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 3 - Criação de contadores de unidade e dezenas de segundos e unidades e dezenas de minutos.

Dicas:

1. Faça as seguintes alterações no circuito divisor de clock desenvolvido no Exercício 2:
    a. Adicione à arquitetura os sinais necessarios para as seguintes contagens:
        - unidade de segundos (*range* de 0 a 9)
        - dezenas segundos (*range* de 0 a 5)
        - unidade de minutos (contagem de 0 a 9)
        - dezenas de minutos (contagem de 0 a 5).
    b. Mantenha na arquitetura o processo do ***contador decrescente*** e crie um segundo processo que descreve o incremento desses contadores (do item a) na borda de subida de clock somente se as condições forem satisfeitas:
        - contador de unidade de segundos
            - ***contador decrescente*** igual a 0.

OBS: Um sinal de um processo pode ser consultado em outro processo, mas de forma alguma um sinal pode ser atualizado em dois processos distintos! Ou seja, podemos consultar neste segundo processo o sinal que armazena o valor do contador decrescente (atualizado no primeiro processo), mas não podemos atualizá-lo no segundo processo.

        - contador de dezena de segundos
            - ***contador decrescente*** for igual a 0.
            - **E** contador de unidade segundos for igual a 9
        - contador de unidade de minutos
            - ***contador decrescente*** for igual a 0.
            - **E** contador de unidade segundos for igual a 9
            - **E** contador de dezena de segundos for igual a 5

        - contador de dezenas de minutos:
            - ***contador decrescente*** for igual a 0.
            - **E** contador de unidade segundos for igual a 9
            - **E** contador de dezena de segundos for igual a 5
            - **E** contador de unidade de minutos for igual a 9
    c. Adicionar as portas de saída necessárias para disponibilizar o valor de cada um dos contadores de segundos e minutos.
    d. Remover a porta de saída *blink.*
2. Analisar o RTL
3. Atualizar o testbench para a nova interface.
4. Testar em simulação

DICA: Testes de condições em VHDL

Condições em VHDL podem ser descritas com a diretiva **"if else"**.

```
if condition_test then
            -- lógica realizada se condição for verdadeira
else
            -- lógica realizada se a condição for false
end if;
```

onde condition_test representa as condições que selecionam qual lógica será realizada pelo hardware. Uma forma de descrever essas condições é utilizando os operadores relacionais que comparam valores **do mesmo tipo**. Esses operadores são:

v1 = v2	verdadeiro se v1 e v2 forem iguais
v1 /= v2	verdadeiro se v1 e v2 forem diferentes
v1 < v2	verdadeiro se v1 for menor do que v2
v1 <= v2	verdadeiro se v1 for menor ou igual a v2
v1 > v2	verdadeiro se v1 for maior do que v2
v1 >= v2	verdadeiro se v1 for maior ou igual a v2

Por exemplo, o hardware descrito pelo código a seguir decrementa o sinal counter apenas se ele for maior do que zero.

```
if counter > 0 then
     counter <= counter - 1;
else
     counter <= 999;
end if;
```

As operações relacionais podem ser combinadas com operações booleanas para testar múltiplas condições. O exemplo a seguir mostra uma forma de descrever quando o contador de dezena deve ser atualizado combinando as operações relacionais que testam as condições que devem ser satisfeitas.

```
if ((count_reg = 0) and
   (seconds_reg = 9) and
   (tens_of_seconds_reg = 5) and
   (minutes_reg = 9)) then

    if tens_of_minutes_reg = 5 then
       tens_of_minutes_reg <= 0;
    else
        tens_of_minutes_reg <= tens_of_minutes_reg + 1;
    end if;
end if;
```

DICA: Atualização do testbench

As portas de entrada e saída da entidade foram alteradas, então a sua instância no *testbench* precisa ser atualizada. Para isso as seguintes alterações devem ser feitas no testbench:

- Substitua a instância do DUT por:

```
    DUT : entity work.list1(rtl)
    port map (
        clk => clk,
        rst => rst,
        seconds => seconds,
        tens_of_seconds => tens_of_seconds,
        minutes => minutes,
        tens_of_minutes => tens_of_minutes
    );
```

- Substitua a declaração de sinais por:

```
    signal clk : std_logic := '1';
    signal rst : std_logic := '1';
    signal seconds : integer range 0 to 9;
    signal tens_of_seconds : integer range 0 to 5;
    signal minutes : integer range 0 to 9;
    signal tens_of_minutes : integer range 0 to 5;
```