

# Curso de Introdução à Design de Hardware Reconfigurável em FPGA

Profs. Amanda Martinez e Zoé Magalhães

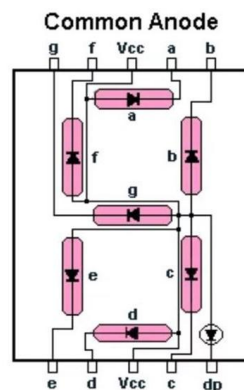
22/07/2023

## Lista de Exercícios 1

O objetivo desta lista de exercícios é conduzir os alunos através de uma série de desafios que, ao serem concluídos, resultarão no desenvolvimento de um projeto funcional: um cronômetro. Em todos os exercícios da lista fornecemos dicas para auxiliar no desenvolvimento, mas é importante salientar que existem outras opções de desenvolvimento, ou seja, não é mandatório seguir o fluxo de desenvolvimento que estamos sugerindo.

### Exercício 4 - Codificação do sinal exibido no display de 7 segmentos

O objetivo desse exercício é alterar o design feito no exercício 3 para que as contagens sejam disponibilizadas no formato para exibição em um display de 7 segmentos com ânodo comum, como o ilustrado na figura:



Considerando que cada pino do display está conectado em uma GPIO da FPGA, para exibir um algarismo no display, o *design* deve configurar para nível lógico alto apenas as GPIOs conectadas ao pino VCC e os catodos dos segmentos que devem ficar apagados. Os catodos dos segmentos que devem ser acesos precisam ser configurados em nível lógico baixo para aplicar uma diferença de potencial elétrico necessária para acender o LED do segmento.

O objetivo desse exercício é alterar as saídas que disponibilizam a contagem como um sinal do tipo inteiro para que disponibilizem como um sinal do tipo *std\_logic\_vector* de 7 bits, em que cada bit indica o estado de um catodo para que o algarismo correspondente ao valor da contagem seja exibido no display.

O controle do ânodo será abordado em um próximo exercício.

1. Faça as seguintes alterações no *design* do exercício 3.
  - a. Altere o tipo das portas de saída que disponibilizam os contadores de segundos e minutos para que cada uma disponibilize um sinal digital de 7 bits.
  - b. Para cada uma dessas saídas de 7 bits adicione um circuito que codifica o valor da saída para exibição do valor do respectivo contador, considerando que cada uma dessas saídas está conectada aos pinos a-g (catodos) de um display de 7 segmentos.
2. Verifique o RTL gerado
3. Atualize o testbench
4. Execute a simulação comportamental e observe se os sinais se comportam como esperado.
5. Adicione os sinais internos do DUT que registram o valor dos contadores de saída ao waveform e observe se a relação entre uma saída de 7 bits e seu respectivo contador está correta.

### DICA: Codificação da saída

Constantes podem ser declaradas na arquitetura com a definição dos padrões de 7 bits para exibição de cada algarismo. Para isso, inclua na parte declarativa da arquitetura (antes de begin) a declaração das seguintes constantes:

```
constant DISPLAY_7SEG_ZERO : std_logic_vector(6 downto 0) := "0000001";
constant DISPLAY_7SEG_ONE : std_logic_vector(6 downto 0) := "1001111";
constant DISPLAY_7SEG_TWO : std_logic_vector(6 downto 0) := "0010010";
constant DISPLAY_7SEG_THREE : std_logic_vector(6 downto 0) := "0000110";
constant DISPLAY_7SEG_FOUR : std_logic_vector(6 downto 0) := "1001100";
constant DISPLAY_7SEG_FIVE : std_logic_vector(6 downto 0) := "0100100";
constant DISPLAY_7SEG_SIX : std_logic_vector(6 downto 0) := "0100000";
constant DISPLAY_7SEG_SEVEN : std_logic_vector(6 downto 0) := "0001111";
constant DISPLAY_7SEG_EIGHT : std_logic_vector(6 downto 0) := "0000000";
constant DISPLAY_7SEG_NINE : std_logic_vector(6 downto 0) := "0000100";
constant DISPLAY_7SEG_INVALID : std_logic_vector(6 downto 0) := "1111110";
```

A codificação das portas de saída de 7 bits podem ser descritas com a diretiva “**with select**”. Essa diretiva descreve um circuito multiplexador, em que um sinal seleciona o valor que será atribuído a outro sinal.

O trecho de código a seguir mostra essa diretiva sendo utilizada atribuir a `seconds` um valor selecionado por `seconds_reg`. Por exemplo, quando `seconds_reg` é igual a 2, `seconds` é atribuído com o valor `DISPLAY_7SEG_TWO`.

```
with seconds_reg select seconds <=
    DISPLAY_7SEG_ZERO when 0,
    DISPLAY_7SEG_ONE when 1,
```

```
DISPLAY_7SEG_TWO when 2,  
DISPLAY_7SEG_THREE when 3,  
DISPLAY_7SEG_FOUR when 4,  
DISPLAY_7SEG_FIVE when 5,  
DISPLAY_7SEG_SIX when 6,  
DISPLAY_7SEG_SEVEN when 7,  
DISPLAY_7SEG_EIGHT when 8,  
DISPLAY_7SEG_NINE when 9,  
DISPLAY_7SEG_INVALID when others;
```

A diretiva deve contemplar todos os valores possíveis do sinal seletor para que não exista uma condição em que o hardware descrito tenha comportamento imprevisto. Por isso, uma boa prática é incluir a opção **when others**. Essa opção indica o valor padrão atribuído quando o sinal seletor não é igual a nenhuma das alternativas . Nesse exemplo, a opção **when others** corresponde a um valor de `seconds_reg` fora do intervalo fechado de 0 a 9. Nesse caso, o sinal `seconds_reg` é configurado com um valor que acende o display de forma que não exibe nenhum algarismo decimal.