ROTEIRO LABORATÓRIO 01 – MODELAGEM DE SISTEMAS DIGITAIS UTILIZANDO LINGUAGEM DE DESCRIÇÃO DE HARDWARE E FPGA

Observações:

a) Integrantes por equipe: 05 (máximo)

b) A apresentação dos resultados será realizada por cada equipe de modo presencial no Laboratório A36. Deverão ser apresentados os seguintes itens:

- Diagrama em Blocos Representativo da solução;
- Relatório de Síntese (Número de elementos lógicos, quantidade de pinos utilizados);
- Demonstração no Kit FPGA.

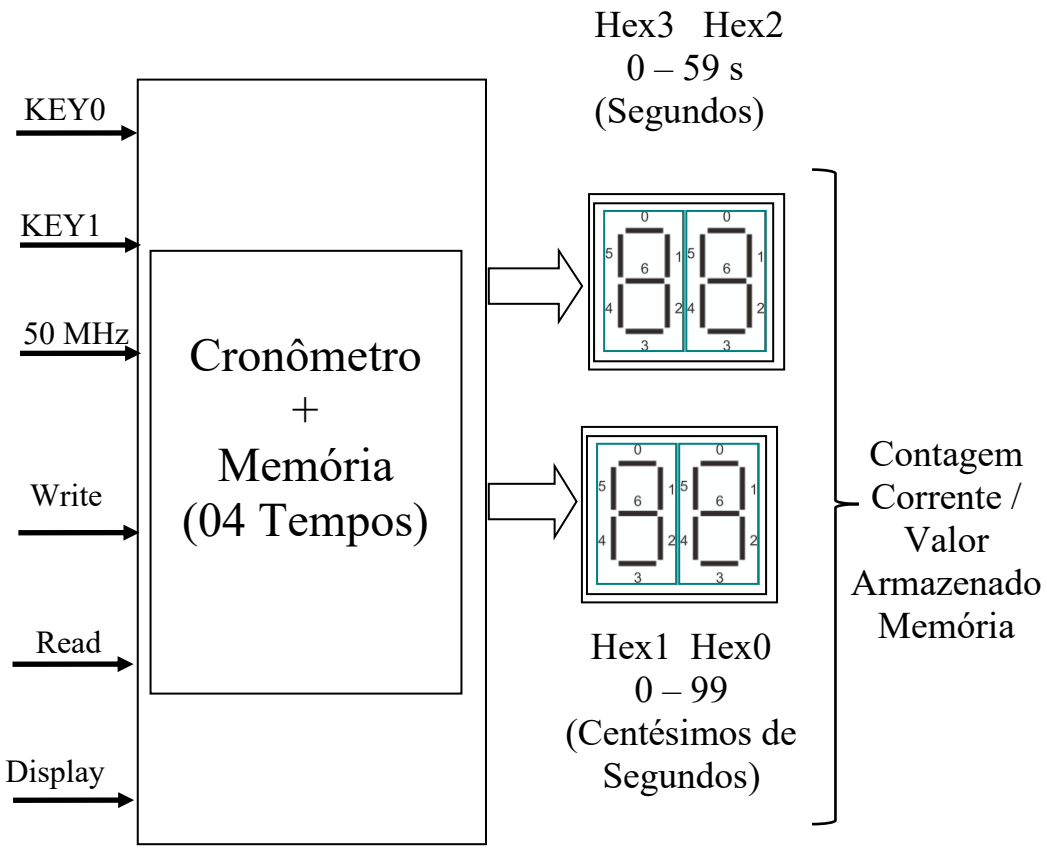
d) Data limite da apresentação: 03/10/2025;

e) Não haverá aula nos dias: 15/09, 17/09, 22/09, 24/09, 29/09, 01/10. O Professor estará disponível nestes dias e nos horários da aula para esclarecimentos de dúvidas, bem como para disponibilizar os kits de FPGA para testes do projeto de cada equipe.

f) Avaliação 01: 06/10/2025, Laboratório A31.

Projeto 01 (5,0)

Projete o sistema da Figura abaixo utilizando a linguagem de descrição de hardware (Verilog) de acordo com as funcionalidades descritas a seguir.



DESCRIÇÃO DAS FUNCIONALIDADES DO SISTEMA

KEY0: Inicialização do sistema em nível lógico zero (0). Quando esta chave for pressionada, a contagem do cronômetro deverá ser igual a 00:00.

KEY1:

1: Inicia e/ou continua a contagem crescente do cronômetro e exibe o valor no display;

0: suspende a contagem do cronômetro;

Write:

0: Efetua uma operação de escrita em uma memória interna de 04 posições. Cada vez que a tecla for pressionada, o tempo corrente do cronômetro deverá ser armazenado sequencialmente na posição 0, 01, 02 e 03, retornando para a posição 0 e assim sucessivamente, sobrescrevendo o valor anteriormente armazenado.

1: mantém o contador de endereços de escrita com o valor anterior;

Read:

0: Efetua uma operação de leitura em uma memória interna de 04 posições. Cada vez que a tecla for pressionada, o tempo do cronômetro armazenado na memória deverá ser exibido no display (HEX 4, 5, 6 e 7), começando da posição 00, 01, 02 e 03, retornando para a posição 0 e assim sucessivamente.

1: mantém o contador de endereços de leitura com o valor anterior;

Display:

1: Exibe o valor da contagem do cronômetro

0: Exibe o valor armazenado na memória

Família: Cyclone II

Dispositivo: EP2C20F484C7

Sugestões:

1. Fazer um divisor de frequência para obter um clock igual a 100 Hz ou próximo deste valor. Porém para simulação, faça o teste utilizando o clock de 50 MHz e um divisor por 2 para que a simulação possa ser executada mais rapidamente. No momento da programação na placa, insira o divisor "original" (50 MHz/fator = 100 Hz);
2. Utilizar um detector de transição para gerar um pulso de duração igual ao período do clock do sistema para controlar a quantidade de acionamentos das chaves de escrita e leitura na memória;
3. Utilizar um contador de endereços para a escrita e outro para leitura com a contagem controlada pelos sinais de escrita e leitura externos;

Observações para projetos com chaves push-botton para acionamento das operações:

Utilize um detector de transição para efetuar a detecção da mudança de estado da chave conforme o código a seguir. Desta maneira, cada vez que a chave for pressionada, será gerado um pulso na saída do detector de transição da largura do clock. Este pulso poderá ser utilizado para acionar a FSM utilizada para controlar todo o sistema:

```
module detector_trans (input signal_in, clock, clear, output reg edge_detected);

reg signal_in_prev;

//----- Detector de transição -----------

always @(posedge clock or negedge clear)

        if(!clear) begin

                edge_detected <= 1'b0;
                signal_in_prev <= 1'b0;

        end

        else if( signal_in_prev == 1'b0 && signal_in == 1'b1) begin

                edge_detected <= 1'b1;
                signal_in_prev <= signal_in;
        end

        else begin

                edge_detected <= 1'b0;
                signal_in_prev <= signal_in;
        end

endmodule
```