Prática de Eletrônica Digital 1 - (119466)

Turma E (Unb - Gama)

Projeto Final

Gerador de números aleatórios

Novembro 16, 2016

Nome	Matrícula	Assinatura
Arthur Temporim	14/0016759	
Eduardo Nunes	14/0056189	

1 Sumário

2 Introdução

Neste relatório é apresentado o resultado do projeto de um Gerador de números aleatórios para a disciplina de prática de eletrônica digital 1. São apresentados o funcionamento do projeto, com representações em imagens, as especificações das entradas e saídas utilizadas no kit BASYS3, juntamente, com o arquivo de restrição, o código VHDL assimm como diagrama do circuito.

3 Projeto

3.1 Objetivo

O objetivo deste projeto é implementar um gerador de números aleatórios de 2 elevado a m, menos 1 estados. Um gerador de números aleatórios é um algoritmo que gera uma sequência de números, os quais são aproximadamente independentes um dos outros. A saída da maioria dos geradores de números aleatórios não é verdadeiramente aleatória, ela somente aproxima algumas das propriedades dos números aleatórios.

3.2 Funcionamento

O projeto funciona da seguinte forma: O primeiro passo é transformar o valor binário do grupo, no caso o k (variável que seleciona um grupo de até 16 bits). Logo, em seguida a opção de ativar o modo de geração de um número, opção de nível baixo, ou de validar a geração, nível alto. Supondo se que foi selecionado a opção de gerar um número aleatório, então o projeto aplica a geração aleatória. Em seguida, da o shift nos bits de borda de subida, deslocando assim os bit que pode ser com base em um clock ou de uma chave seletora. Este faz a verificação de se os 4 últimos digitos forem iguais ao valor a variavél que seleciona o valor, no caso S. Então, há a contagem de quantas vezes o valor de S aparece no vetor. Logo em seguida, o resultado é exposto no display.

3.3 Código VHDL

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity Main is
6
7    port(
8      modo : in  std_logic := '0'; -- Gera / Valida. [sel 1]
9      k : in  std_logic_vector (3 downto 0)  := "1000"; -- Seleciona um grupo de d
10     s : in  std_logic_vector (3 downto 0)  := "0000"; -- Seleciona um valor.
   [sel 6~9]
11     hab_clk : in  std_logic := '1'; -- Habilitador de clock. [sel 10]
12     but_clk : in  std_logic := '0'; -- Bot o pra trabalhar como clock. [sel 11]
13     clk : in  std_logic := '0'; -- Clock do circuito.
14     display : out std_logic_vector (6 to 0); -- Display usado para mostrar estat
15     leds : out std_logic_vector (15 downto 0)  -- LEDs de sa da do circuito.
16   );
17
18 end Main;
19
20 architecture Behavioral of Main is
21
22   -- Vetor usado para deslocamento de bits.
23   signal vetor : std_logic_vector (15 downto 0) := "0000000000000010";
24   -- Guarda quantas vezes o valor de 'S' aparece.
25   signal estatistic : std_logic_vector (3 downto 0) := "0000";
26   -- Sinal para conectar estat stica com display.
27   signal bcd : std_logic_vector (6 to 0);
28   -- Conta quantas vezes o valor de 'S' aparece no vetor.
29   signal conta_s : integer range 0 to 15;
30
31 begin
32
33   process (vetor, clk, modo)
34     -- Transforma o valor bin rio do grupo 'k' em inteiro.
35     variable grupo : integer range 0 to 15;
36   begin
37     -- Fun  o GERA e VALIDA implementadas juntas.
38     if (modo = '0') then
39       -- Vari vel que cont m tamanho do grupo.
40       grupo := to_integer(unsigoed(k));
41       -- Aplica a gera  o aleat ria.
42       vetor(grupo) <= vetor(0) xor vetor(1);
43       -- Da o shift nos bits em borda de subida.
```

```
44        if (clk'event and clk = '1' and hab_clk = '1') then
45          vetor <= std_logic_vector(unsigned(vetor) srl 1);
46        end if;
47        -- Se os 4 ltimos digitos do vetor foram iguais ao valor de 'S' ent o c
48        if (vetor(0) = s(0) and vetor(1) = s(1) and vetor(2) = s(2) and vetor(3) =
49          conta_s <= conta_s + 1;
50        end if;
51      end if;
52    end process;
53
54    -- Atribui valor inteiro da contagem para sinal.
55    estatistica <= std_logic_vector(to_unsigned(conta_s, 4));
56
57    -- BCD.
58    process (estatistica, clk)
59    begin
60      if (estatistica = "0000") then -- 0
61        bcd <= "1111110";
62      elsif (estatistica = "0001") then -- 1
63        bcd <= "0110000";
64      elsif (estatistica = "0010") then -- 2
65        bcd <= "1101101";
66      elsif (estatistica = "0011") then -- 3
67        bcd <= "1111001";
68      elsif (estatistica = "0100") then -- 4
69        bcd <= "0110010";
70      elsif (estatistica = "0101") then -- 5
71        bcd <= "1011010";
72      elsif (estatistica = "0110") then -- 6
73        bcd <= "1011111";
74      elsif (estatistica = "0111") then -- 7
75        bcd <= "1110000";
76      elsif (estatistica = "1000") then -- 8
77        bcd <= "1111111";
78      elsif (estatistica = "1001") then -- 9
79        bcd <= "1111011";
80      elsif (estatistica = "1010") then -- A
81        bcd <= "1110111";
82      elsif (estatistica = "1011") then -- B
83        bcd <= "0011111";
84      elsif (estatistica = "1100") then -- C
85        bcd <= "1001110";
86      elsif (estatistica = "1101") then -- D
87        bcd <= "0111101";
88      elsif (estatistica = "1110") then -- E
89        bcd <= "1001111";
```

```
90      else
91        bcd <= "1000111"; -- Caso defaul -> 'F'
92      end if;
93    end process;
94
95    -- Inverte os valores do display pois    anodo.
96    display <= bcd xor "1111111";
97
98    -- Atribui o valor do vetor deslocado aos LEDs de saida.
99    leds <= vetor;
100
101 end Behavioral;
```

3.4 Diagrama do circuito

Figure 1: Diagrama do circuito de um gerador de números aleatórios

3.5 Resultados

3.6 Metodologia de testes

4 Discussão

Neste projeto final, os objetivos foram atingidos havendo a geração de números pseudo-aleatórios.

5 Conclusões

Com a realização deste experimento foi possível adquirir conhecimento a respeito de mais funções do VHDL que facilitam o trabalho do desenvolvedor assim como de projetar um circuito na FPGA utilizando se de mais recursos e ganhar bastante conhecimento a respeito da geração de números aletórios, que consiste em conhecer a existencia outras formas e métodos de se fazer a geração e verificação destes.

6 Referências Bibliográficas

Prática de Eletrônica Digital I 2016.2 professores Henrique Marra Taira Menegaz,Leonardo Aguayo, Lourdes Mattos Brasil, Marcus Vinícius Chaffim Costa, Mariana Costa Bernardes Matias. UnB - FGA Agosto de 2015.