EXERCÍCIO PROGRAMA - EXPERIÊNCIA NO LABORATÓRIO DIGITAL

Autores: Miguel Shiniti Aguena (miguel.aguena@usp.br), Javier Ulises Solis Lastra (jsolis@usp.br), Bruno de Carvalho Albertini (balbertini@usp.br)

Este é o EP cujas soluções você deve desenvolver e entregar no dia 10/02/2026, das 10h às 12h.

Siga as especificações de cada problema corretamente. Desenvolva suas soluções de acordo com os *toplevels* fornecidos. Utilize as entidades fornecidas junto a este enunciado (via e-mail). Utilize o software Quartus Prime Lite para desenvolver as soluções, e entregue os arquivos .QAR (Quartus Prime Archive File) de cada solução na data e horário especificados. Em caso de dúvidas, contate os autores deste EP.

Em todos os problemas, é permitido utilizar blocos **`always`**, bem como variáveis inteiras e operações aritméticas, sem a necessidade de se fazer somadores completos.

SUMÁRIO

PROBLEMA 1 - CONTAGEM REGRESSIVA

Contadores são dispositivos muito importantes em Sistemas Digitais. Eles permitem contar o tempo, quantidade de entradas de um usuário, quantidade de ocorrências de um evento, entre outros.

Para este problema, você irá desenvolver um contador regressivo de segundos. Trata-se de um dispositivo que, dada uma quantia de segundos, retorna um sinal em alto quando este período tiver passado.

Diversos dispositivos computacionais possuem um sinal oscilatório chamado *clock*, usado como um metrônomo para sincronizar eventos. O modelo de placa FPGA utilizada no Laboratório Digital possui um gerador de *clock* de 50 MHz - 50 milhões de oscilações por segundo. Em outras palavras, um segundo pode ser medido a cada 50 milhões de oscilações.

Um contador de um segundo (**`one_second_counter`**) específico para 50 MHz foi fornecido junto a este enunciado (ver SOBRE AS ENTIDADES FORNECIDAS). Utilize ele para implementar seu contador regressivo.

O seu *toplevel* é o seguinte:

```
module regressive_counter (
    input clock,
    input reset,
    input [3:0] seconds_period,
    input start,
    output [6:0] hex5_seconds_left,
    output reg ledr9_finished
);

endmodule
```

`clock` é o *clock* de 50 MHz da placa FPGA DE0-CV, **`reset`** é uma entrada que reinicia o circuito para o estado inicial, **`seconds_period`** é uma entrada de 4 bits que define o período de tempo que o contador regressivo deve contar, **`start`** é a entrada que inicia a contagem, **`hex5_seconds_left`** é um sinal conectado ao display de 7 segmentos HEX5 da DE0-CV que mostra quantos segundos faltam, e **`ledr9_finished`** é um sinal conectado ao LEDR9 da DE0-CV que acende quando o tempo restante chegar ao fim.

Sempre que o circuito está no estado inicial (antes de **`start`** ser acionada), **`hex5_seconds_left`** deve exibir o período de tempo desejado

(**seconds_period**) no display. Quando o sinal **start** é acionado, **hex5_seconds_left** deve exibir os segundos que faltam na contagem regressiva. Quando a contagem for encerrada, **ledr9_finished** deve ser alto. O sinal **reset** deve levar o circuito ao estado inicial sempre que for acionado.

Caso **start** seja colocada em baixo no meio da contagem, esta deve ser interrompida. Perceba que se a contagem não for interrompida, ela será reiniciada ao chegar no fim, com o valor que estiver inserido em **seconds_period**.

PROBLEMA 2 - SEMÁFORO PARA VEÍCULOS

O prefeito de São Paulo contratou a sua empresa para desenvolver um sistema de semáforo a ser instalado na Avenida Paulista.

O semáforo veicular deve exibir aos motoristas um sinal VERDE durante 15 segundos, antes de exibir AMARELO durante 2 segundos, exibindo então VERMELHO por 10 segundos, voltando a exibir VERDE após este.

O seu *toplevel* é o seguinte:

```
module vehicular_semaphore (
      input clock,
      input reset,
      input start,
      output [6:0] hex5_seconds_left,
      output ledr9_green,
      output ledr8_yellow,
      output ledr7_red
);

endmodule
```

`clock` é o *clock* de 50 MHz da placa FPGA DE0-CV, **`reset`** é uma entrada que reinicia o circuito para o estado inicial, **`start`** é a entrada que inicia o semáforo, **`hex5_seconds_left`** é um sinal conectado ao display de 7 segmentos HEX5 da DE0-CV que mostra quantos segundos faltam até a próxima transição, e **`ledr9_green`**, **`ledr8_yellow`** e **`ledr7_red`** são sinais que representam as saídas VERDE, AMARELO e VERMELHO, respectivamente conectadas nos LEDR9, LEDR8 e LEDR7 da placa FPGA.

Perceba que você pode reutilizar ou adaptar sua solução para o problema anterior aqui.

No estado inicial, o semáforo não deve exibir qualquer comportamento, exceto **`ledr9_green`** em alto (o semáforo sempre começa em VERDE) e **`hex5_seconds_left`**, o qual deve mostrar o tempo restante até a próxima transição (15 segundos). Do momento em que **`start`** é acionada, o semáforo deve se comportar conforme descrito: **`ledr9_green`** deve permanecer alto por 15 segundos, e então **`ledr8_yellow`** deve permanecer alto por 2 segundos, e finalmente **`ledr7_red`** deve permanecer alto por 10 segundos, repetindo o comportamento a partir daí. Em nenhum momento deve haver mais de um sinal alto

entre **`ledr9_green`**, **`ledr8_yellow`** e **`ledr7_red`** ao mesmo tempo. O sinal **`reset`** deve levar o circuito ao estado inicial sempre que for acionado.

Caso **`start`** seja colocada em baixo em qualquer momento, o sistema de semáforo simplesmente tem suas rotinas interrompidas.

PROBLEMA 3 - SEMÁFORO COMPLETO

Além do sistema de semáforo para veículos, o prefeito de São Paulo também solicitou que sua empresa desenvolva um semáforo para pedestres atravessarem a Avenida Paulista, o qual deve funcionar junto ao primeiro.

No sistema de semáforo, deve ser instalado um botão SOLICITAR TRAVESSIA, o qual, ao ser acionado quando o semáforo para veículos exibe sinal VERDE aos motoristas, deve fazer este imediatamente exibir AMARELO, e enfim, VERMELHO.

Ademais, o semáforo para pedestres deve exibir um sinal ATRAVESSE somente quando o semáforo para veículos exibir VERMELHO. Nos últimos 3 segundos antes de o semáforo para veículos mudar de VERMELHO para VERDE, o semáforo para pedestres deve piscar rapidamente o sinal ATRAVESSE, indicando que o fechamento do sinal se aproxima. A frequência desta oscilação deve ser de 4 Hz (4 vezes por segundo).

O seu *toplevel* é o seguinte:

```
module full_semaphore (
      input clock,
      input reset,
      input start,
      input cross_request,
      output [6:0] hex5_seconds_left,
      output ledr9_green,
      output ledr8_yellow,
      output ledr7_red,
      output ledr0_cross
);

endmodule
```

`clock` é o *clock* de 50 MHz da placa FPGA DE0-CV, **`reset`** é uma entrada que reinicia o circuito para o estado inicial, **`start`** é a entrada que inicia o semáforo, **`cross_request`** é a entrada que representa o sinal SOLICITAR TRAVESSIA, **`hex5_seconds_left`** é um sinal conectado ao display de 7 segmentos HEX5 da DE0-CV que mostra quantos segundos faltam até a próxima transição, **`ledr9_green`**, **`ledr8_yellow`** e **`ledr7_red`** são sinais que representam as saídas VERDE, AMARELO e VERMELHO, respectivamente conectadas nos LEDR9, LEDR8 e LEDR7 da placa FPGA, e **`ledr0_cross`** é o sinal que representa a saída ATRAVESSAR, conectado ao LEDR0 da DE0-CV.

Incremente sua solução para o problema anterior com estas adições.

No estado inicial, o semáforo não deve exibir qualquer comportamento, exceto **`ledr9_green`** em alto (o semáforo sempre começa em VERDE) e **`hex5_seconds_left`**, o qual deve mostrar o tempo restante até a próxima transição (15 segundos). Do momento em que **`start`** é acionada, o semáforo deve se comportar conforme descrito: **`ledr9_green`** deve permanecer alto por 15 segundos, e então **`ledr8_yellow`** deve permanecer alto por 2 segundos, e finalmente **`ledr7_red`** deve permanecer alto por 10 segundos, repetindo o comportamento a partir daí. Em nenhum momento deve haver mais de um sinal alto entre **`ledr9_green`**, **`ledr8_yellow`** e **`ledr7_red`** ao mesmo tempo. O sinal **`reset`** deve levar o circuito ao estado inicial sempre que for acionado.

Caso **`start`** seja colocada em baixo em qualquer momento, o sistema de semáforo simplesmente tem suas rotinas interrompidas.

Se a entrada **`cross_request`** for acionada enquanto o semáforo exibe VERDE, ele deve imediatamente exibir AMARELO, ainda seguindo o comportamento anteriormente descrito. A entrada **`cross_request`** não deve causar nenhum efeito se o semáforo já estiver exibindo AMARELO ou VERMELHO.

A saída **`ledr0_cross`** deve permanecer alta sempre que o semáforo estiver exibindo VERMELHO (note: isso independe de **`cross_request`** ter sido acionada ou não). Nos últimos 3 segundos da exibição de VERMELHO, a saída **`ledr0_cross`** deve oscilar rapidamente a uma frequência de 4 Hz.

SOBRE AS ENTIDADES FORNECIDAS

Junto a este EP, foram fornecidas (via e-mail) duas entidades que podem ser utilizadas nas soluções. Elas são as seguintes:

```
module one_second_counter (
    input clock,
    input reset,
    input start,
    output finished
);
```

one_second_counter é um contador de um segundo. **clock** é o *clock* de 50 MHz da placa FPGA DE0-CV, **reset** é a entrada que reinicia o circuito para o estado inicial, e **start** é a entrada que permite a contagem. Uma vez que essa última for acionada, a saída **finished** fica em alto por um ciclo de *clock* a cada segundo que passa. Colocar **start** em baixo no meio da contagem apenas a interrompe no meio do processo.

```
module hexa7seg (
    input [3:0] hexa,
    output reg [6:0] sseg
);
```

hexa7seg é um multiplexador para displays de 7 segmentos. Ele recebe um número de 4 bits **hexa** e retorna na saída **sseg** tal número em um formato legível em tais displays, facilitando a observação de valores numéricos.

SOBRE A ENTREGA

No dia 10/02/2026, das 10h às 12h, você deve nos entregar os arquivos .QAR contendo as respectivas soluções de cada problema. Iremos avaliar o comportamento das suas soluções configurando placas FPGA DE0-CV com elas. Caso suas soluções não estejam funcionando nas placas, você pode continuar resolvendo problemas, bem como tirar dúvidas, neste período entre às 10h e 12h presencialmente no Laboratório Digital - estipulamos este tempo para que você possa resolver quaisquer empecilhos que surjam.