



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

CENTRO TECNOLÓGICO

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Circuitos Lógicos

Laboratório 8. Contador Binário “free run”

Nome: _____ Data: ____/____/____

Implemente, na FPGA, o código do contador binário apresentado na parte teórica da aula, apresentando o resultado da contagem de 8 bits e o sinal de estouro do contador (max_tick) em 9 LEDs discretos.

Atenção! A frequência de relógio da FPGA é 100MHz e, por esta razão, faz-se necessário realizar uma divisão do sinal de clk para facilitar a visualização.

Para realizar esta tarefa, vamos incluir um sinal de habilitação (*enable*) no contador binário que somente permite que o próximo estado seja atribuído ao estado atual no registrador, após um determinado número de ciclos de relógio.

Observe, abaixo o código modificado.

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4  entity free_run_bin_counter is
5      generic(N : integer := 8);
6      port(
7          clk      : in  std_logic;
8          reset    : in  std_logic;
9          enable    : in  std_logic;
10         max_tick : out std_logic;
11         q        : out std_logic_vector(N-1 downto 0)
12     );
13 end free_run_bin_counter;
14
15 architecture arch of free_run_bin_counter is
16     signal r_reg : unsigned(N-1 downto 0);
17     signal r_next : unsigned(N-1 downto 0);
18 begin
19     -- register
20     process(clk, reset)
21     begin
22         if (reset='1') then
23             r_reg <= (others=>'0');
24         elsif (clk'event and clk='1') then
25             if (enable='1') then
26                 r_reg <= r_next;
27             end if;
28         end if;
29     end process;
30     -- next-state logic
31     r_next <= r_reg + 1;
32     -- output logic
33     q <= std_logic_vector(r_reg);
34     max_tick <= '1' when r_reg=(2**N - 1) else '0';
35 end arch;
```



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

CENTRO TECNOLÓGICO

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

No arquivo *top layer*, onde fazemos a chamada a instanciação de um destes contadores, devemos gerar este sinal de habilitação a cada certo número de contagens. Inicialmente, optamos por usar uma contagem de 0 a 49.999.999, criando, desta forma, uma habilitação a cada meio segundo. Observe o código abaixo.

```
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24
25  entity top is
26      Port ( clk : in STD_LOGIC;
27            led : out STD_LOGIC_VECTOR (8 downto 0));
28  end top;
29
30  architecture lab8 of top is
31      constant N : integer := 49999999;
32      signal enable : std_logic;
33      signal divide_clk : integer range 0 to N;
34
35  begin
36      bin_counter_unit_0 : entity work.free_run_bin_counter
37          port map(
38              clk      => clk,
39              reset    => '0',
40              enable   => enable,
41              max_tick => led(8),
42              q        => led(7 downto 0)
43          );
44
45      enable <= '1' when divide_clk = N else '0';
46
47      PROCESS (clk)
48          BEGIN
49              IF (clk'EVENT AND clk='1') THEN
50                  divide_clk <= divide_clk+1;
51                  IF divide_clk = N THEN
52                      divide_clk <= 0;
53                  END IF;
54              END IF;
55          END PROCESS;
56
57  end lab8;
```



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

CENTRO TECNOLÓGICO

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Tarefas de hoje:

- Modifique o tempo de habilitação para que o circuito sequencial realize contagens a cada segundo.
- Como podemos modificar o código para realizar contagens a cada 100ms?
- Como podemos converter este contador num contador de 4 bits?