

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

- Introdução
- Descrição do projeto
- Discussão
- Conclusões
- Referências Bibliograficas

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 assim 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 dígitos forem iguais ao valor a variável 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         — Gera / Valida. [sel 1]
9         modo : in std_logic := '0';
10        — Selecciona um grupo de at 16 bits. [sel 2 - 5]
11        k : in std_logic_vector (3 downto 0) := "1000";
12        — Selecciona um valor. [sel 6~9]
13        s : in std_logic_vector (3 downto 0) := "0000";
14        — Habilitador de clock. [sel 10]
15        hab_clk : in std_logic := '1';
16        — Bot o pra trabalhar como clock. [sel 11]
17        but_clk : in std_logic := '0';
18        — Clock do circuito.
19        clk : in std_logic := '0';
20        — Display usado para mostrar estatísticas.
21        display : out std_logic_vector (6 to 0);
22        — LEDs de sa da do circuito.
23        leds : out std_logic_vector (15 downto 0)
24    );
25
26 end Main;
27
28 architecture Behavioral of Main is
29
30     — Vetor usado para deslocamento de bits.
31     signal vetor : std_logic_vector (15 downto 0) := "0000000000000010";
32     — Guarda quantas vezes o valor de 'S' aparece.
33     signal estatistica : std_logic_vector (3 downto 0) := "0000";
34     — Sinal para conectar estatística com display.
35     signal bcd : std_logic_vector (6 to 0);
36     — Conta quantas vezes o valor de 'S' aparece no vetor.
37     signal conta_s : integer range 0 to 15;
38
39 begin
40
41     process (vetor, clk, modo)
42         — Transforma o valor binário do grupo 'k' em inteiro.
43         variable grupo : integer range 0 to 15;
44     begin
```

```

45  — Função GERA e VALIDA implementadas juntas.
46  if (modo = '0') then
47      — Variável que contém tamanho do grupo.
48      grupo := to_integer(unsigned(k));
49      — Aplica a geração aleatória.
50      vetor(grupo) <= vetor(0) xor vetor(1);
51      — Da o shift nos bits em borda de subida.
52      if (clk'event and clk = '1' and hab_clk = '1') then
53          vetor <= std_logic_vector(unsigned(vetor) srl 1);
54      elsif (but_clk'event and but_clk = '1') then
55          vetor <= std_logic_vector(unsigned(vetor) srl 1);
56      end if;
57      — VALIDA
58  — Se os 4 últimos dígitos do vetor foram iguais ao valor de 'S' então conta.
59  if (vetor(0) = s(0) and vetor(1) = s(1) and vetor(2) = s(2) and vetor(3) = s(3))
60      conta_s <= conta_s + 1;
61      end if;
62  end if;
63  end process;
64
65  — Atribui valor inteiro da contagem para sinal.
66  estatistica <= std_logic_vector(to_unsigned(conta_s, 4));
67
68  — BCD.
69  process (estatistica, clk)
70  begin
71      if (estatistica = "0000") then — 0
72          bcd <= "1111110";
73      elsif (estatistica = "0001") then — 1
74          bcd <= "0110000";
75      elsif (estatistica = "0010") then — 2
76          bcd <= "1101101";
77      elsif (estatistica = "0011") then — 3
78          bcd <= "1111001";
79      elsif (estatistica = "0100") then — 4
80          bcd <= "0110010";
81      elsif (estatistica = "0101") then — 5
82          bcd <= "1011010";
83      elsif (estatistica = "0110") then — 6
84          bcd <= "1011111";
85      elsif (estatistica = "0111") then — 7
86          bcd <= "1110000";
87      elsif (estatistica = "1000") then — 8
88          bcd <= "1111111";
89      elsif (estatistica = "1001") then — 9
90          bcd <= "1111011";

```

```

91     elsif (estatistica = "1010") then — A
92         bcd <= "1110111";
93     elsif (estatistica = "1011") then — B
94         bcd <= "0011111";
95     elsif (estatistica = "1100") then — C
96         bcd <= "1001110";
97     elsif (estatistica = "1101") then — D
98         bcd <= "0111101";
99     elsif (estatistica = "1110") then — E
100         bcd <= "1001111";
101     else
102         bcd <= "1000111"; — Caso default -> 'F'
103     end if;
104 end process;
105
106 — Inverte os valores do display pois anodo.
107 display <= not bcd;
108
109 — Atribui o valor do vetor deslocado aos LEDs de saída.
110 leds <= vetor;
111
112 end Behavioral;

```

3.4 XDC

```
1 ## This file is a general .xdc for the Basys3 rev B board
2 ## To use it in a project:
3 ## - uncomment the lines corresponding to used pins
4 ## - rename the used ports (in each line, after get_ports) according to the top
5
6 ## Clock signal
7 set_property PACKAGE_PIN W5 [get_ports clk];
8     set_property IOSTANDARD LVCMOS33 [get_ports clk];
9     create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports c
10
11 ## Switches
12 #A
13 set_property PACKAGE_PIN V17 [get_ports {k[0]}];
14     set_property IOSTANDARD LVCMOS33 [get_ports {k[0]}];
15 set_property PACKAGE_PIN V16 [get_ports {k[1]}];
16     set_property IOSTANDARD LVCMOS33 [get_ports {k[1]}];
17 set_property PACKAGE_PIN W16 [get_ports {k[2]}];
18     set_property IOSTANDARD LVCMOS33 [get_ports {k[2]}];
19 set_property PACKAGE_PIN W17 [get_ports {k[3]}];
20     set_property IOSTANDARD LVCMOS33 [get_ports {k[3]}];
21 #B
22 set_property PACKAGE_PIN W15 [get_ports {s[4]}];
23     set_property IOSTANDARD LVCMOS33 [get_ports {s[4]}];
24 set_property PACKAGE_PIN V15 [get_ports {s[5]}];
25     set_property IOSTANDARD LVCMOS33 [get_ports {s[5]}];
26 set_property PACKAGE_PIN W14 [get_ports {s[6]}];
27     set_property IOSTANDARD LVCMOS33 [get_ports {s[6]}];
28 set_property PACKAGE_PIN W13 [get_ports {s[7]}];
29     set_property IOSTANDARD LVCMOS33 [get_ports {s[7]}];
30 #set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
31 #set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
32 #set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
33 #set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
34 set_property PACKAGE_PIN T2 [get_ports hab_clk]
35     set_property IOSTANDARD LVCMOS33 [get_ports hab_clk]
36 set_property PACKAGE_PIN R3 [get_ports but_clk]
37     set_property IOSTANDARD LVCMOS33 [get_ports but_clk]
38 #set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
39 #set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
40 #set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
41 #set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
42 #set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
43 #set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
44 set_property PACKAGE_PIN R2 [get_ports modo]
```

```

45     set_property IOSTANDARD LVCMOS33 [get_ports modo]
46
47
48 ## LEDs
49 set_property PACKAGE_PIN U16 [get_ports {led[0]}]
50 set_property PACKAGE_PIN E19 [get_ports {led[1]}]
51     set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
52 set_property PACKAGE_PIN U19 [get_ports {led[2]}]
53     set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
54 set_property PACKAGE_PIN V19 [get_ports {led[3]}]
55     set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
56 set_property PACKAGE_PIN W18 [get_ports {led[4]}]
57     set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
58 set_property PACKAGE_PIN U15 [get_ports {led[5]}]
59     set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
60 set_property PACKAGE_PIN U14 [get_ports {led[6]}]
61     set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
62 set_property PACKAGE_PIN V14 [get_ports {led[7]}]
63     set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
64 set_property PACKAGE_PIN V13 [get_ports {led[8]}]
65     set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
66 set_property PACKAGE_PIN V3 [get_ports {led[9]}]
67     set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
68 set_property PACKAGE_PIN W3 [get_ports {led[10]}]
69     set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
70 set_property PACKAGE_PIN U3 [get_ports {led[11]}]
71     set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
72 set_property PACKAGE_PIN P3 [get_ports {led[12]}]
73     set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
74 set_property PACKAGE_PIN N3 [get_ports {led[13]}]
75     set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
76 set_property PACKAGE_PIN P1 [get_ports {led[14]}]
77     set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
78 set_property PACKAGE_PIN L1 [get_ports {led[15]}]
79     set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]
80
81
82 ##7 segment display
83 #set_property PACKAGE_PIN W7 [get_ports {bcd[0]}];
84 # set_property IOSTANDARD LVCMOS33 [get_ports {bcd[0]}];
85 #set_property PACKAGE_PIN W6 [get_ports {bcd[1]}];
86 # set_property IOSTANDARD LVCMOS33 [get_ports {bcd[1]}];
87 #set_property PACKAGE_PIN U8 [get_ports {bcd[2]}];
88 # set_property IOSTANDARD LVCMOS33 [get_ports {bcd[2]}];
89 #set_property PACKAGE_PIN V8 [get_ports {bcd[3]}];
90 # set_property IOSTANDARD LVCMOS33 [get_ports {bcd[3]}];

```

set_property IOSTANDARD


```

91 #set_property PACKAGE_PIN U5 [get_ports {bcd[4]}}];
92 # set_property IOSTANDARD LVCMOS33 [get_ports {bcd[4]}}];
93 #set_property PACKAGE_PIN V5 [get_ports {bcd[5]}}];
94 # set_property IOSTANDARD LVCMOS33 [get_ports {bcd[5]}}];
95 #set_property PACKAGE_PIN U7 [get_ports {bcd[6]}}];
96 # set_property IOSTANDARD LVCMOS33 [get_ports {bcd[6]}}];
97
98
99 set_property PACKAGE_PIN W7 [get_ports {display1[0]}}];
100 set_property IOSTANDARD LVCMOS33 [get_ports {display1[0]}}];
101 set_property PACKAGE_PIN W6 [get_ports {display1[1]}}];
102 set_property IOSTANDARD LVCMOS33 [get_ports {display1[1]}}];
103 set_property PACKAGE_PIN U8 [get_ports {display1[2]}}];
104 set_property IOSTANDARD LVCMOS33 [get_ports {display1[2]}}];
105 set_property PACKAGE_PIN V8 [get_ports {display1[3]}}];
106 set_property IOSTANDARD LVCMOS33 [get_ports {display1[3]}}];
107 set_property PACKAGE_PIN U5 [get_ports {display1[4]}}];
108 set_property IOSTANDARD LVCMOS33 [get_ports {display1[4]}}];
109 set_property PACKAGE_PIN V5 [get_ports {display1[5]}}];
110 set_property IOSTANDARD LVCMOS33 [get_ports {display1[5]}}];
111 set_property PACKAGE_PIN U7 [get_ports {display1[6]}}];
112 set_property IOSTANDARD LVCMOS33 [get_ports {display1[6]}}];
113
114
115
116 #set_property PACKAGE_PIN V7 [get_ports dp]
117 #set_property IOSTANDARD LVCMOS33 [get_ports dp]
118
119 #set_property PACKAGE_PIN U2 [get_ports {an[0]}}]
120 #set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}}]
121 #set_property PACKAGE_PIN U4 [get_ports {an[1]}}]
122 #set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}}]
123 #set_property PACKAGE_PIN V4 [get_ports {an[2]}}]
124 #set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}}]
125 #set_property PACKAGE_PIN W4 [get_ports {an[3]}}]
126 #set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}}]
127
128
129 ##Buttons
130 #set_property PACKAGE_PIN U18 [get_ports hab_clk] set_property IOSTANDARD
131 #set_property PACKAGE_PIN T18 [get_ports but_clk]
132 # set_property IOSTANDARD LVCMOS33 [get_ports but_clk]
133 #set_property PACKAGE_PIN W19 [get_ports btnL] #set_property IOSTANDARD
134 #set_property PACKAGE_PIN T17 [get_ports btnR]
135 #set_property IOSTANDARD LVCMOS33 [get_ports btnR]
136 #set_property PACKAGE_PIN U17 [get_ports btnD]

```

```

137     #set_property IOSTANDARD LVCOS33 [get_ports btnD]
138
139
140
141 ###Pmod Header JA
142 ###Sch name = JA1
143 #set_property PACKAGE_PIN J1 [get_ports {JA[0]}]
144     #set_property IOSTANDARD LVCOS33 [get_ports {JA[0]}]
145 ###Sch name = JA2
146 #set_property PACKAGE_PIN L2 [get_ports {JA[1]}]
147     #set_property IOSTANDARD LVCOS33 [get_ports {JA[1]}]
148 ###Sch name = JA3
149 #set_property PACKAGE_PIN J2 [get_ports {JA[2]}]
150     #set_property IOSTANDARD LVCOS33 [get_ports {JA[2]}]
151 ###Sch name = JA4
152 #set_property PACKAGE_PIN G2 [get_ports {JA[3]}]
153     #set_property IOSTANDARD LVCOS33 [get_ports {JA[3]}]
154 ###Sch name = JA7
155 #set_property PACKAGE_PIN H1 [get_ports {JA[4]}]
156     #set_property IOSTANDARD LVCOS33 [get_ports {JA[4]}]
157 ###Sch name = JA8
158 #set_property PACKAGE_PIN K2 [get_ports {JA[5]}]
159     #set_property IOSTANDARD LVCOS33 [get_ports {JA[5]}]
160 ###Sch name = JA9
161 #set_property PACKAGE_PIN H2 [get_ports {JA[6]}]
162     #set_property IOSTANDARD LVCOS33 [get_ports {JA[6]}]
163 ###Sch name = JA10
164 #set_property PACKAGE_PIN G3 [get_ports {JA[7]}]
165     #set_property IOSTANDARD LVCOS33 [get_ports {JA[7]}]
166
167
168
169 ###Pmod Header JB
170 ###Sch name = JB1
171 #set_property PACKAGE_PIN A14 [get_ports {JB[0]}]
172     #set_property IOSTANDARD LVCOS33 [get_ports {JB[0]}]
173 ###Sch name = JB2
174 #set_property PACKAGE_PIN A16 [get_ports {JB[1]}]
175     #set_property IOSTANDARD LVCOS33 [get_ports {JB[1]}]
176 ###Sch name = JB3
177 #set_property PACKAGE_PIN B15 [get_ports {JB[2]}]
178     #set_property IOSTANDARD LVCOS33 [get_ports {JB[2]}]
179 ###Sch name = JB4
180 #set_property PACKAGE_PIN B16 [get_ports {JB[3]}]
181     #set_property IOSTANDARD LVCOS33 [get_ports {JB[3]}]
182 ###Sch name = JB7

```

```

183 #set_property PACKAGE_PIN A15 [get_ports {JB[4]}]
184 #set_property IOSTANDARD LVC MOS33 [get_ports {JB[4]}]
185 ##Sch name = JB8
186 #set_property PACKAGE_PIN A17 [get_ports {JB[5]}]
187 #set_property IOSTANDARD LVC MOS33 [get_ports {JB[5]}]
188 ##Sch name = JB9
189 #set_property PACKAGE_PIN C15 [get_ports {JB[6]}]
190 #set_property IOSTANDARD LVC MOS33 [get_ports {JB[6]}]
191 ##Sch name = JB10
192 #set_property PACKAGE_PIN C16 [get_ports {JB[7]}]
193 #set_property IOSTANDARD LVC MOS33 [get_ports {JB[7]}]
194
195
196
197 ##Pmod Header JC
198 ##Sch name = JC1
199 #set_property PACKAGE_PIN K17 [get_ports {JC[0]}]
200 #set_property IOSTANDARD LVC MOS33 [get_ports {JC[0]}]
201 ##Sch name = JC2
202 #set_property PACKAGE_PIN M18 [get_ports {JC[1]}]
203 #set_property IOSTANDARD LVC MOS33 [get_ports {JC[1]}]
204 ##Sch name = JC3
205 #set_property PACKAGE_PIN N17 [get_ports {JC[2]}]
206 #set_property IOSTANDARD LVC MOS33 [get_ports {JC[2]}]
207 ##Sch name = JC4
208 #set_property PACKAGE_PIN P18 [get_ports {JC[3]}]
209 #set_property IOSTANDARD LVC MOS33 [get_ports {JC[3]}]
210 ##Sch name = JC7
211 #set_property PACKAGE_PIN L17 [get_ports {JC[4]}]
212 #set_property IOSTANDARD LVC MOS33 [get_ports {JC[4]}]
213 ##Sch name = JC8
214 #set_property PACKAGE_PIN M19 [get_ports {JC[5]}]
215 #set_property IOSTANDARD LVC MOS33 [get_ports {JC[5]}]
216 ##Sch name = JC9
217 #set_property PACKAGE_PIN P17 [get_ports {JC[6]}]
218 #set_property IOSTANDARD LVC MOS33 [get_ports {JC[6]}]
219 ##Sch name = JC10
220 #set_property PACKAGE_PIN R18 [get_ports {JC[7]}]
221 #set_property IOSTANDARD LVC MOS33 [get_ports {JC[7]}]
222
223
224 ##Pmod Header JXADC
225 ##Sch name = XA1_P
226 #set_property PACKAGE_PIN J3 [get_ports {JXADC[0]}]
227 #set_property IOSTANDARD LVC MOS33 [get_ports {JXADC[0]}]
228 ##Sch name = XA2_P

```

```

229 #set_property PACKAGE_PIN L3 [get_ports {JXADC[1]}]
230 #set_property IOSTANDARD LVC MOS33 [get_ports {JXADC[1]}]
231 ##Sch name = XA3_P
232 #set_property PACKAGE_PIN M2 [get_ports {JXADC[2]}]
233 #set_property IOSTANDARD LVC MOS33 [get_ports {JXADC[2]}]
234 ##Sch name = XA4_P
235 #set_property PACKAGE_PIN N2 [get_ports {JXADC[3]}]
236 #set_property IOSTANDARD LVC MOS33 [get_ports {JXADC[3]}]
237 ##Sch name = XA1_N
238 #set_property PACKAGE_PIN K3 [get_ports {JXADC[4]}]
239 #set_property IOSTANDARD LVC MOS33 [get_ports {JXADC[4]}]
240 ##Sch name = XA2_N
241 #set_property PACKAGE_PIN M3 [get_ports {JXADC[5]}]
242 #set_property IOSTANDARD LVC MOS33 [get_ports {JXADC[5]}]
243 ##Sch name = XA3_N
244 #set_property PACKAGE_PIN M1 [get_ports {JXADC[6]}]
245 #set_property IOSTANDARD LVC MOS33 [get_ports {JXADC[6]}]
246 ##Sch name = XA4_N
247 #set_property PACKAGE_PIN N1 [get_ports {JXADC[7]}]
248 #set_property IOSTANDARD LVC MOS33 [get_ports {JXADC[7]}]
249
250
251
252 ##VGA Connector
253 #set_property PACKAGE_PIN G19 [get_ports {vgaRed[0]}]
254 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaRed[0]}]
255 #set_property PACKAGE_PIN H19 [get_ports {vgaRed[1]}]
256 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaRed[1]}]
257 #set_property PACKAGE_PIN J19 [get_ports {vgaRed[2]}]
258 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaRed[2]}]
259 #set_property PACKAGE_PIN N19 [get_ports {vgaRed[3]}]
260 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaRed[3]}]
261 #set_property PACKAGE_PIN N18 [get_ports {vgaBlue[0]}]
262 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaBlue[0]}]
263 #set_property PACKAGE_PIN L18 [get_ports {vgaBlue[1]}]
264 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaBlue[1]}]
265 #set_property PACKAGE_PIN K18 [get_ports {vgaBlue[2]}]
266 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaBlue[2]}]
267 #set_property PACKAGE_PIN J18 [get_ports {vgaBlue[3]}]
268 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaBlue[3]}]
269 #set_property PACKAGE_PIN J17 [get_ports {vgaGreen[0]}]
270 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaGreen[0]}]
271 #set_property PACKAGE_PIN H17 [get_ports {vgaGreen[1]}]
272 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaGreen[1]}]
273 #set_property PACKAGE_PIN G17 [get_ports {vgaGreen[2]}]
274 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaGreen[2]}]

```

```

275 #set_property PACKAGE_PIN D17 [get_ports {vgaGreen[3]}]
276 #set_property IOSTANDARD LVC MOS33 [get_ports {vgaGreen[3]}]
277 #set_property PACKAGE_PIN P19 [get_ports Hsync]
278 #set_property IOSTANDARD LVC MOS33 [get_ports Hsync]
279 #set_property PACKAGE_PIN R19 [get_ports Vsync]
280 #set_property IOSTANDARD LVC MOS33 [get_ports Vsync]
281
282
283 ##USB-RS232 Interface
284 #set_property PACKAGE_PIN B18 [get_ports RsRx]
285 #set_property IOSTANDARD LVC MOS33 [get_ports RsRx]
286 #set_property PACKAGE_PIN A18 [get_ports RsTx]
287 #set_property IOSTANDARD LVC MOS33 [get_ports RsTx]
288
289
290 ##USB HID (PS/2)
291 #set_property PACKAGE_PIN C17 [get_ports PS2Clk]
292 #set_property IOSTANDARD LVC MOS33 [get_ports PS2Clk]
293 #set_property PULLUP true [get_ports PS2Clk]
294 #set_property PACKAGE_PIN B17 [get_ports PS2Data]
295 #set_property IOSTANDARD LVC MOS33 [get_ports PS2Data]
296 #set_property PULLUP true [get_ports PS2Data]
297
298
299 ##Quad SPI Flash
300 ##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using
301 ##STARTUPE2 primitive.
302 #set_property PACKAGE_PIN D18 [get_ports {QspiDB[0]}]
303 #set_property IOSTANDARD LVC MOS33 [get_ports {QspiDB[0]}]
304 #set_property PACKAGE_PIN D19 [get_ports {QspiDB[1]}]
305 #set_property IOSTANDARD LVC MOS33 [get_ports {QspiDB[1]}]
306 #set_property PACKAGE_PIN G18 [get_ports {QspiDB[2]}]
307 #set_property IOSTANDARD LVC MOS33 [get_ports {QspiDB[2]}]
308 #set_property PACKAGE_PIN F18 [get_ports {QspiDB[3]}]
309 #set_property IOSTANDARD LVC MOS33 [get_ports {QspiDB[3]}]
310 #set_property PACKAGE_PIN K19 [get_ports QspiCSn]
311 #set_property IOSTANDARD LVC MOS33 [get_ports QspiCSn]

```

3.5 Diagrama do circuito

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

3.6 Resultados

3.7 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 aleatórios, 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.