

BOMBA RELÓGIO

Projeto Final

GRUPO D3:

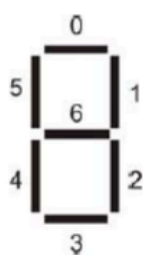
Ana Caroline Evangelista da Silva
João Vitor Ferreira Reis
Ycaro Rodrigo Rocha



UnB | CIC

RESUMO

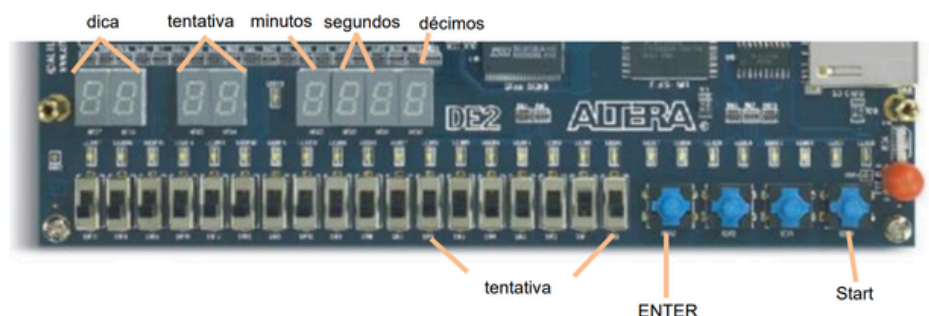
Este relatório apresenta o desenvolvimento de uma bomba relógio, um projeto elaborado por Ana Caroline Evangelista da Silva, João Vitor Ferreira Reis e Ycaro Rodrigo Rocha Santana, discentes da Universidade de Brasília na disciplina "Laboratório de Circuitos Lógicos" (LCL), sob a supervisão do professor Marcelo Grandi Mandelli. O projeto foi modelado e simulado utilizando o software Quartus II v.13.0, que permite a descrição, simulação e configuração de circuitos digitais. Além disso, a implementação foi realizada em uma placa DE2, baseada em tecnologia FPGA.



displays de 7 segmentos

ABSTRACT

This report presents the development of a time bomb project, designed by Ana Caroline Evangelista da Silva, João Vitor Ferreira Reis, and Ycaro Rodrigo Rocha Santana, students at the University of Brasília in the "Logic Circuits Laboratory" (LCL) course, under the supervision of Professor Marcelo Grandi Mandelli. The project was modeled and simulated using Quartus II v.13.0, a software that enables the description, simulation, and configuration of digital circuits. Additionally, the implementation was carried out on an DE2 board based on FPGA technology.



placa DE2 com mecanismos usados

INTRODUÇÃO

Desde os tempos mais remotos, o ser humano demonstrou a necessidade de desenvolver ferramentas para proteger seus bens. Uma das formas mais eficazes de segurança é o uso de cofres, que garantem acesso restrito a objetos de valor. Para aumentar a proteção, muitos desses sistemas utilizam mecanismos que exigem a inserção de uma senha correta para liberar o acesso.

Neste projeto, foi desenvolvido um sistema de segurança para um cofre baseado em uma senha escolhida de 7 bits, implementando conceitos de circuitos lógicos digitais. O sistema conta com um comparador de palavras implementado para verificar se a senha inserida pelo usuário corresponde à senha gerada.

Após a primeira tentativa de inserção da senha, um cronômetro regressivo de 5 minutos é ativado, determinando o tempo disponível para a autenticação. Durante esse período, o sistema permite a tentativa ilimitada de tentativas sem penalização. Caso a senha correta seja inserida dentro do tempo estipulado, o temporizador é interrompido e a abertura do cofre é autorizada. No entanto, se o tempo expirar sem a inserção da sequência correta, um mecanismo de segurança é acionado, ativando um sistema de defesa, a bomba é ativada.

A implementação do sistema envolve a utilização de circuitos combinacionais e sequenciais para processar as entradas, controlar os estados do sistema e gerenciar a contagem regressiva. A integração desses componentes garante o funcionamento adequado do sistema, possibilitando a autenticação segura e a execução das ações conforme o tempo disponível.

OBJETIVO E PROCEDIMENTO

A implementação deste sistema de segurança tem como propósito aplicar os conhecimentos adquiridos ao longo da disciplina de Circuitos Lógicos no desenvolvimento de um sistema digital baseado em senha. O projeto busca explorar a utilização de circuitos combinacionais e sequenciais garantir a proteção de um cofre, permitindo sua abertura apenas mediante a inserção da senha correta dentro de um tempo determinado.

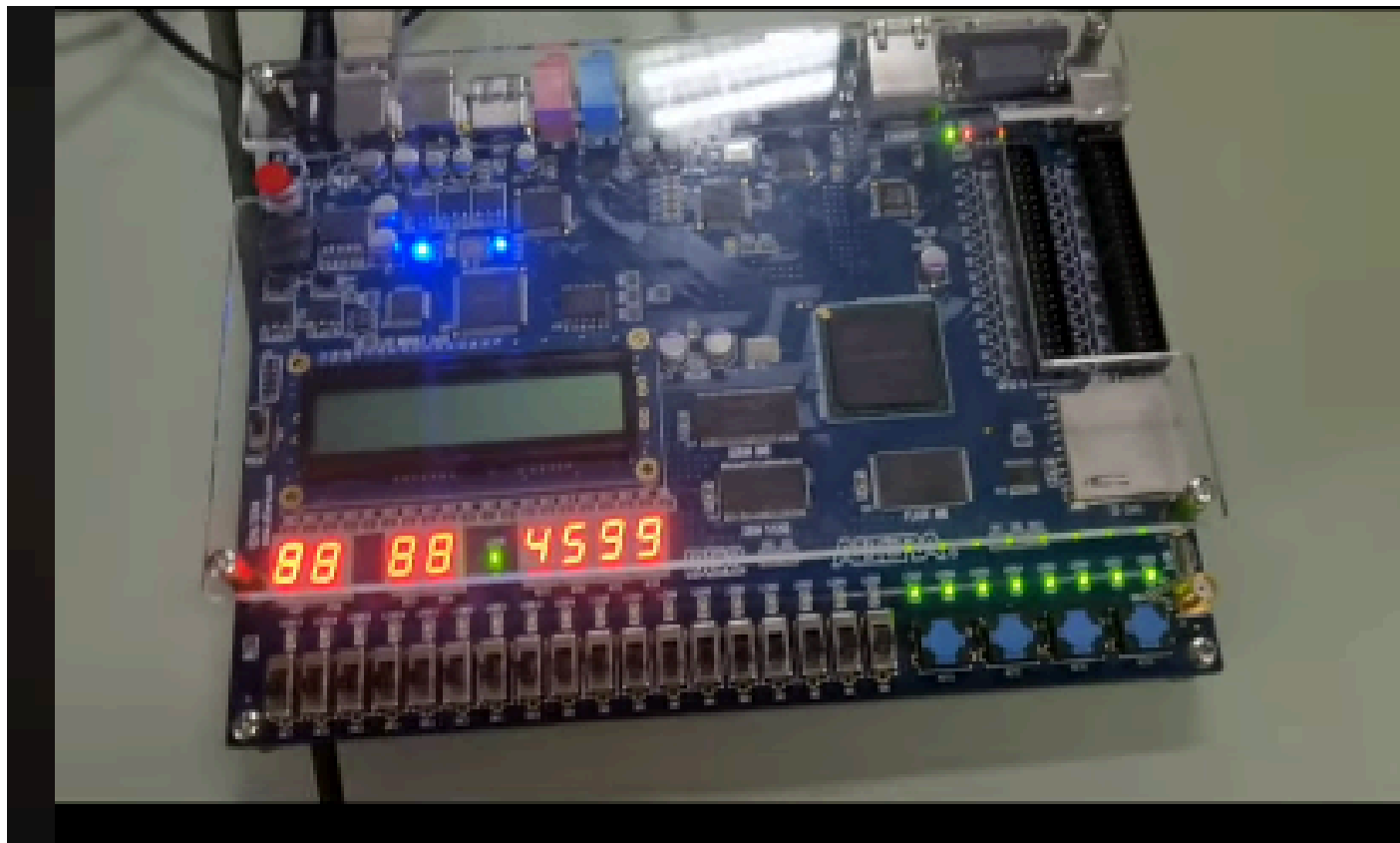
Além de servir como um mecanismo de proteção, o sistema implementado incorpora um elemento de contagem regressiva, transformando o cofre em uma bomba relógio fictícia. Caso a senha correta não seja inserida dentro do prazo estabelecido, um efeito visual será acionado nos displays e LEDs, simulando uma explosão. Essa abordagem adiciona um fator de urgência e desafio ao sistema para o desbloqueio mais rápido.

Para a implementação do sistema de segurança do cofre, foram desenvolvidos três módulos principais: um comparador, que verifica a correspondência entre a senha inserida e a gerada; um cronômetro regressivo, que controla o tempo disponível para desbloqueio; e um gerador de senha que permite a escolha do usuário com antecedência.

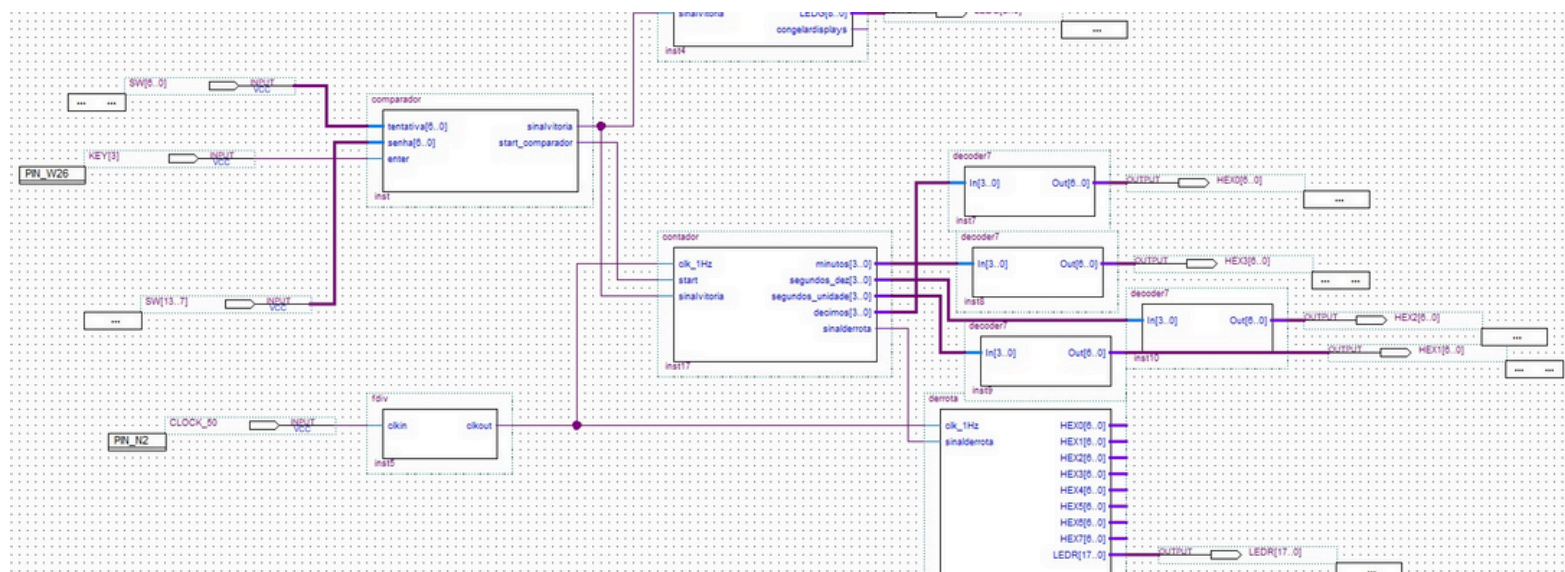
O funcionamento do sistema ocorre da seguinte forma:

- **Início:** Ao pressionar Start (KEY[0]), o sistema aguarda a inserção de uma senha oculta e inicia um cronômetro regressivo de 4 minutos e 59 segundos, exibido nos displays HEX0-HEX3.
- **Tentativa de desbloqueio:** O jogador insere uma senha utilizando as chaves SW[6:0] e confirma pressionando ENTER (KEY[3]), exibindo a tentativa no display HEX4.
- **Interação e encerramento:**
 - Se a senha estiver correta, o cronômetro congela e os LEDGs se acendem, indicando que a bomba foi desativada e o cofre aberto.
 - O jogo continua até que o jogador acerte a senha ou o tempo se esgote.
 - Se o tempo acabar antes da senha correta ser inserida, a bomba é ativada, acionando efeitos visuais nos LEDRs e displays HEX para simular uma explosão.

RESULTADO

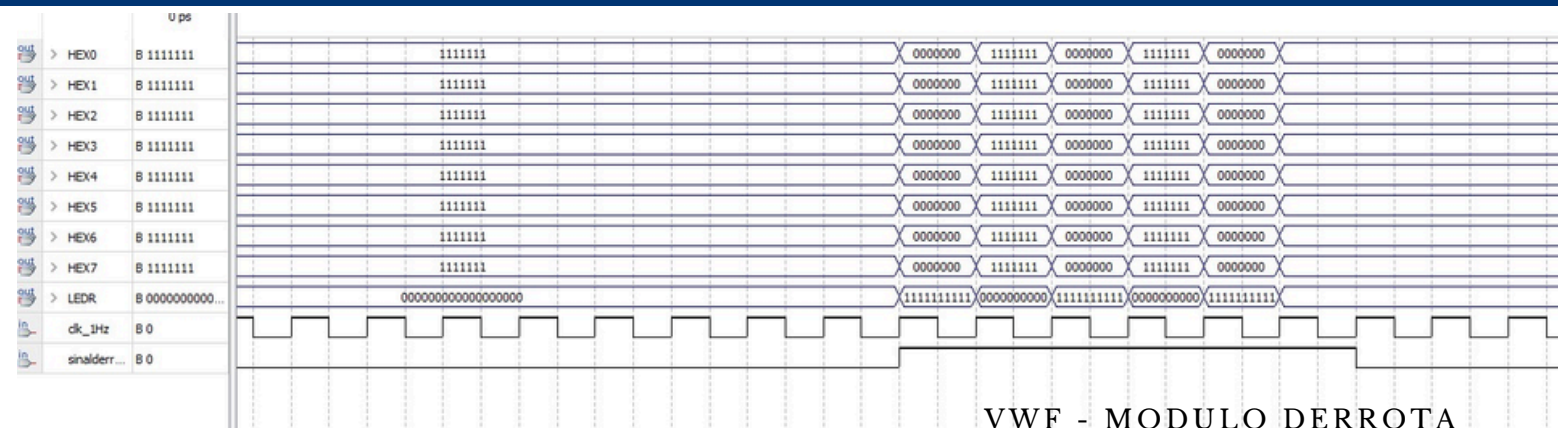


PLACA FUNCIONANDO COM SENHA DE 7 BITS, OS QUAIS, NESSE EXEMPLO, SÃO: 0000000. OS LEDS VERDES ACESOS IMPLICAM EM SENHA CORRETA, COMO SE OBSERVA NAS PINAGENS A DIREITA QUE COINCIDEM COM OS DA ESQUERDA.



ESQUEMÁTICO PRINCIPAL

RESULTADO



VWF - MODULO DERROTA

```

module derrota (
    input wire clk_1Hz,           // Clock de 1Hz
    input wire sinalderrota,      // Entrada que ativa o efeito de derrota (1 = liga)
    output reg [6:0] HEX0,        // Display 0
    output reg [6:0] HEX1,        // Display 1
    output reg [6:0] HEX2,        // Display 2
    output reg [6:0] HEX3,        // Display 3
    output reg [6:0] HEX4,        // Display 4
    output reg [6:0] HEX5,        // Display 5
    output reg [6:0] HEX6,        // Display 6
    output reg [6:0] HEX7,        // Display 7
    output reg [17:0] LEDR        // LEDs vermelhos (0 a 17)
);

    reg toggle; // Registrador para controlar o estado do piscar

    // Controle do toggle (inverte o estado a cada pulso de clock)
    always @(posedge clk_1Hz) begin
        if (sinalderrota) begin
            toggle <= ~toggle; // Inverte o estado se "sinalderrota" for 1
        end else begin
            toggle <= 1'b0;     // Mantém desligado se "sinalderrota" for 0
        end
    end

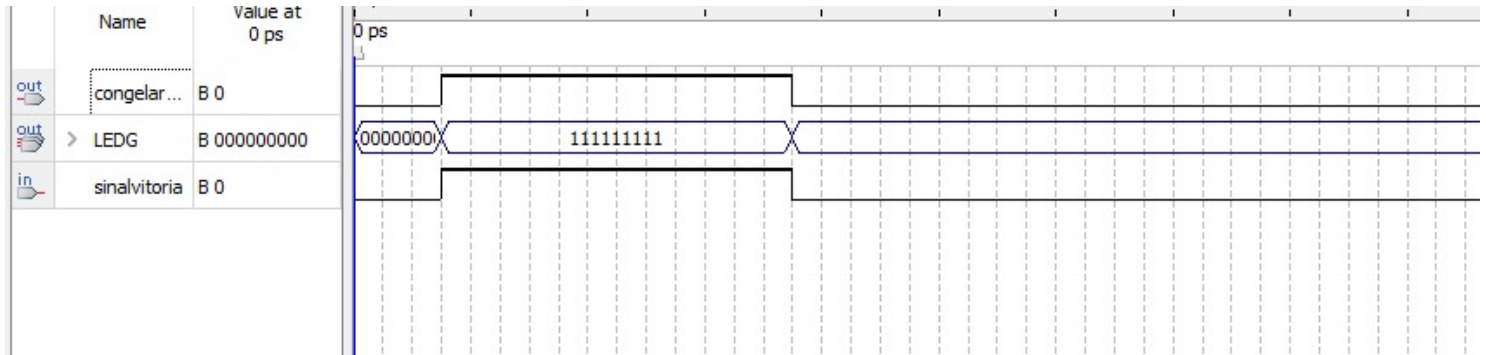
    // Lógica para os displays e LEDs
    always @(*) begin
        if (sinalderrota && toggle) begin
            // Todos os displays mostram "8" (todos os segmentos acesos)
            HEX0 = 7'b0000000; // Padrão para o dígito 8
            HEX1 = 7'b0000000;
            HEX2 = 7'b0000000;
            HEX3 = 7'b0000000;
            HEX4 = 7'b0000000;
            HEX5 = 7'b0000000;
            HEX6 = 7'b0000000;
            HEX7 = 7'b0000000;

            // Todos os LEDs vermelhos acesos (0 a 17)
            LEDR = 18'b111111111111111111;
        end else begin
            // Desliga todos os displays e LEDs
            HEX0 = 7'b1111111; // Todos os segmentos apagados
            HEX1 = 7'b1111111;
            HEX2 = 7'b1111111;
            HEX3 = 7'b1111111;
            HEX4 = 7'b1111111;
            HEX5 = 7'b1111111;
            HEX6 = 7'b1111111;
            HEX7 = 7'b1111111;

            LEDR = 18'b000000000000000000; // LEDs apagados
        end
    end
end
    
```

VERILOG MODULO DERROTA

RESULTADO

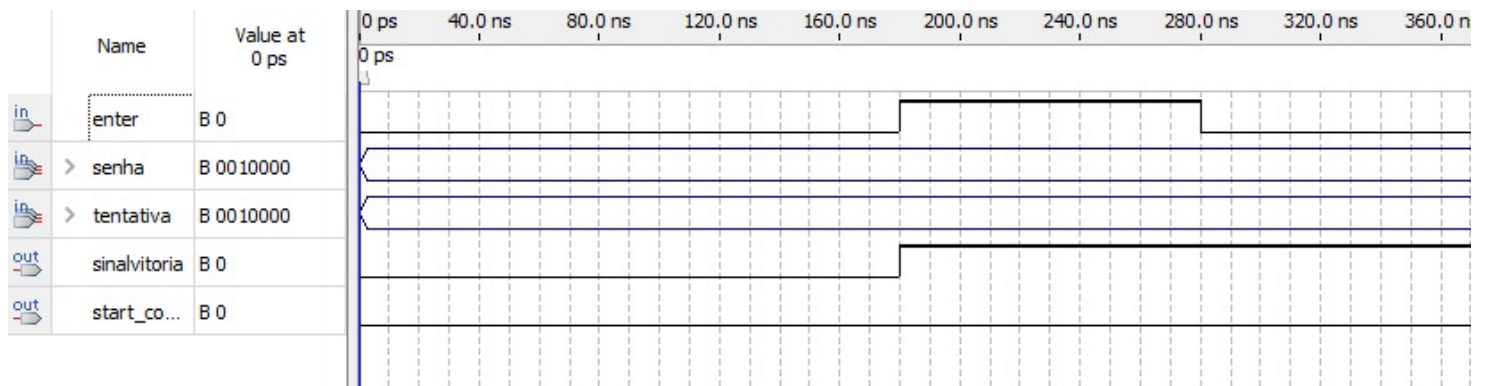


VWF - MODULO VITORIA

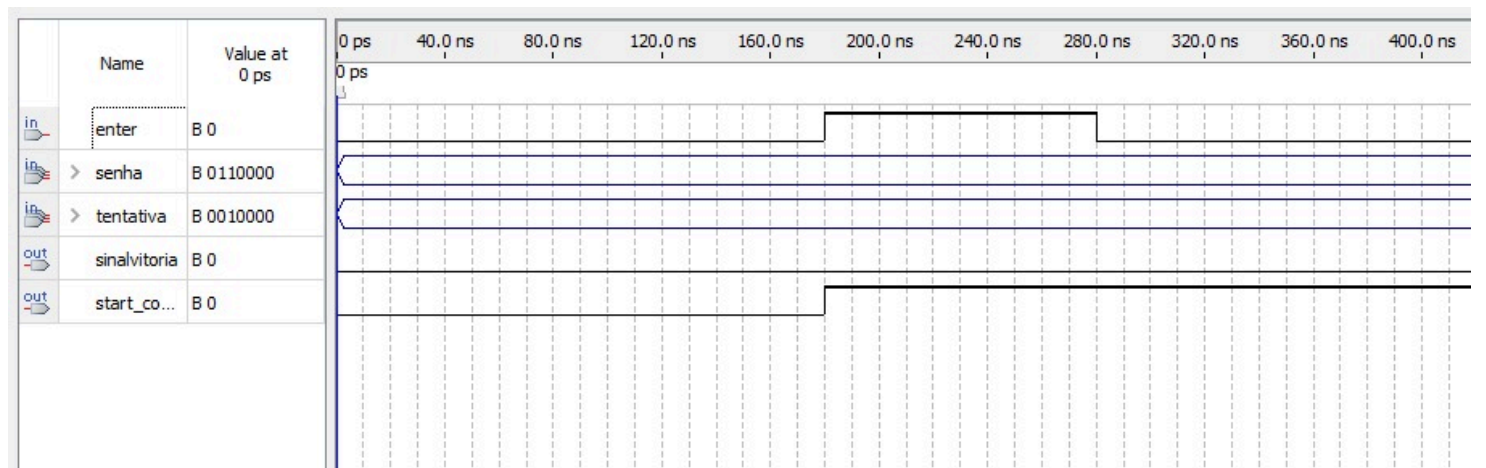
```
1  module vitoria (  
2      input wire sinalvitoria,  
3      output reg [8:0] LEDG,  
4      output reg congelardisplays  
5  );  
6      always @(*) begin  
7          if (sinalvitoria) begin  
8              LEDG <= 9'b111111111;  
9              congelardisplays <= 1;  
10         end else begin  
11             LEDG <= 9'b000000000;  
12             congelardisplays <= 0;  
13         end  
14     end  
15 endmodule
```

VERILOG MODULO VITORIA

RESULTADO



VWF MODULO COMPARADOR CASO CERTO



VWF MODULO COMPARADOR CASO ERRADO

```

1  module comparador (
2      input wire [6:0] tentativa,
3      input wire [6:0] senha,
4      input wire enter,
5      output reg sinalvitoria,
6      output reg start_comparador
7  );
8      always @(posedge enter) begin
9          if (tentativa == senha) begin
10             sinalvitoria <= 1;
11             start_comparador <= 0;
12          end else begin
13             sinalvitoria <= 0;
14             start_comparador <= 1;
15          end
16      end
17  endmodule

```

VERILOG MODULO COMPARADOR

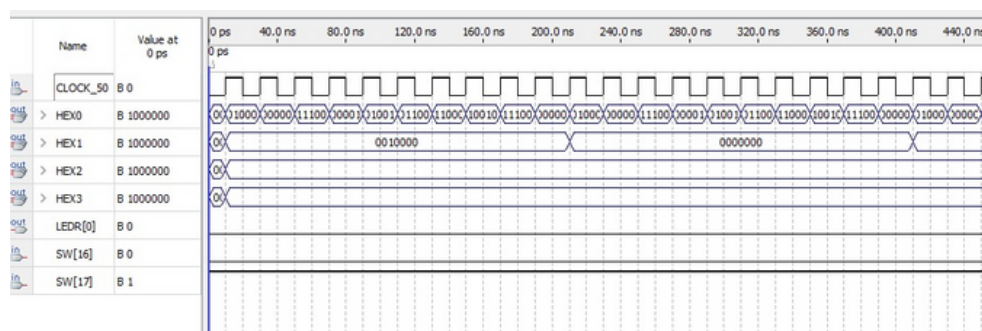
RESULTADO

```

1 module contador (
2     input wire clk_1Hz,
3     input wire start,
4     input wire sinalvitoria,
5     output reg [3:0] minutos,
6     output reg [3:0] segundos_dez,
7     output reg [3:0] segundos_unidade,
8     output reg [3:0] decimos,
9     output reg sinalderrota
10 );
11
12 reg [3:0] contador_minutos;
13 reg [3:0] contador_segundos_dez;
14 reg [3:0] contador_segundos_unidade;
15 reg [3:0] contador_decimos;
16 reg rodando;
17
18 always @(negedge start or posedge clk_1Hz) begin
19     if (!start) begin
20         contador_minutos <= 4;
21         contador_segundos_dez <= 5;
22         contador_segundos_unidade <= 9;
23         contador_decimos <= 9;
24         rodando <= 1;
25         sinalderrota <= 0;
26     end else if (rodando && !sinalvitoria) begin
27         if (contador_decimos > 0) begin
28             contador_decimos <= contador_decimos - 1;
29         end else begin
30             contador_decimos <= 9;
31             if (contador_segundos_unidade > 0) begin
32                 contador_segundos_unidade <= contador_segundos_unidade - 1;
33             end else begin
34                 contador_segundos_unidade <= 9;
35                 if (contador_segundos_dez > 0) begin
36                     contador_segundos_dez <= contador_segundos_dez - 1;
37                 end else begin
38                     contador_segundos_dez <= 5;
39                     if (contador_minutos > 0) begin
40                         contador_minutos <= contador_minutos - 1;
41                     end else begin
42                         // Tempo esgotado: dispara derrota
43                         rodando <= 0;
44                         sinalderrota <= 1;
45                     end
46                 end
47             end
48         end
49     end
50 end
51
52 always @(posedge clk_1Hz) begin
53     minutos <= contador_minutos;
54     segundos_dez <= contador_segundos_dez;
55     segundos_unidade <= contador_segundos_unidade;
56     decimos <= contador_decimos;
57 end
58 endmodule

```

VERILOG MODULO CONTADOR



VWF MODULO CONTADOR

DIFICULDADES

O desenvolvimento da bomba relógio apresentou desafios, especialmente na hora de ligar todos os dispositivos em conjunto, visto que separadamente eles funcionavam perfeitamente, mas quando integrados não. Foi necessário uma revisão detalhada dos códigos e auxílio dos monitores para o êxito desse trabalho.

CONCLUSÃO

O desenvolvimento deste projeto permitiu a aplicação prática dos conhecimentos adquiridos na disciplina de Circuitos Lógicos, consolidando conceitos como circuitos combinacionais e sequenciais. A construção do sistema exigiu um aprofundamento técnico significativo, além da necessidade de superar desafios relacionados à implementação e sincronização dos módulos.

Durante a realização do experimento, o trabalho em equipe foi essencial para a resolução de problemas e otimização do sistema. A troca de conhecimentos entre os integrantes e o suporte oferecido pelo professor Marcelo Grandi Mandelli foram fundamentais para o êxito do projeto. Além disso, o grupo expressa profunda gratidão aos monitores, que ofereceram suporte técnico e disponibilizaram equipamentos, contribuindo diretamente para a conclusão do trabalho.

Por fim, além do aprendizado técnico, o projeto reforçou a importância da colaboração, organização e comprometimento para a execução bem-sucedida de atividades acadêmicas desafiadoras.

Link da bomba funcionando:



[bomba em execução](#)

REFERÊNCIAS

Referências bibliográficas

- Pedroni, V., Eletrônica Digital Moderna e VHDL, Campus, 2010.
- Aprender3, slides Circuitos Lógicos, 2024.
- SIVARAMAN, K.; MONGIA, S. SystemVerilog for Hardware Design. 1. ed. Boston: Addison-Wesley, 2006.
- PILATO, R. A. SystemVerilog: A Guide to Learning. 1. ed. San Francisco: Springer, 2011.
- Sistemas Digitais: Princípios e Aplicações. Pearson, 11ª edição, 2010.
- BROWN, Stephen; VRANESIC, Zvonko. Fundamentals of Digital Logic with Verilog Design. 3rd ed. New York: McGraw-Hill, 2013.
- MAXFIELD, Clive. The Design Warrior's Guide to FPGAs: Devices, Tools, and Flows. Newnes, 2004.