grupo quaisquer tipos de limitações ao longo do desenvolvimento do projeto.

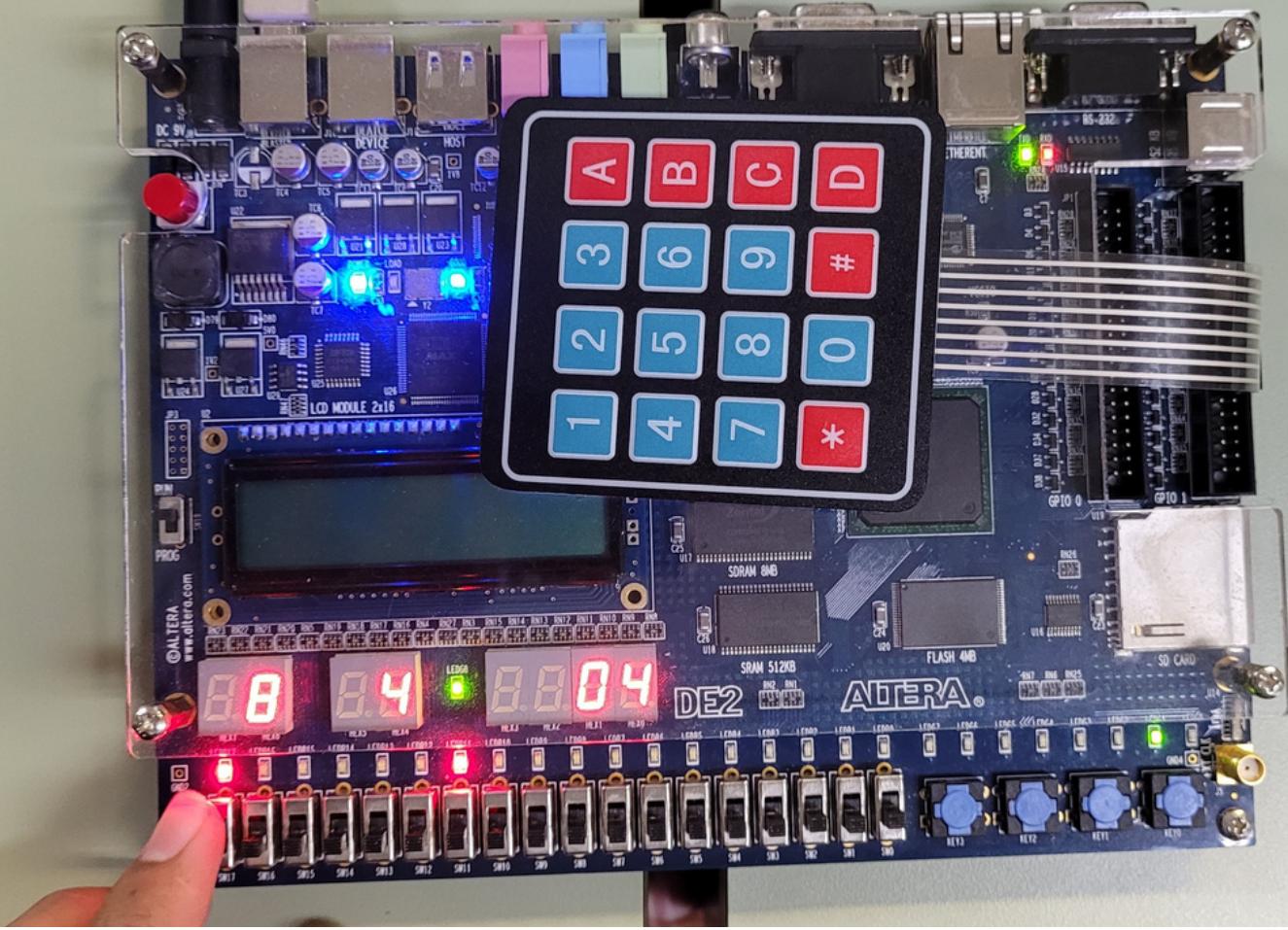
A versão final do circuito até o momento da escrita deste relatório pode ter seu funcionamento visto na placa FPGA DE2 no vídeo a seguir, o qual evidencia diversos casos possíveis no uso da calculadora, incluindo as três operações e resultados negativos. O seguinte vídeo também mostra como as LEDs e os displays estão em funcionamento durante e após as operações.

<https://youtu.be/gwSPA-8shKI>

Por fim, a implementação da calculadora possibilitou uma melhor compreensão dos princípios teóricos estudados ao longo do curso, ao mesmo tempo em que concedeu uma vivência prática e enriquecedora do manejo do kit FPGA. De modo geral, o projeto obteve êxito ao produzir uma calculadora operacional que atendeu às exigências estipuladas.



Teclado Matricial 4x4



CONCLUSÃO

Dessa forma, o projeto da calculadora resultou em um sistema funcional que atendeu às especificações propostas e atingiu os objetivos esperados. Nessa ótica, o teclado matricial foi utilizado com êxito para a entrada de dados, com uma exceção, e o display de 7 segmentos do kit FPGA DE2 mostrou os resultados corretamente, demonstrando a confecção e sintetização corretas do circuito.

Além disso, o grupo conseguiu trabalhar com o software Quartus II e com a linguagem SystemVerilog proveitosamente, de forma a adquirir conhecimentos e prática com ambos. Por fim, foram trabalhados diversos conteúdos vistos no semestre em Laboratório de Circuitos Lógicos, como máquina de estados e memória ROM, e reforçados por meio do desenvolvimento deste trabalho, garantindo uma aplicação prática para as teorias e tornando o processo bem mais rentável.

BIBLIOGRAFIA

- VISUALIZING Data with 7-Segment Displays. Publicado por Sebastian Lague. 2022. 1 vídeo (34 min). Disponível em: <<https://youtu.be/hEDQpqhY2MA>>. Acesso em 16 de fevereiro de 2023.
- BINARY to BCD and BCD to Binary. Real Digital. c2023. Disponível em: <<https://www.realdigital.org/doc/6dae6583570fd816d1d675b93578203d>>. Acesso em 16 de fevereiro de 2023.
- Pedroni, V., Eletrônica Digital Moderna e VHDL, Campus, 2010
- Tocci, R. J. e Widmer, N. S, Sistemas digitais: princípios e aplicações, LTC, 2010
- Floyd, T., Sistemas digitais fundamentos e aplicações, Bookman, 2011
- Harris, D. M. e Harrys S. L. Digital design and computer architecture, Morgan Kaufmann, 2017

