

UNIVERSIDADE SÃO JUDAS TADEU
Faculdade de Tecnologia e Ciências Exatas
Engenharia Eletrônica – 4ºAno

Felipe Bonifácio	201411014
Marcus Henrique C. da Silva	201314331
Otavio Augusto F. Cruzatto	817116140
Vinicius Ferreira Mylonas	81712774

Projeto:

Controle de Motor AC com PWM

Sistemas Eletrônicos Digitais/Eletrônica de Potência

São Paulo – SP

2017

1. INTRODUÇÃO

Na vida profissional, um engenheiro deve dispor das mais diversas ferramentas para alcançar a melhor eficiência na execução dos projetos. Sendo assim, se faz necessária a busca por conhecimento. Neste projeto fomos incumbidos de realizar algo que, para nossa compreensão, era extremamente difícil, pois não dispúnhamos de muita informação sobre o método. Da escassez surgiu a necessidade, dela surgiu o empenho e com ele nosso grupo concluiu o projeto com sucesso.

2. OBJETIVO

Conforme solicitado pelos professores, o escopo do projeto era projetar e confeccionar um inversor de frequência com a FPGA em VHDL para controlar a velocidade de um Motor AC através da modulação PWM.

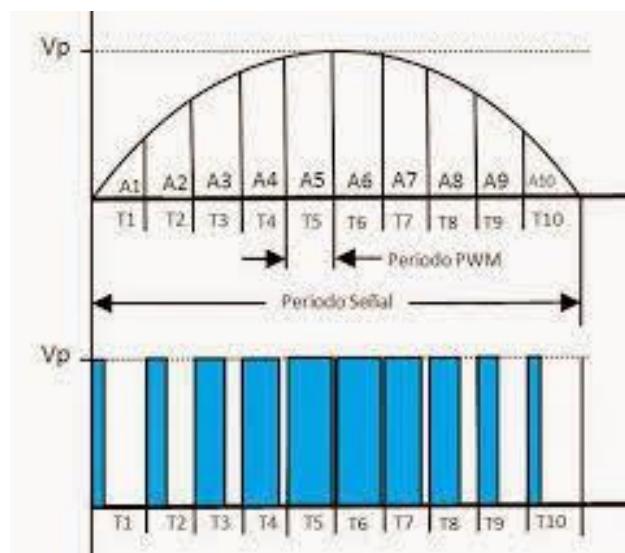
3. METODOLOGIA

Para realizar todo o controle do projeto, utilizamos uma placa de desenvolvimento com tecnologia FPGA (*Field Programmable Gate Array ou Matriz de Portas Programáveis em Campo*) disponibilizada pela faculdade. Produzidos a partir de ligas de semicondutores, estes são chips são feitos com blocos lógicos programáveis que podem ser configurados de acordo com a necessidade do projeto. Quando configurados, os blocos funcionam quase como dispositivos eletrônicos reais, ou seja, adotam o comportamento do hardware conforme descrito pelo projetista. Para que isto ocorra, é necessário utilizar a linguagem de descrição de hardware VHDL (*Very High Speed Integrated Circuits Hardware Description Language*) em que é possível compilar, verificar erros e realizar simulações do projeto antes da finalização.

Descrição lógica

Como podemos observar, a FPGA é um dispositivo lógico e, portanto, trabalha com baixas correntes e tensões contínuas. Para realizar o controle do motor de corrente alternada, é necessário um sinal de corrente alternada, ou seja um sinal senoidal, e potência bem superior à que a placa FPGA pode fornecer.

Utilizamos o conceito de modulação por largura de pulso, ou PWM (Pulse-Width Modulation), para que a FPGA pudesse fornecer este sinal variando o Duty Cycle de forma crescente até o que representaria um quarto do período da senóide e decrescente deste ponto até meio período conforme imagem abaixo. Desta forma criamos o semicírculo positivo da onda.



Contudo, a placa só trabalha com tensões positivas e não foi possível criarmos o semicírculo negativo. A solução encontrada foi fornecer dois sinais de meia onda defasados entre si em 180° demonstrados abaixo e distribuir cada um em um pino para que o tratamento deles fosse realizado por hardware.

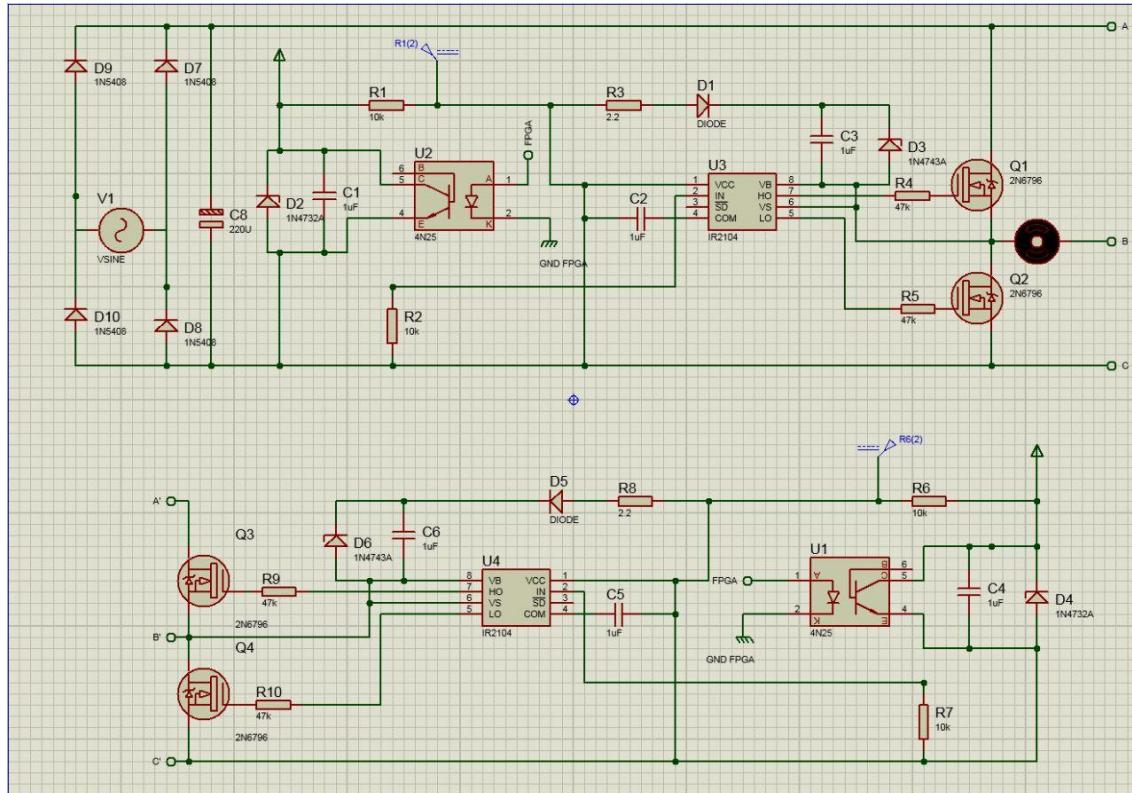


Para controlar a velocidade do motor, dimensionamos o PWM para que variasse a frequência da senóide de acordo com o valor que o usuário inserisse nos botões de controle da FPGA conforme anexo A.

Todos os programas, na íntegra se encontram no anexo B.

Hardware

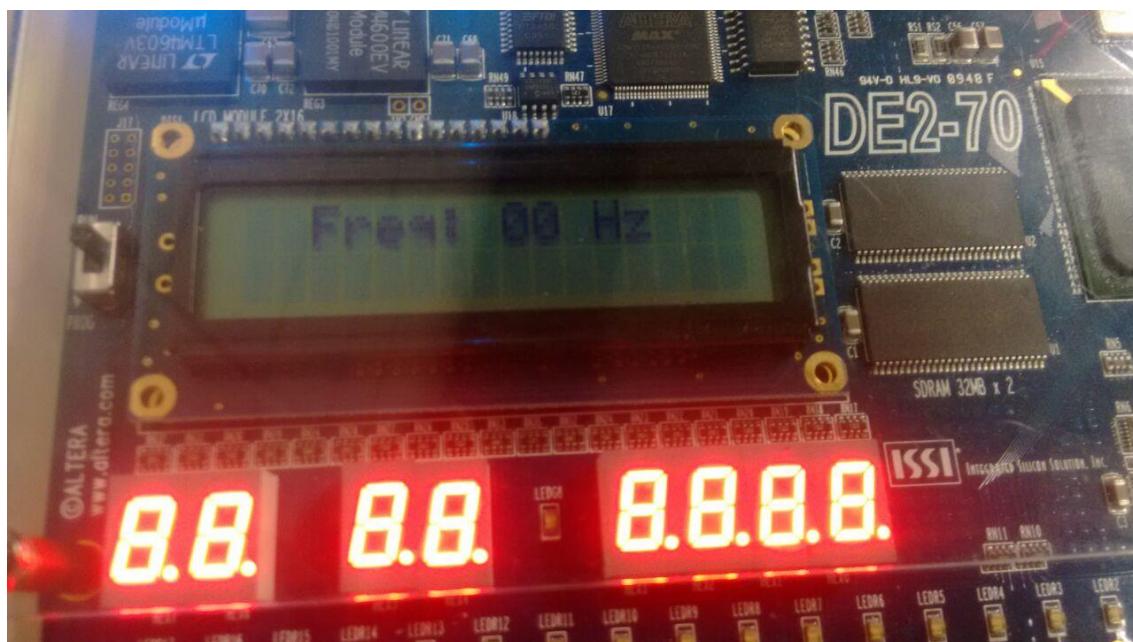
Além de executar a função de transformar os dois sinais de meia onda em uma senóide completa, o circuito utilizado tem como objetivo fornecer a potência adequada para que o motor funcione corretamente através do sinal recebido da FPGA. Nele utilizamos o MOSFET IRF540 em uma aplicação muito utilizada para o controle de motores, a ponte H. O objetivo desta ponte, é controlar a direção de rotação do motor utilizado. O CI IR2104 é quem, após receber o sinal da fpga, controla a ponte H.



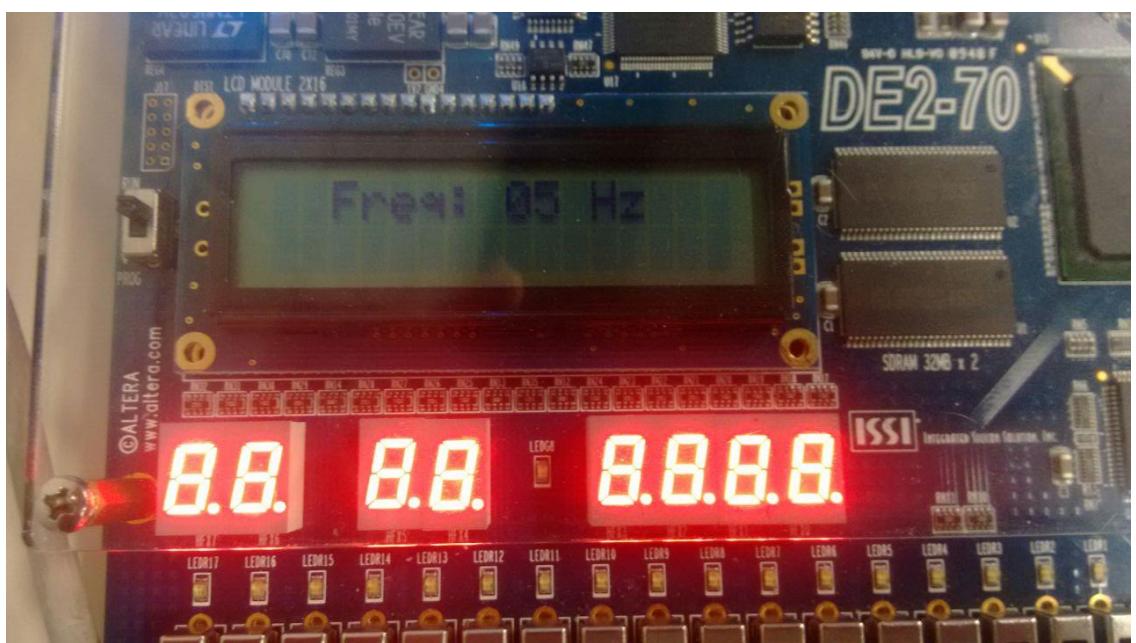
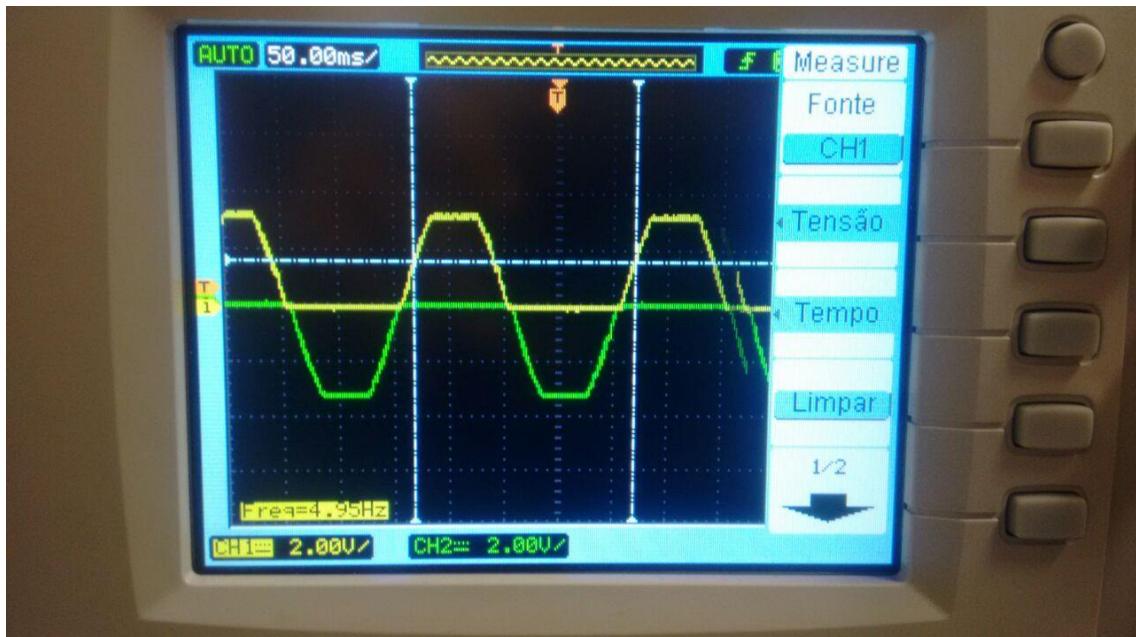
Anexo A

Representação gráfica do sinal no Osciloscópio de acordo com o valor inserido na FPGA variando entre 0Hz e 60Hz em passos de 5Hz.

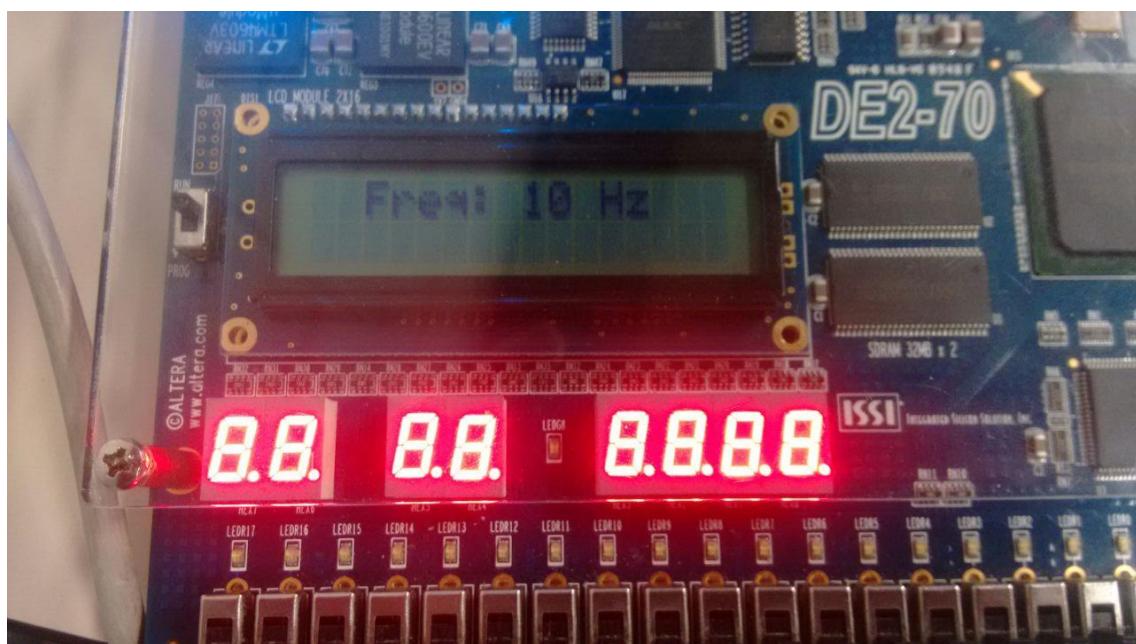
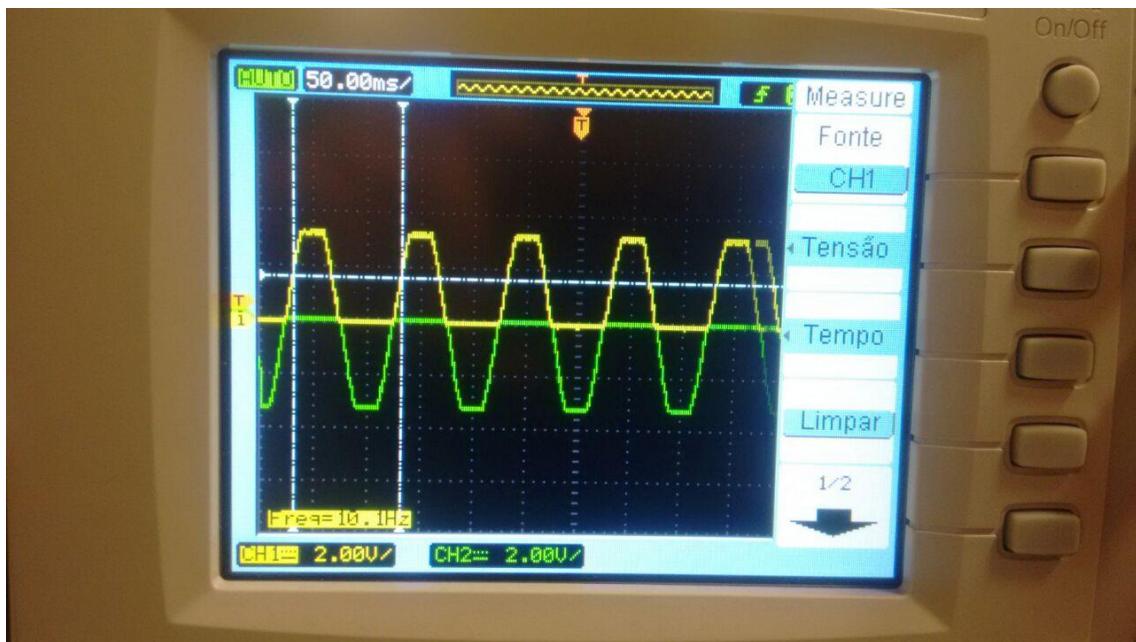
0Hz:



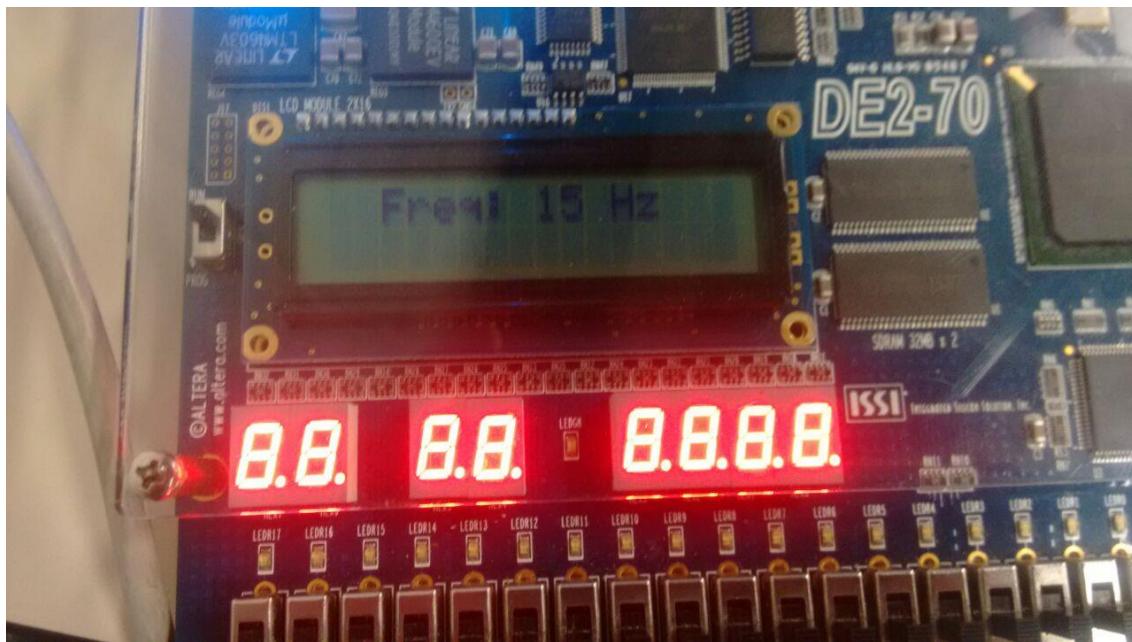
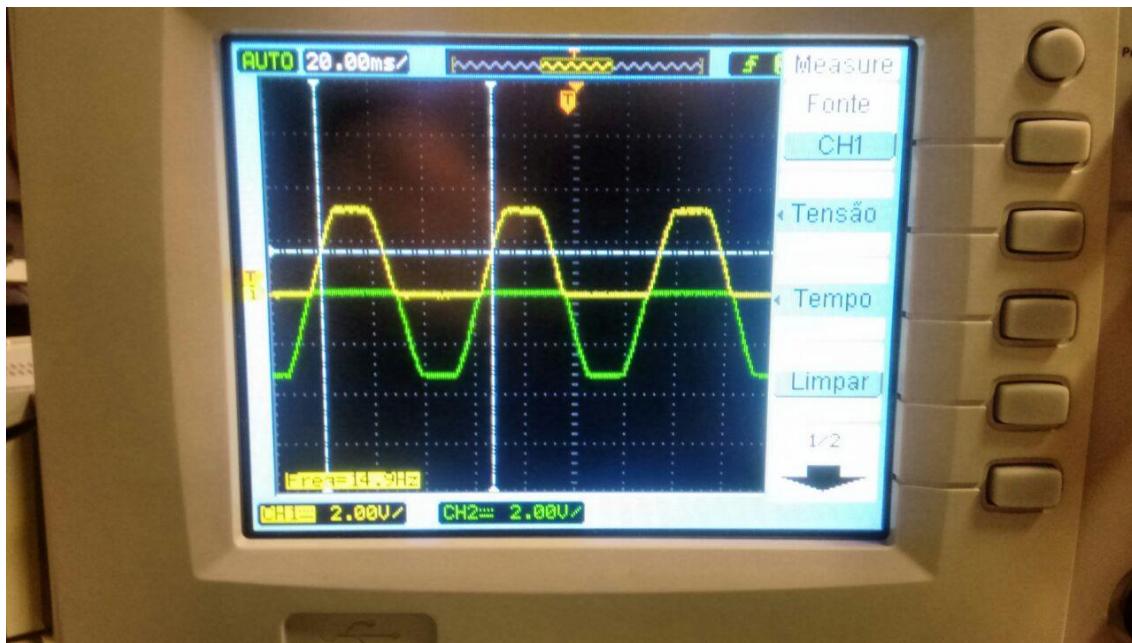
5Hz:



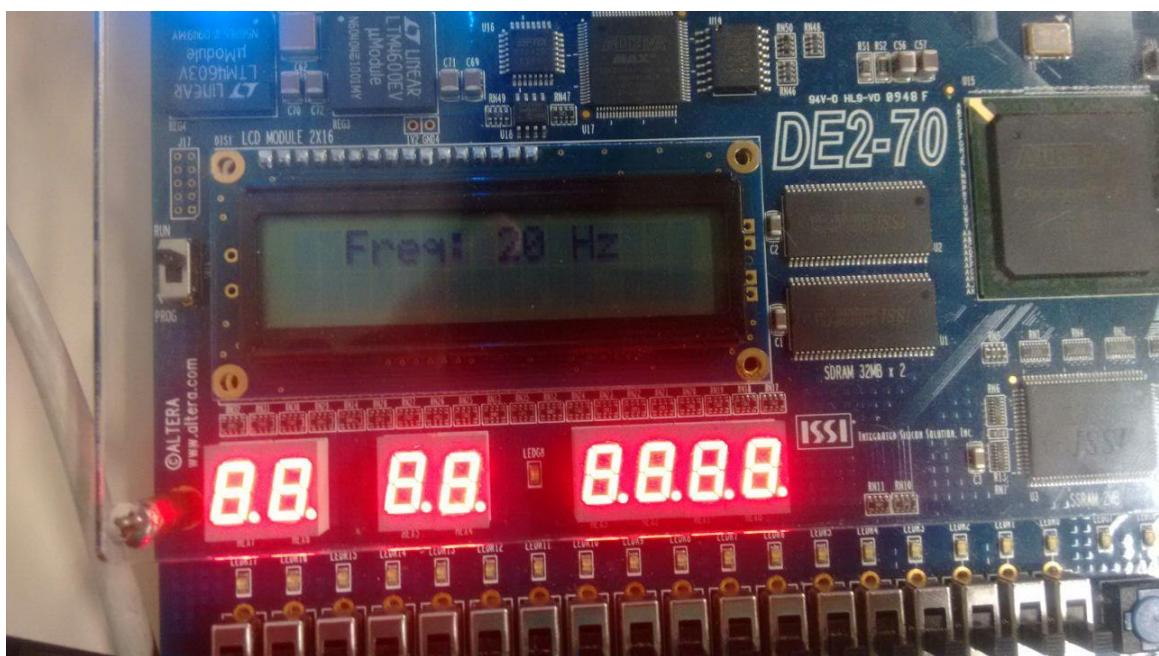
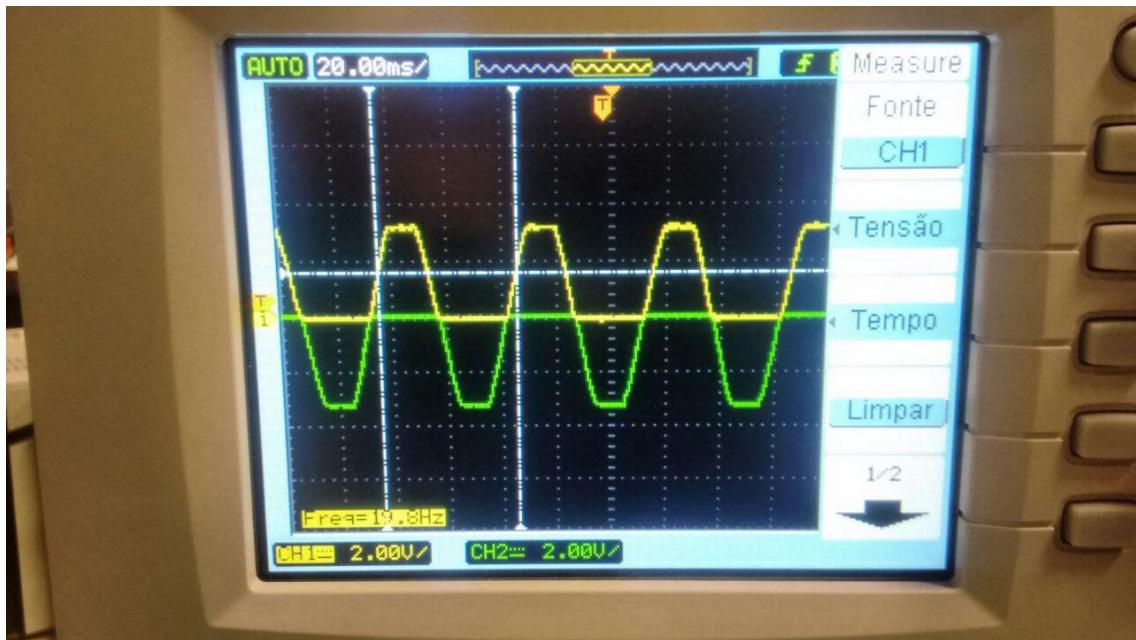
10 Hz:



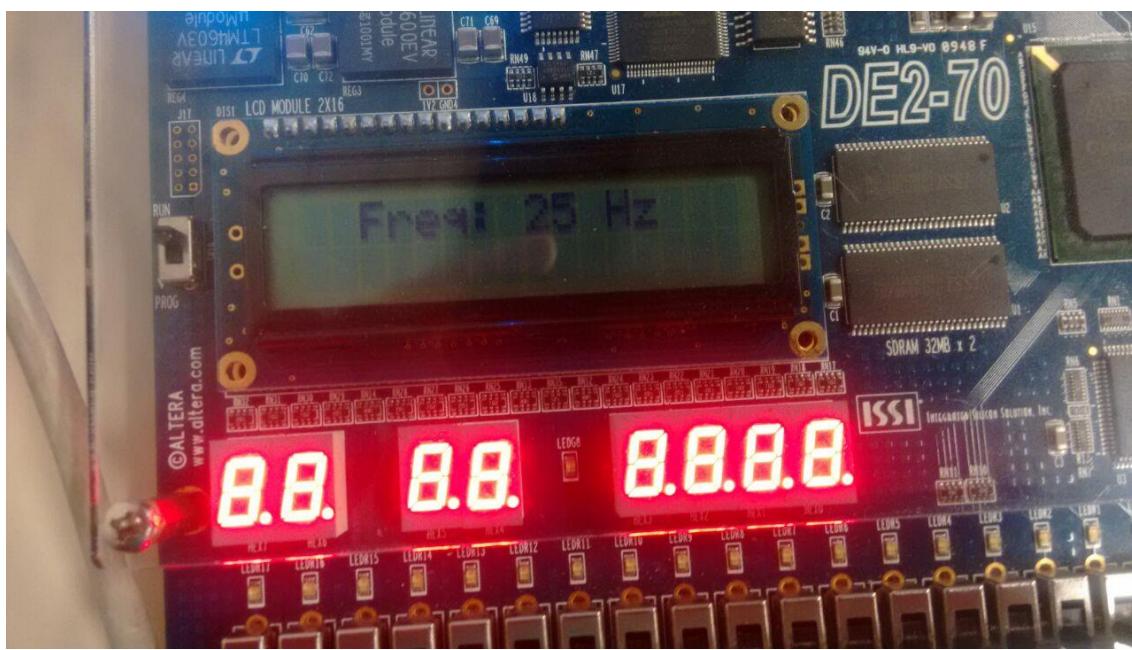
15 Hz:



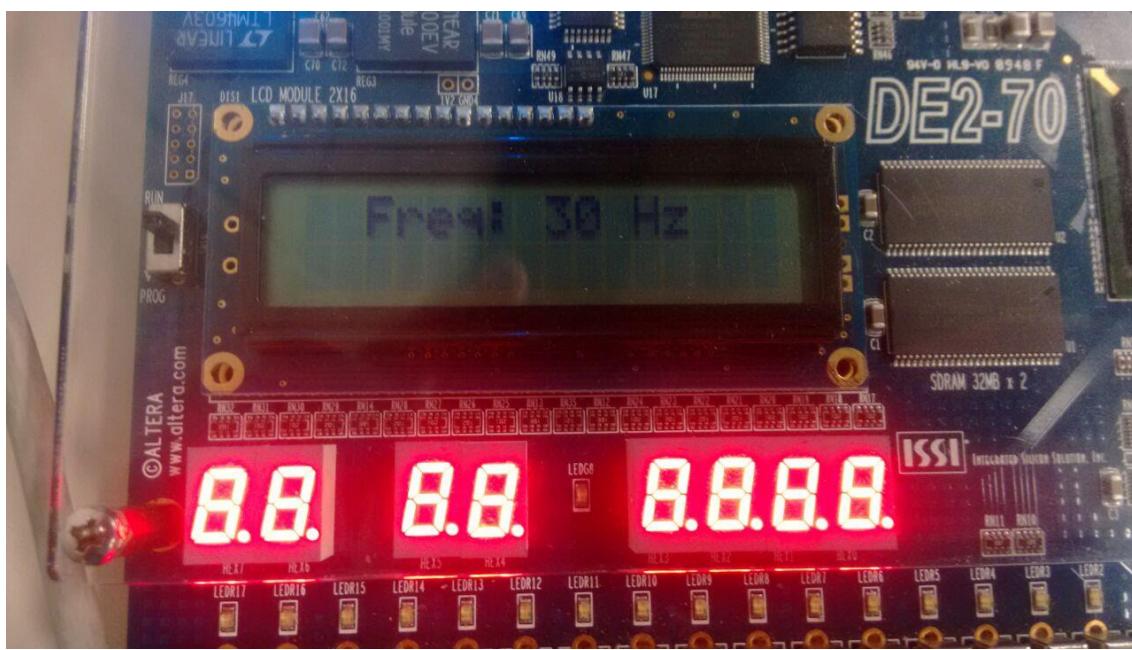
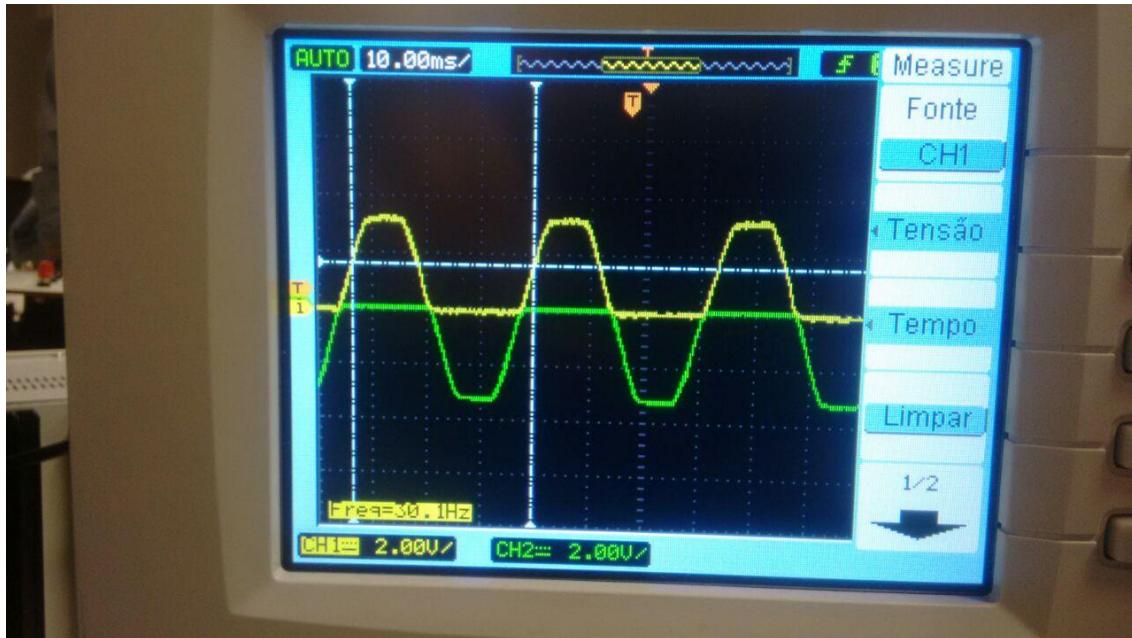
20 Hz:



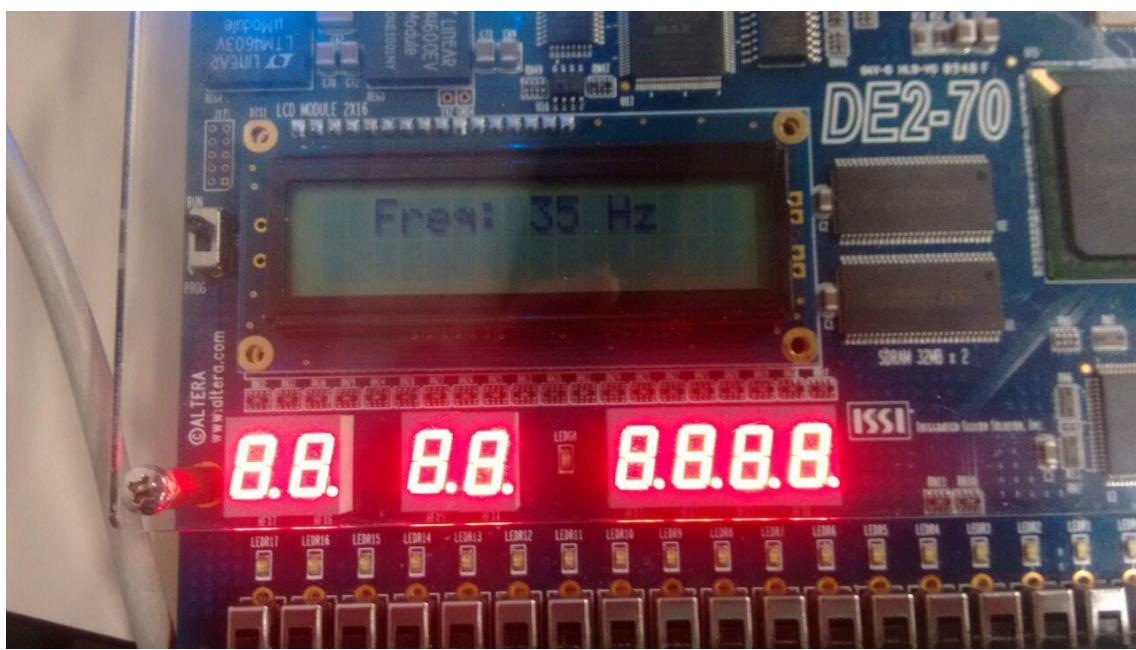
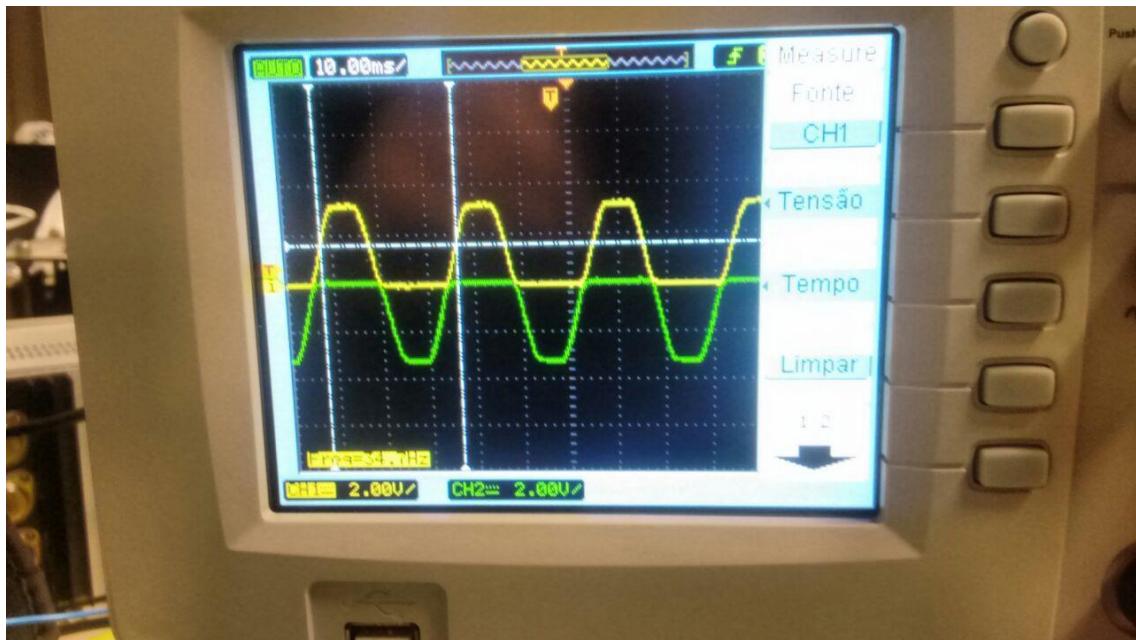
25 Hz:



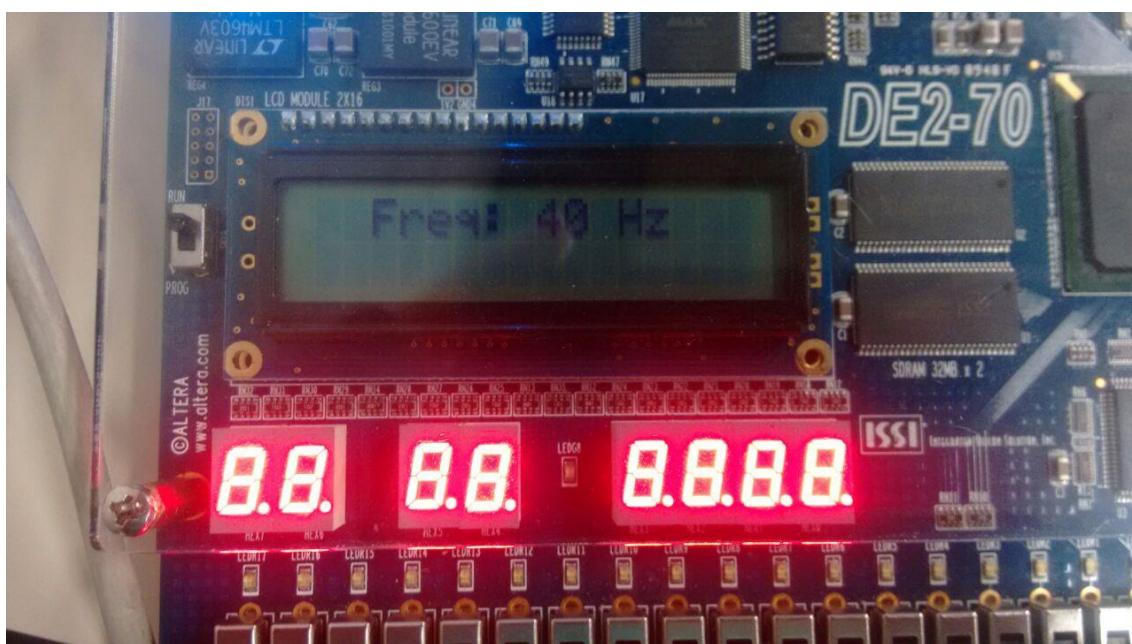
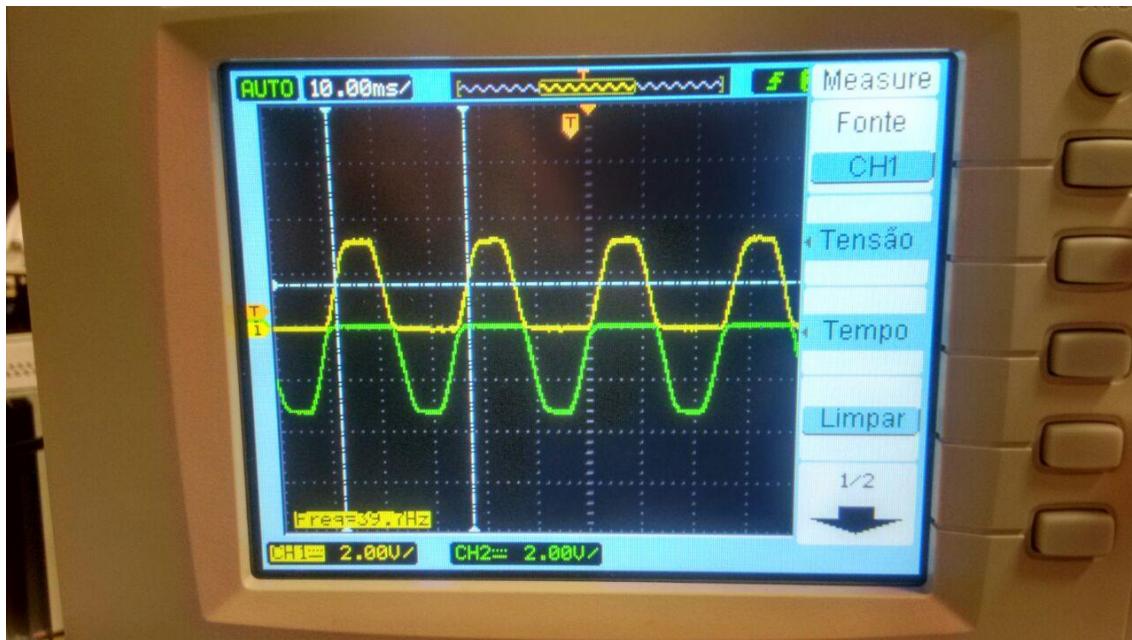
30 Hz:



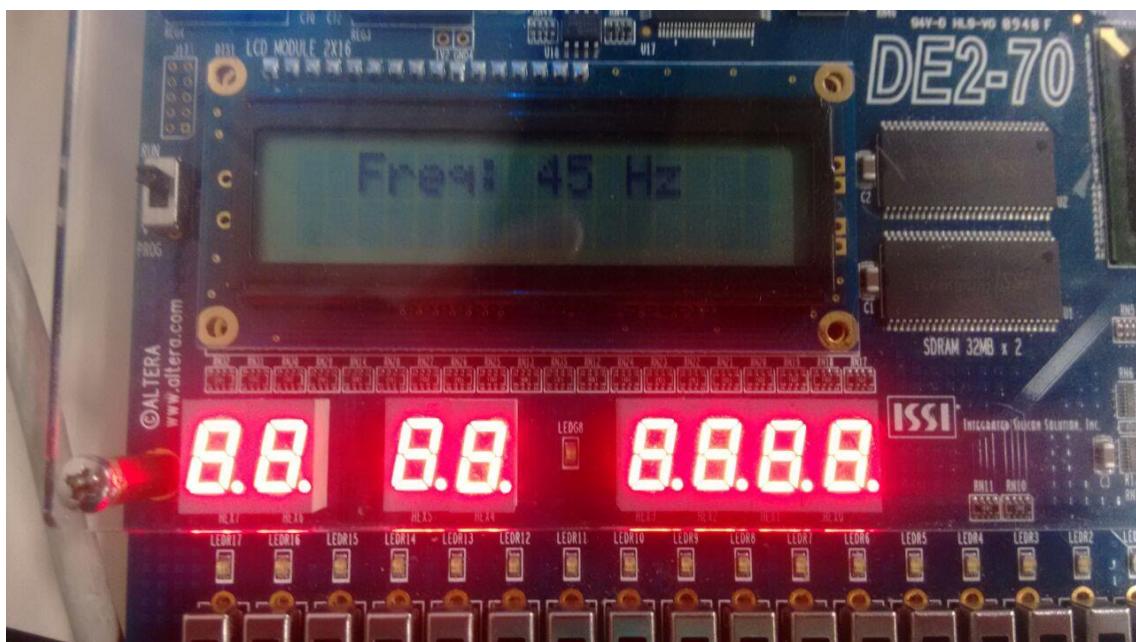
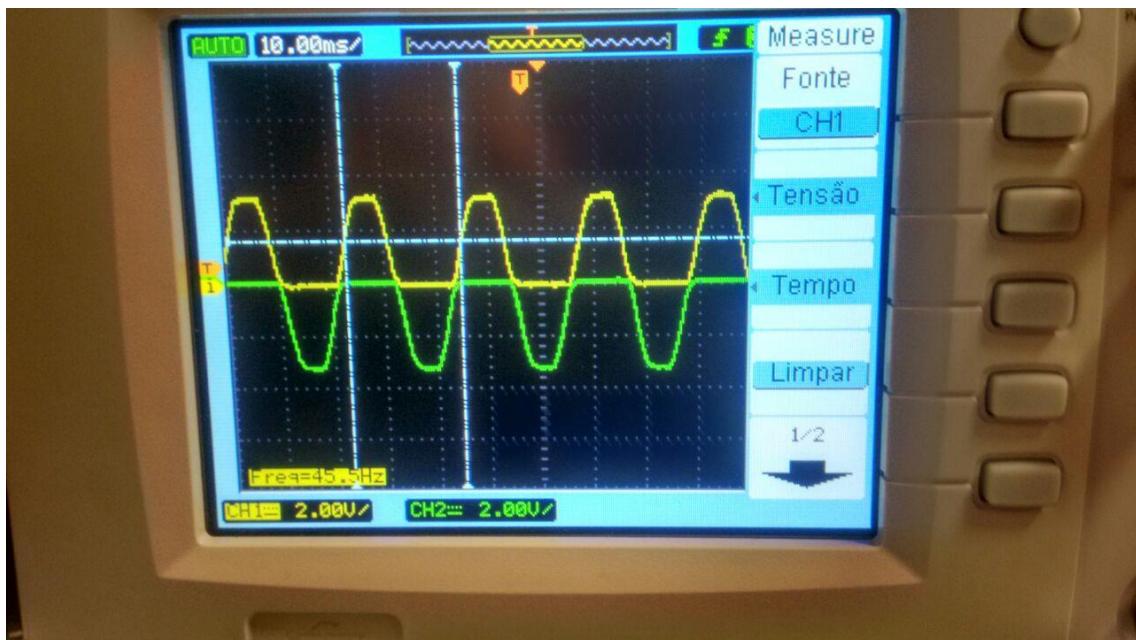
35 Hz:



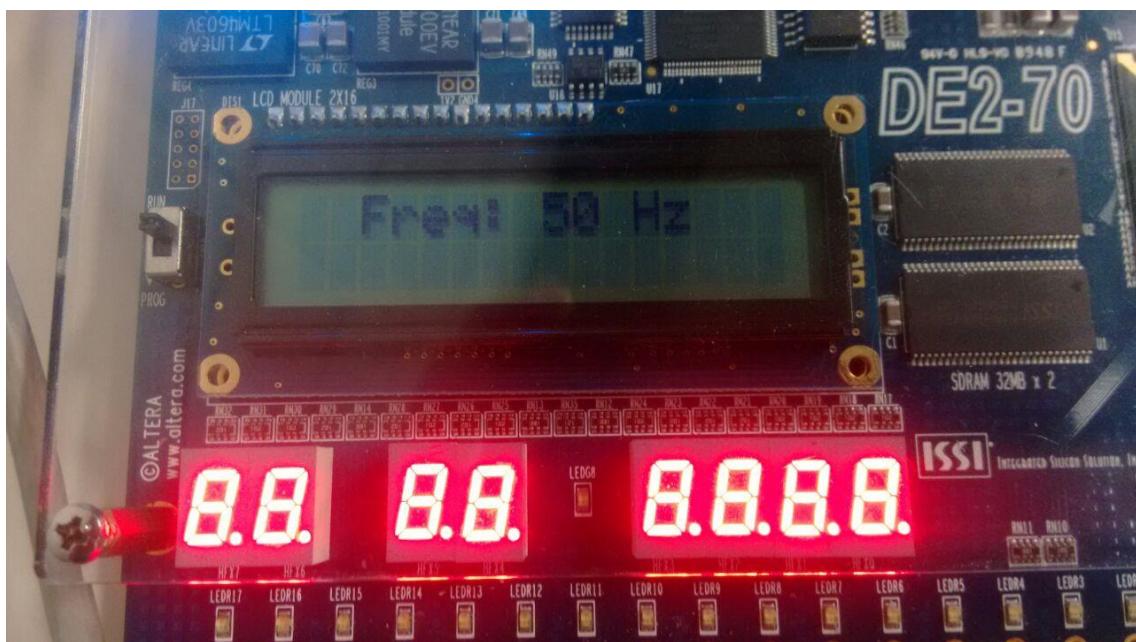
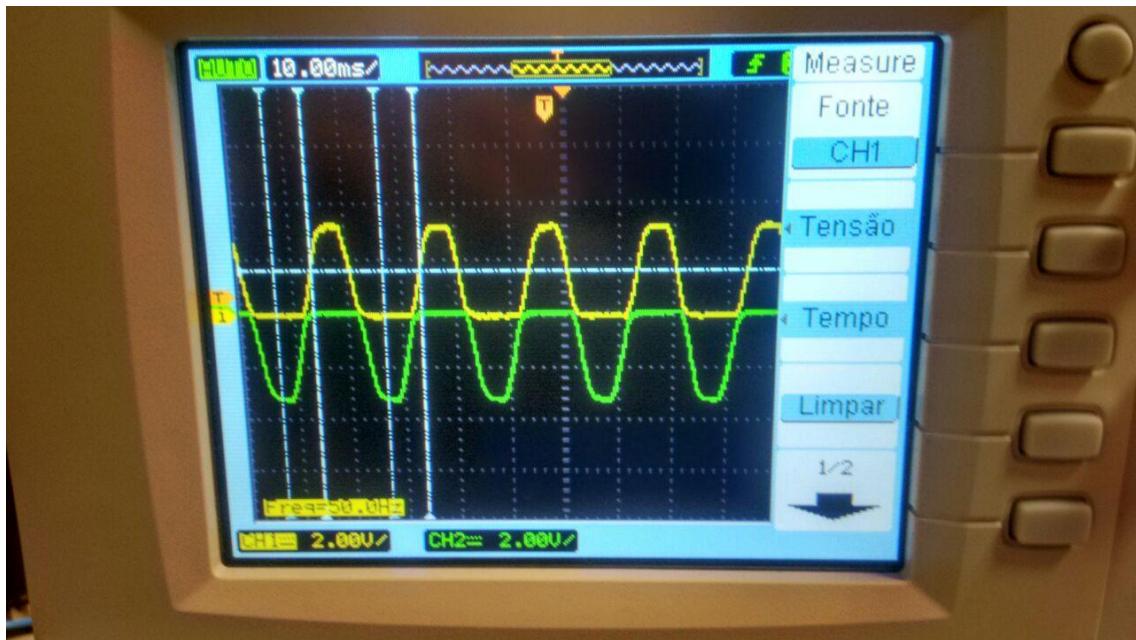
40 Hz:



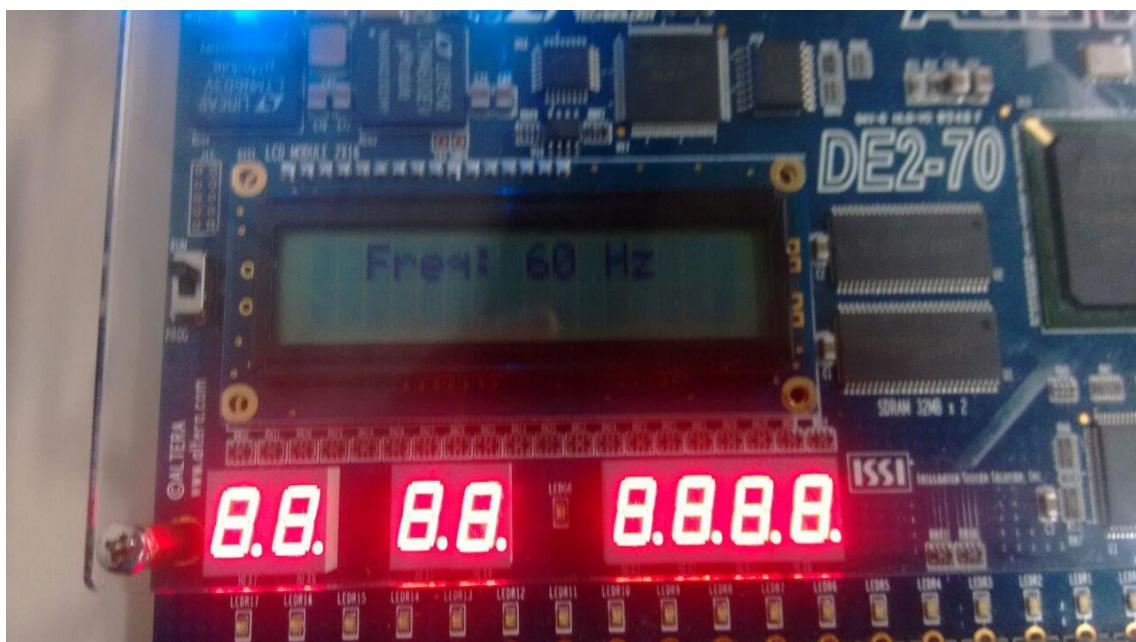
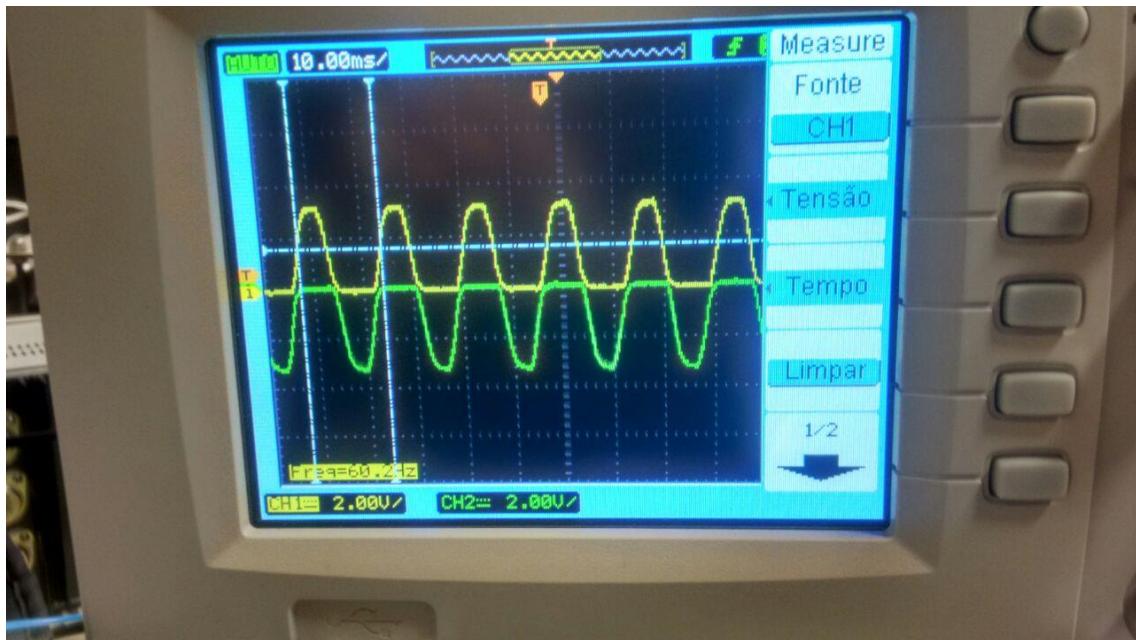
45 Hz:



50 Hz:



60 Hz:



Anexo B

O código:

Botão

```
1
2
3  library ieee;
4  use ieee.std_logic_1164.all;
5  use ieee.std_logic_unsigned.all;
6
7  entity button is
8    port(
9      signal in_button      : in  std_logic;
10     signal clk_button     : in  std_logic;
11     signal out_pin_button : out std_logic
12   );
13 end button;
14
15 architecture logica of button is
16 signal out_button : std_logic := '1';
17 begin
18
19   process(clk_button)
20   begin
21
22     if(rising_edge(clk_button)) then
23       if(in_button = '0') then
24         out_button <= '0';
25       else
26         out_button <= '1';
27       end if;
28     end if;
29
30   end process;
31
32   out_pin_button <= out_button;
33
34 end logica;
```

PWM

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.std_logic_unsigned.all;
4
5 entity pwm is
6 port(
7     signal clk_50MHz : in std_logic;
8     signal pwm_ciclo_ativo_us : in std_logic_vector(7 downto 0);
9     signal pwm_saida : out std_logic
10 );
11 end entity;
12
13 architecture logica of pwm is
14     signal pwm_sinal_interno : std_logic := '0';
15     signal pwm_período_us : std_logic_vector(7 downto 0) := "11001000";
16 begin
17
18     process(clk_50MHz)
19         variable contador_base : integer := 0;
20         variable contador_lus : integer := 0;
21     begin
22
23         if(rising_edge(clk_50MHz)) then
24             contador_base := contador_base + 1;
25             if(contador_base = 50) then
26
27                 contador_lus := contador_lus + 1;
28
29                 if(contador_lus = pwm_período_us) then
30                     contador_lus := 0;
31                 end if;
32
33                 if(contador_lus < pwm_ciclo_ativo_us) then
34                     pwm_sinal_interno <= '1';
35                 else
36                     pwm_sinal_interno <= '0';
37                 end if;
38
39                 contador_base := 0;
40
41             end if;
42         end if;
43     end process;
44
45     pwm_saida <= pwm_sinal_interno;
46
47 end architecture;
```

Controladora

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4
5  entity controladora is
6    port(
7      signal clk_50MHz_top : in std_logic;
8      signal pwm_saida_positivo_top : out std_logic;
9      signal pwm_saida_negativo_top : out std_logic;
10     signal increment_top : in std_logic;
11     signal decrement_top : in std_logic;
12     signal RS_top : out std_logic;
13     signal RW_top : buffer bit;
14     signal E_top : out std_logic;
15     signal DB_top : out std_logic_vector(7 downto 0);
16     signal lcd_on_top : out std_logic;
17     signal lcd_blon_top : out std_logic
18   );
19 end entity;
20
21 architecture logica of controladora is
22
23   signal freq_setada_top : std_logic_vector(3 downto 0);
24   signal ciclo_ativo_positivo_aux : std_logic_vector(7 downto 0);
25   signal ciclo_ativo_negativo_aux : std_logic_vector(7 downto 0);
26   signal estado_pwm_top : std_logic_vector(3 downto 0);
27
28   component pwm
29     port(
30       signal clk_50MHz : in std_logic;
31       signal pwm_ciclo_ativo_us : in std_logic_vector(7 downto 0);
32       signal pwm_saida : out std_logic
33     );
34 end component;
35
36   component controlePwm
37     port(
38       signal estado_pwm : in std_logic_vector(3 downto 0);
39       signal clk_50Mhz : in std_logic;
40       signal ciclo_ativo_senoide_positiva : out std_logic_vector(7 downto 0);
41       signal ciclo_ativo_senoide_negativa : out std_logic_vector(7 downto 0)
42     );
43 end component;
44
45   component set_freq
46     port(
47       signal clk_50_MHz : in std_logic;
48       signal increment : in std_logic;
49       signal decrement : in std_logic;
50       signal freq_setada : out std_logic_vector(3 downto 0);
51       signal clk_button_aux : in std_logic
52     );
53 end component;
54
55   component lcd
56     port(
57       signal clk : in std_logic;
58       signal RS : out std_logic;
59       signal RW : out std_logic;
60       signal E : buffer bit;
61       signal DB : out std_logic_vector(7 downto 0);
62       signal lcd_on : out std_logic;
63       signal lcd_blon : out std_logic;
64       signal freq : in std_logic_vector(3 downto 0)
65     );
66 end component;
67
68 begin
69
```

```

70      pwm_1           : pwm          port map(clk_50MHz => clk_50MHz_top,
71      pwm_ciclo_ativo_us => ciclo_ativo_positivo_aux, pwm_saida =>
72      pwm_saida_positivo_top);
73      pwm_2           : pwm          port map(clk_50MHz => clk_50MHz_top,
74      pwm_ciclo_ativo_us => ciclo_ativo_negativo_aux, pwm_saida =>
75      pwm_saida_negativo_top);
76      controlePwm_1 : controlePwm port map(estado_pwm => freq_setada_top, clk_50MHz
77      => clk_50MHz_top, ciclo_ativo_senoide_positiva => ciclo_ativo_positivo_aux,
78      ciclo_ativo_senoide_negativa => ciclo_ativo_negativo_aux);
79      set_freq_1     : set_freq    port map(clk_50_MHz => clk_50MHz_top, increment =>
80      increment_top, decrement => decrement_top, freq_setada => freq_setada_top,
81      clk_button_aux => clk_50MHz_top);
82      lcd_1          : lcd          port map(clk => clk_50MHz_top, RS => RS_top, RW =>
83      RW_top, E => E_top, DB => DB_top, lcd_on => lcd_on_top, lcd_blon =>
84      lcd_blon_top, freq => freq_setada_top);

end architecture;

```

Controle PWM

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4  use ieee.numeric_std.all;
5
6  entity controlePwm is
7    port(
8      signal estado_pwm           : in std_logic_vector(3 downto 0);
9      signal clk_50Mhz            : in std_logic;
10     signal ciclo_ativo_senoide_positiva : out std_logic_vector(7 downto 0);
11     signal ciclo_ativo_senoide_negativa : out std_logic_vector(7 downto 0)
12   );
13 end entity;
14
15 architecture logica of controlePwm is
16   type seno is array (0 to 45) of std_logic_vector(7 downto 0);
17   signal seno_onda: seno := (
18     "00000000", "00010010", "00100011", "00110101", "01000110", "01010111", "01101000", "01110000",
19     "110000", "10000111", "10010110", "10100100", "10110001", "10111110", "11001001", "11010011",
20     "11101001", "11100101", "11110100", "11110011", "11110111", "11111011", "11111110",
21     "11111111", "11111111", "11111110", "11110111", "11110011", "11101100", "11100101", "11100111",
22     "00101101", "01011101", "01100101", "01101001", "01100011", "010100100", "010100101", "00101011",
23     "00100111", "01111000", "01101000", "01010111", "01001100", "00110101", "00100011", "00010010",
24     "00000000");
25   signal ciclo_ativo_controlado_positivo_aux : std_logic_vector(7 downto 0);
26   signal ciclo_ativo_controlado_negativo_aux : std_logic_vector(7 downto 0);
27 begin
28
29   process(clk_50Mhz, estado_pwm)
30     variable contador_base      : integer := 0;
31     variable contador_lus       : integer := 0;
32     variable posicao_no_vetor_seno : integer := 0;
33     variable seno_positivo_negativo : integer := 0;
34 begin
35
36   if(rising_edge(clk_50Mhz)) then
37
38     contador_base := contador_base + 1;
39     if(contador_base = 50) then
40       contador_lus := contador_lus + 1;
41       contador_base := 0;
42     end if;
43
44     case estado_pwm is
45       when "0000" =>
46         ciclo_ativo_controlado_positivo_aux <= seno_onda(0);
47         ciclo_ativo_controlado_negativo_aux <= seno_onda(0);
48
49       when "0001" =>
50         if(contador_lus >= 2174) then
51           if(posicao_no_vetor_seno < 45) then
52             posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
53           else
54             posicao_no_vetor_seno := 0;
55
56             if(seno_positivo_negativo = 0) then
57               seno_positivo_negativo := 1;
58             else
59               seno_positivo_negativo := 0;
60             end if;
61
62           end if;
63           contador_lus := 0;
64         end if;
65
66       end case;
67
68   end if;
69 end process;
70
71 end architecture;
```

```

62
63
64      if(seno_positivo_negativo == 0) then
65          ciclo_ativo_controlado_negativo_aux <=
66              seno_onda(posicao_no_vetor_seno);
67      else
68          ciclo_ativo_controlado_positivo_aux <=
69              seno_onda(posicao_no_vetor_seno);
70      end if;
71
72      when "0010" =>
73
74          if(contador_lus >= 1087) then
75              if(posicao_no_vetor_seno < 45) then
76                  posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
77              else
78                  posicao_no_vetor_seno := 0;
79
80              if(seno_positivo_negativo == 0) then
81                  seno_positivo_negativo := 1;
82              else
83                  seno_positivo_negativo := 0;
84              end if;
85
86          end if;
87
88          if(seno_positivo_negativo == 0) then
89              ciclo_ativo_controlado_negativo_aux <=
90                  seno_onda(posicao_no_vetor_seno);
91          else
92              ciclo_ativo_controlado_positivo_aux <=
93                  seno_onda(posicao_no_vetor_seno);
94          end if;
95
96      when "0011" =>
97
98          if(contador_lus >= 725) then
99              if(posicao_no_vetor_seno < 45) then
100                  posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
101              else
102                  posicao_no_vetor_seno := 0;
103
104              if(seno_positivo_negativo == 0) then
105                  seno_positivo_negativo := 1;
106              else
107                  seno_positivo_negativo := 0;
108              end if;
109
110          end if;
111
112          if(seno_positivo_negativo == 0) then
113              ciclo_ativo_controlado_negativo_aux <=
114                  seno_onda(posicao_no_vetor_seno);
115
116      when "0100" =>
117
118          if(contador_lus >= 543) then
119              if(posicao_no_vetor_seno < 45) then
120                  posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
121              else
122                  posicao_no_vetor_seno := 0;
123
124

```

```

125         if(seno_positivo_negativo = 0) then
126             seno_positivo_negativo := 1;
127         else
128             seno_positivo_negativo := 0;
129         end if;
130
131     end if;
132     contador_lus := 0;
133 end if;
134
135 if(seno_positivo_negativo = 0) then
136     ciclo_ativo_controlado_negativo_aux <=
137     seno_onda(posicao_no_vetor_seno);
138 else
139     ciclo_ativo_controlado_positivo_aux <=
140     seno_onda(posicao_no_vetor_seno);
141 end if;
142
when "0101" =>
143
144     if(contador_lus >= 435) then
145         if(posicao_no_vetor_seno < 45) then
146             posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
147         else
148             posicao_no_vetor_seno := 0;
149
150         if(seno_positivo_negativo = 0) then
151             seno_positivo_negativo := 1;
152         else
153             seno_positivo_negativo := 0;
154         end if;
155
156         end if;
157         contador_lus := 0;
158     end if;
159
160 if(seno_positivo_negativo = 0) then
161     ciclo_ativo_controlado_negativo_aux <=
162     seno_onda(posicao_no_vetor_seno);
163 else
164     ciclo_ativo_controlado_positivo_aux <=
165     seno_onda(posicao_no_vetor_seno);
166 end if;
167
when "0110" =>
168
169     if(contador_lus >= 362) then
170         if(posicao_no_vetor_seno < 45) then
171             posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
172         else
173             posicao_no_vetor_seno := 0;
174
175         if(seno_positivo_negativo = 0) then
176             seno_positivo_negativo := 1;
177         else
178             seno_positivo_negativo := 0;
179         end if;
180
181         end if;
182         contador_lus := 0;
183     end if;
184
185 if(seno_positivo_negativo = 0) then
186     ciclo_ativo_controlado_negativo_aux <=
187     seno_onda(posicao_no_vetor_seno);

```

```

188
189 when "0111" =>
190
191     if(contador_lus >= 311) then
192         if(posicao_no_vetor_seno < 45) then
193             posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
194         else
195             posicao_no_vetor_seno := 0;
196
197         if(seno_positivo_negativo = 0) then
198             seno_positivo_negativo := 1;
199         else
200             seno_positivo_negativo := 0;
201         end if;
202
203     end if;
204     contador_lus := 0;
205 end if;
206
207 if(seno_positivo_negativo = 0) then
208     ciclo_ativo_controlado_negativo_aux <=
209     seno_onda(posicao_no_vetor_seno);
210 else
211     ciclo_ativo_controlado_positivo_aux <=
212     seno_onda(posicao_no_vetor_seno);
213 end if;
214
215 when "1000" =>
216
217     if(contador_lus >= 272) then
218         if(posicao_no_vetor_seno < 45) then
219             posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
220         else
221             posicao_no_vetor_seno := 0;
222
223         if(seno_positivo_negativo = 0) then
224             seno_positivo_negativo := 1;
225         else
226             seno_positivo_negativo := 0;
227         end if;
228         contador_lus := 0;
229     end if;
230
231     if(seno_positivo_negativo = 0) then
232         ciclo_ativo_controlado_negativo_aux <=
233         seno_onda(posicao_no_vetor_seno);
234 else
235     ciclo_ativo_controlado_positivo_aux <=
236     seno_onda(posicao_no_vetor_seno);
237 end if;
238
239 when "1001" =>
240
241     if(contador_lus >= 242) then
242         if(posicao_no_vetor_seno < 45) then
243             posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
244         else
245             posicao_no_vetor_seno := 0;
246
247         if(seno_positivo_negativo = 0) then
248             seno_positivo_negativo := 1;
249         else
250             seno_positivo_negativo := 0;
251         end if;
252
253     end if;
254     contador_lus := 0;

```

```

253     end if;
254
255     if(seno_positivo_negativo = 0) then
256         ciclo_ativo_controlado_negativo_aux <=
257             seno_onda(posicao_no_vetor_seno);
258     else
259         ciclo_ativo_controlado_positivo_aux <=
260             seno_onda(posicao_no_vetor_seno);
261     end if;
262
263 when "1010" =>
264
265     if(contador_lus >= 217) then
266         if(posicao_no_vetor_seno < 45) then
267             posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
268         else
269             posicao_no_vetor_seno := 0;
270
271         if(seno_positivo_negativo = 0) then
272             seno_positivo_negativo := 1;
273         else
274             seno_positivo_negativo := 0;
275         end if;
276         contador_lus := 0;
277     end if;
278
279     if(seno_positivo_negativo = 0) then
280         ciclo_ativo_controlado_negativo_aux <=
281             seno_onda(posicao_no_vetor_seno);
282     else
283         ciclo_ativo_controlado_positivo_aux <=
284             seno_onda(posicao_no_vetor_seno);
285     end if;
286
287 when "1011" =>
288
289     if(contador_lus >= 198) then
290         if(posicao_no_vetor_seno < 45) then
291             posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
292         else
293             posicao_no_vetor_seno := 0;
294
295         if(seno_positivo_negativo = 0) then
296             seno_positivo_negativo := 1;
297         else
298             seno_positivo_negativo := 0;
299         end if;
300         contador_lus := 0;
301     end if;
302
303     if(seno_positivo_negativo = 0) then
304         ciclo_ativo_controlado_negativo_aux <=
305             seno_onda(posicao_no_vetor_seno);
306     else
307         ciclo_ativo_controlado_positivo_aux <=
308             seno_onda(posicao_no_vetor_seno);
309     end if;
310
311 when "1100" =>
312
313     if(contador_lus >= 181) then
314         if(posicao_no_vetor_seno < 45) then
315             posicao_no_vetor_seno := posicao_no_vetor_seno + 1;

```

```

316
317         if(seno_positivo_negativo = 0) then
318             seno_positivo_negativo := 1;
319         else
320             seno_positivo_negativo := 0;
321         end if;
322
323     end if;
324     contador_lus := 0;
325 end if;
326
327 if(seno_positivo_negativo = 0) then
328     ciclo_ativo_controlado_negativo_aux <=
329     seno_onda(posicao_no_vetor_seno);
330 else
331     ciclo_ativo_controlado_positivo_aux <=
332     seno_onda(posicao_no_vetor_seno);
333 end if;
334
335 when others =>
336
337     if(contador_lus >= 181) then
338         if(posicao_no_vetor_seno < 45) then
339             posicao_no_vetor_seno := posicao_no_vetor_seno + 1;
340         else
341             posicao_no_vetor_seno := 0;
342
343             if(seno_positivo_negativo = 0) then
344                 seno_positivo_negativo := 1;
345             else
346                 seno_positivo_negativo := 0;
347             end if;
348             contador_lus := 0;
349         end if;
350
351         if(seno_positivo_negativo = 0) then
352             ciclo_ativo_controlado_negativo_aux <=
353             seno_onda(posicao_no_vetor_seno);
354         else
355             ciclo_ativo_controlado_positivo_aux <=
356             seno_onda(posicao_no_vetor_seno);
357         end if;
358
359     end case;
360
361 end if;
362
363 end process;
364
365 ciclo_ativo_senoide_positiva <= ciclo_ativo_controlado_positivo_aux;
366 ciclo_ativo_senoide_negativa <= ciclo_ativo_controlado_negativo_aux;
367
368 end architecture;

```

LCD

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.std_logic_unsigned.all;
4
5 entity lcd is
6     generic (fclk: natural := 50_000_000); -- 50MHz , cristal do kit EE03
7     port(
8         signal clk : in std_logic;
9         signal RS : out std_logic;
10        signal RW : out std_logic;
11        signal E : buffer bit;
12        signal DB : out std_logic_vector(7 downto 0);
13        signal lcd_on : out std_logic;
14        signal lcd_blon : out std_logic;
15        signal freq : in std_logic_vector(3 downto 0)
16    );
17
18 end lcd;
19
20 architecture hardware of lcd is
21     type state is (FunctionSet1, FunctionSet2, FunctionSet3,
22
23         FunctionSet4,FunctionSet5,FunctionSet6,FunctionSet7,FunctionSet8,FunctionSet9,FunctionSet10,FunctionSet11,FunctionSet12,
24
25         FunctionSet13,FunctionSet14,FunctionSet15,FunctionSet16,FunctionSet17,FunctionSet18,FunctionSet19,ClearDisplay,
26         DisplayControl, EntryMode,
27         WriteData1, WriteData2, WriteData3, WriteData4,
28         WriteData5,WriteData6,WriteData7,WriteData8,WriteData9,WriteData10,WriteData11,
29
30         WriteData12,WriteData13,WriteData14,WriteData15,WriteData16,WriteData17,WriteData18,WriteData19,SetAddress,SetAddress1, ReturnHome);
31
32     signal pr_state, nx_state: state;
33 begin
34
35     lcd_on <= '1';
36     lcd_blon <='1';
37
38     ----Clock generator (E->500Hz) -----
39     process (clk)
40         variable count: natural range 0 to fclk/1000;
41     begin
42         if (clk' event and clk = '1') then
43             count := count + 1;
44             if (count=fclk/1000) then
45                 E <= not E;
46                 count := 0;
47                 end if;
48             end if;
49         end process;
50
51     -----Lower section of FSM-----
52     process (E)
53     begin
54         if (E' event and E = '1') then
55             -- IF (rst= '0') THEN
56             pr_state <= FunctionSet1;
57             --ELSE
58             pr_state <= nx_state;
59             --END IF;
60             end if;
61         end process;
62
63     -----Upper section of FSX-----
64     process (pr_state)
65     begin
66
67         case freq is

```

```

63
64  -- =====
65  when "0000" =>
66
67      case pr_state is
68
69          when FunctionSet1 =>
70              RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
71              FunctionSet2;
72
73          when FunctionSet2 =>
74              RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
75              FunctionSet3;
76
77          when FunctionSet3 =>
78              RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
79              FunctionSet4;
80
81          when FunctionSet4 =>
82              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
83              FunctionSet5;
84
85          when FunctionSet5 =>
86              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
87              FunctionSet6;
88
89          when FunctionSet6 =>
90              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
91              FunctionSet7;
92
93          when FunctionSet7 =>
94              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
95              FunctionSet8;
96
97          when FunctionSet8 =>
98              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
99              FunctionSet9;
100
101         when FunctionSet9 =>
102             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
103             FunctionSet10;
104
105         when FunctionSet10 =>
106             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
107             FunctionSet11;
108
109         when FunctionSet11 =>
110             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
111             FunctionSet12;
112
113         when FunctionSet12 =>
114             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
115             FunctionSet13;
116
117         when FunctionSet13 =>
118             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=

```

```

119
120
121      when FunctionSet18 =>
122          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
123          FunctionSet19;
124
125      when FunctionSet19 =>
126          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
127          ClearDisplay ;
128
129      when ClearDisplay =>
130          RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
131          DisplayControl;
132
133      when DisplayControl =>
134          RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
135          EntryMode;
136
137      when EntryMode =>
138          RS<= '0'; RW<= '0'; DB <= "00000110"; nx_state <=
139          WriteData1;
140
141      when WriteData1 =>
142          RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
143          SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
144
145      when SetAddress1 =>
146          RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
147          WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
148          0 COLUNA 3
149
150      when WriteData2 =>
151          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
152          WriteData3; --'F'
153
154      when WriteData3 =>
155          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
156          WriteData4; --'r'
157
158      when WriteData4 =>
159          RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
160          WriteData5; --'e'
161
162      when WriteData5 =>
163          RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
164          WriteData6; --'q'
165
166      when WriteData6 =>
167          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
168          WriteData7; --':'
169
170      when WriteData7 =>
171          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
172          WriteData8; --' '
173
174      when WriteData8 =>
175          RS <= '1'; RW <= '0'; DB <= X"30"; nx_state <=
176          WriteData9; --'0'
177
178      when WriteData9 =>
179          RS <= '1'; RW <= '0'; DB <= X"30"; nx_state <=
180          WriteData10; --'0'
181
182      when WriteData10 =>
183          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
184          WriteData11; --' '
185
186      when WriteData11 =>
187          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
188          WriteData12; --'H'
189
190      when WriteData12 =>
191          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
192          ReturnHome; -- 'z'

```

```

173
174
175      when ReturnHome =>
176          RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
177          WriteData1;
178
179      when others =>
180          RS <= '0'; RW <= '0';
181
182      end case;
183
184      =====
185      when "0001" =>
186
187          case pr_state is
188
189              when FunctionSet1 =>
190                  RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
191                  FunctionSet2;
192
193              when FunctionSet2 =>
194                  RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
195                  FunctionSet3;
196
197              when FunctionSet3 =>
198                  RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
199                  FunctionSet4;
200
201              when FunctionSet4 =>
202                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
203                  FunctionSet5;
204
205              when FunctionSet5 =>
206                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
207                  FunctionSet6;
208
209              when FunctionSet6 =>
210                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
211                  FunctionSet7;
212
213              when FunctionSet7 =>
214                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
215                  FunctionSet8;
216
217              when FunctionSet8 =>
218                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
219                  FunctionSet9;
220
221              when FunctionSet9 =>
222                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
223                  FunctionSet10;
224
225              when FunctionSet10 =>
226                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
227                  FunctionSet11;
228
229              when FunctionSet11 =>
230                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
231                  FunctionSet12;
232
233              when FunctionSet12 =>
234                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
235                  FunctionSet13;
236
237              when FunctionSet13 =>
238                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
239                  FunctionSet14;
240
241              when FunctionSet14 =>
242                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
243                  FunctionSet15;
244
245              when FunctionSet15 =>
246                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
247                  FunctionSet16;

```

```

230
231      when FunctionSet16 =>
232          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
233          FunctionSet17;
234
235      when FunctionSet17 =>
236          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
237          FunctionSet18;
238
239      when FunctionSet18 =>
240          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
241          FunctionSet19;
242
243      when FunctionSet19 =>
244          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
245          ClearDisplay ;
246
247      when ClearDisplay =>
248          RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
249          DisplayControl;
250
251      when DisplayControl =>
252          RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
253          EntryMode;
254
255      when EntryMode =>
256          RS<= '0'; RW<= '0'; DB <= "00000110"; nx_state <=
257          WriteData1;
258
259      when WriteData1 =>
260          RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
261          SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
262
263      when SetAddress1 =>
264          RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
265          WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
266          0 COLUNA 3
267
268      when WriteData2 =>
269          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
270          WriteData3; --'F'
271
272      when WriteData3 =>
273          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
274          WriteData4; --'r'
275
276      when WriteData4 =>
277          RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
278          WriteData5; --'e'
279
280      when WriteData5 =>
281          RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
282          WriteData6; --'q'
283
284      when WriteData6 =>
285          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
286          WriteData7; --':'
287
288      when WriteData7 =>
289          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
290          WriteData8; --' '
291
292      when WriteData8 =>
293          RS <= '1'; RW <= '0'; DB <= X"30"; nx_state <=
294          WriteData9; --'0'
295
296      when WriteData9 =>
297          RS <= '1'; RW <= '0'; DB <= X"35"; nx_state <=
298          WriteData10; --'5'
299
300      when WriteData10 =>
301          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
302          WriteData11; --' '

```

```

284
285      when WriteData11 =>
286          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
287          WriteData12; --'H'
288
289      when WriteData12 =>
290          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
291          ReturnHome; --'Z'
292
293      when ReturnHome =>
294          RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
295          WriteData1;
296
297      when others =>
298          RS <= '0'; RW <= '0';
299
300  end case;
301
302  =====
303  when "0010" =>
304
305      case pr_state is
306
307          when FunctionSet1 =>
308              RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
309              FunctionSet2;
310
311          when FunctionSet2 =>
312              RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
313              FunctionSet3;
314
315          when FunctionSet3 =>
316              RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
317              FunctionSet4;
318
319          when FunctionSet4 =>
320              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
321              FunctionSet5;
322
323          when FunctionSet5 =>
324              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
325              FunctionSet6;
326
327          when FunctionSet6 =>
328              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
329              FunctionSet7;
330
331          when FunctionSet7 =>
332              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
333              FunctionSet8;
334
335          when FunctionSet8 =>
336              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
337              FunctionSet9;
338
339          when FunctionSet9 =>
340              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
341              FunctionSet10;

```

```

342             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
343             FunctionSet14;
344
345         when FunctionSet14 =>
346             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
347             FunctionSet15;
348
349         when FunctionSet15 =>
350             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
351             FunctionSet16;
352
353         when FunctionSet16 =>
354             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
355             FunctionSet17;
356
357         when FunctionSet17 =>
358             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
359             FunctionSet18;
360
361         when FunctionSet18 =>
362             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
363             FunctionSet19;
364
365         when FunctionSet19 =>
366             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
367             ClearDisplay ;
368
369         when ClearDisplay =>
370             RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
371             DisplayControl;
372
373         when DisplayControl =>
374             RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
375             EntryMode;
376
377         when EntryMode =>
378             RS<= '0'; RW<= '0'; DB <= "00000110"; nx_state <=
379             WriteData1;
380
381         when WriteData1 =>
382             RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
383             SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
384
385         when SetAddress1 =>
386             RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
387             WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
388             0 COLUNA 3
389
390         when WriteData2 =>
391             RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
392             WriteData3; --'F'
393
394         when WriteData3 =>
395             RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
396             WriteData4; --'r'
397
398         when WriteData4 =>
399             RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
400             WriteData5; --'e'
401
402         when WriteData5 =>
403             RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
404             WriteData6; --'q'
405
406         when WriteData6 =>
407             RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
408             WriteData7; --':'
409
410         when WriteData7 =>
411             RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
412             WriteData8; --' '
413
414         when WriteData8 =>

```

```

396           RS <= '1'; RW <= '0'; DB <= X"31"; nx_state <=
397           WriteData9; --'1'
398
399       when WriteData9 =>
400           RS <= '1'; RW <= '0'; DB <= X"30"; nx_state <=
401           WriteData10; --'0'
402
403       when WriteData10 =>
404           RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
405           WriteData11; --'1'
406
407       when WriteData11 =>
408           RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
409           WriteData12; --'H'
410
411       when WriteData12 =>
412           RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
413           ReturnHome; -- 'z'
414
415       when others =>
416           RS <= '0'; RW <= '0';
417
418   end case;
419
420   -- =====
421   when "0011" =>
422
423       case pr_state is
424
425           when FunctionSet1 =>
426               RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
427               FunctionSet2;
428
429           when FunctionSet2 =>
430               RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
431               FunctionSet3;
432
433           when FunctionSet3 =>
434               RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
435               FunctionSet4;
436
437           when FunctionSet4 =>
438               RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
439               FunctionSet5;
440
441           when FunctionSet5 =>
442               RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
443               FunctionSet6;
444
445           when FunctionSet6 =>
446               RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
447               FunctionSet7;
448
449           when FunctionSet7 =>
450               RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
451               FunctionSet8;
452

```

```

453
454      when FunctionSet11 =>
455          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
456          FunctionSet12;
457
458      when FunctionSet12 =>
459          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
460          FunctionSet13;
461
462      when FunctionSet13 =>
463          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
464          FunctionSet14;
465
466      when FunctionSet14 =>
467          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
468          FunctionSet15;
469
470      when FunctionSet15 =>
471          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
472          FunctionSet16;
473
474      when FunctionSet16 =>
475          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
476          FunctionSet17;
477
478      when FunctionSet17 =>
479          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
480          FunctionSet18;
481
482      when FunctionSet18 =>
483          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
484          FunctionSet19;
485
486      when FunctionSet19 =>
487          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
488          ClearDisplay;
489
490      when ClearDisplay =>
491          RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
492          DisplayControl;
493
494      when DisplayControl =>
495          RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
496          EntryMode;
497
498      when EntryMode =>
499          RS <= '0'; RW <= '0'; DB <= "00000110"; nx_state <=
500          WriteData1;
501
502      when WriteData1 =>
503          RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
504          SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
505
506      when SetAddress1 =>
507          RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
508          WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
509          0 COLUNA 3;
510
511      when WriteData2 =>
512          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
513          WriteData3; --'F'
514
515      when WriteData3 =>
516          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
517          WriteData4; --'r'
518
519      when WriteData4 =>
520          RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
521          WriteData5; --'e'
522
523      when WriteData5 =>
524          RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
525          WriteData6; --'q'

```

```

507
508      when WriteData6 =>
509          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
510          WriteData7; --'1'
511
512      when WriteData7 =>
513          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
514          WriteData8; --'1'
515
516      when WriteData8 =>
517          RS <= '1'; RW <= '0'; DB <= X"31"; nx_state <=
518          WriteData9; --'1'
519
520      when WriteData9 =>
521          RS <= '1'; RW <= '0'; DB <= X"35"; nx_state <=
522          WriteData10; --'5'
523
524      when WriteData10 =>
525          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
526          WriteData11; --'1'
527
528      when WriteData11 =>
529          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
530          WriteData12; --'H'
531
532      when WriteData12 =>
533          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
534          ReturnHome; --'z'
535
536      when ReturnHome =>
537          RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
538          WriteData1;
539
540      when others =>
541          RS <= '0'; RW <= '0';
542
543      end case;
544
545      --
546      =====
547
548      when "0100" =>
549
550          case pr_state is
551
552              when FunctionSet1 =>
553                  RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
554                  FunctionSet2;
555
556              when FunctionSet2 =>
557                  RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
558                  FunctionSet3;
559
560              when FunctionSet3 =>
561                  RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
562                  FunctionSet4;
563
564              when FunctionSet4 =>
565                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
566                  FunctionSet5;
567
568              when FunctionSet5 =>
569                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
570                  FunctionSet6;
571
572              when FunctionSet6 =>
573                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
574                  FunctionSet7;
575
576              when FunctionSet7 =>
577                  RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
578                  FunctionSet8;
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762

```

```

563
564      when FunctionSet8 =>
565          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
566              FunctionSet9;
567
568      when FunctionSet9 =>
569          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
570              FunctionSet10;
571
572      when FunctionSet10 =>
573          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
574              FunctionSet11;
575
576      when FunctionSet11 =>
577          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
578              FunctionSet12;
579
580      when FunctionSet12 =>
581          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
582              FunctionSet13;
583
584      when FunctionSet13 =>
585          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
586              FunctionSet14;
587
588      when FunctionSet14 =>
589          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
590              FunctionSet15;
591
592      when FunctionSet15 =>
593          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
594              FunctionSet16;
595
596      when FunctionSet16 =>
597          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
598              FunctionSet17;
599
600      when FunctionSet17 =>
601          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
602              FunctionSet18;
603
604      when FunctionSet18 =>
605          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
606              FunctionSet19;
607
608      when FunctionSet19 =>
609          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
610              ClearDisplay;
611
612      when ClearDisplay =>
613          RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
614              DisplayControl;
615
616      when DisplayControl =>
617          RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
618              EntryMode;
619
620      when EntryMode =>
621          RS <= '0'; RW <= '0'; DB <= "00000110"; nx_state <=
622              WriteData1;
623
624      when WriteData1 =>
625          RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
626              SetAddress1; --ESCREVE UM ESPAÇO EM BRANCO
627
628      when SetAddress1 =>
629          RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
630              WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
631                  0 COLUNA 3;
632
633      when WriteData2 =>
634          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
635              WriteData3; --'F'

```

```

617      when WriteData3 =>
618          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
619              WriteData4; --'r'
620
621      when WriteData4 =>
622          RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
623              WriteData5; --'e'
624
625      when WriteData5 =>
626          RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
627              WriteData6; --'q'
628
629      when WriteData6 =>
630          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
631              WriteData7; --':'
632
633      when WriteData7 =>
634          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
635              WriteData8; --' '
636
637      when WriteData8 =>
638          RS <= '1'; RW <= '0'; DB <= X"32"; nx_state <=
639              WriteData9; --'2'
640
641      when WriteData9 =>
642          RS <= '1'; RW <= '0'; DB <= X"30"; nx_state <=
643              WriteData10; --'0'
644
645      when WriteData10 =>
646          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
647              WriteData11; --' '
648
649      when WriteData11 =>
650          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
651              WriteData12; --'H'
652
653      when WriteData12 =>
654          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
655              ReturnHome; --'z'
656
657      when others =>
658          RS <= '0'; RW <= '0';
659
660  end case;
661
662  ==
663
664
665
666
667
668
669
670
671
672

```

```

673
674      when FunctionSet5 =>
675          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
676          FunctionSet6;
677
678      when FunctionSet6 =>
679          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
680          FunctionSet7;
681
682      when FunctionSet7 =>
683          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
684          FunctionSet8;
685
686      when FunctionSet8 =>
687          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
688          FunctionSet9;
689
690      when FunctionSet9 =>
691          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
692          FunctionSet10;
693
694      when FunctionSet10 =>
695          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
696          FunctionSet11;
697
698      when FunctionSet11 =>
699          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
700          FunctionSet12;
701
702      when FunctionSet12 =>
703          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
704          FunctionSet13;
705
706      when FunctionSet13 =>
707          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
708          FunctionSet14;
709
710      when FunctionSet14 =>
711          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
712          FunctionSet15;
713
714      when FunctionSet15 =>
715          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
716          FunctionSet16;
717
718      when FunctionSet16 =>
719          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
720          FunctionSet17;
721
722      when FunctionSet17 =>
723          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
724          FunctionSet18;
725
726      when FunctionSet18 =>
727          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
          ClearDisplay ;

```

```

728             RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
729             SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
730
731     when SetAddress1 =>
732         RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
733         WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
734         0 COLUNA 3
735
736     when WriteData2 =>
737         RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
738         WriteData3; --'F'
739
737     when WriteData3 =>
740         RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
741         WriteData4; --'r'
742
743     when WriteData4 =>
744         RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
745         WriteData5; --'e'
746
747     when WriteData5 =>
748         RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
749         WriteData6; --'q'
750
751     when WriteData6 =>
752         RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
753         WriteData7; --':'
754
755     when WriteData7 =>
756         RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
757         WriteData8; --' '
758
759     when WriteData8 =>
760         RS <= '1'; RW <= '0'; DB <= X"32"; nx_state <=
761         WriteData9; --'2'
762
763     when WriteData9 =>
764         RS <= '1'; RW <= '0'; DB <= X"35"; nx_state <=
765         WriteData10; --'5'
766
767     when WriteData10 =>
768         RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
769         WriteData11; --' '
770
771     when WriteData11 =>
772         RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
773         WriteData12; --'H'
774
775     when WriteData12 =>
776         RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
777         ReturnHome; --'z'
778
779     when ReturnHome =>
780         RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
781         WriteData1;
782
783     when others =>
784         RS <= '0'; RW <= '0';
785
786 end case;
787
788 -- =====
789 when "0110" =>
790
791     case pr_state is
792
793         when FunctionSet1 =>
794             RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
795             FunctionSet2;
796
797         when FunctionSet2 =>
798             RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=

```

```

        FunctionSet3;

785  when FunctionSet3 =>
786      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
787          FunctionSet4;
788
789  when FunctionSet4 =>
790      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
791          FunctionSet5;
792
793  when FunctionSet5 =>
794      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
795          FunctionSet6;
796
797  when FunctionSet6 =>
798      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
799          FunctionSet7;
800
801  when FunctionSet7 =>
802      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
803          FunctionSet8;
804
805  when FunctionSet8 =>
806      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
807          FunctionSet9;
808
809  when FunctionSet9 =>
810      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
811          FunctionSet10;
812
813  when FunctionSet10 =>
814      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
815          FunctionSet11;
816
817  when FunctionSet11 =>
818      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
819          FunctionSet12;
820
821  when FunctionSet12 =>
822      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
823          FunctionSet13;
824
825  when FunctionSet13 =>
826      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
827          FunctionSet14;
828
829  when FunctionSet14 =>
830      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
831          FunctionSet15;
832
833  when FunctionSet15 =>
834      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
835          FunctionSet16;
836
837  when FunctionSet16 =>
838      RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
        ClearDisplay;
839
840  when ClearDisplay =>
841      RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
842          DisplayControl;

```

```

839
840
841      when DisplayControl =>
842          RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
843              EntryMode;
844
845      when EntryMode =>
846          RS<= '0'; RW<= '0'; DB <= "00000110"; nx_state <=
847              WriteData1;
848
849      when WriteData1 =>
850          RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
851              SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
852
853      when SetAddress1 =>
854          RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
855              WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
856              0 COLUNA 3
857
858      when WriteData2 =>
859          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
860              WriteData3; --'F'
861
862      when WriteData3 =>
863          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
864              WriteData4; --'r'
865
866      when WriteData4 =>
867          RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
868              WriteData5; --'e'
869
870      when WriteData5 =>
871          RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
872              WriteData6; --'q'
873
874      when WriteData6 =>
875          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
876              WriteData7; --':'
877
878      when WriteData7 =>
879          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
880              WriteData8; --' '
881
882      when WriteData8 =>
883          RS <= '1'; RW <= '0'; DB <= X"33"; nx_state <=
884              WriteData9; --'3'
885
886      when WriteData9 =>
887          RS <= '1'; RW <= '0'; DB <= X"30"; nx_state <=
888              WriteData10; --'0'
889
890      when WriteData10 =>
891          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
892              WriteData11; --' '
893
894      when WriteData11 =>
895          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
896              WriteData12; --'H'
897
898      when WriteData12 =>
899          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
900              ReturnHome; -- 'z'
901
902      when ReturnHome =>
903          RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
904              WriteData1;
905
906      when others =>
907          RS <= '0'; RW <= '0';
908
909      end case;
910
911      -----
912      when "0111" =>

```

```

895
896         case pr_state is
897
898             when FunctionSet1 =>
899                 RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
900                 FunctionSet2;
901
902             when FunctionSet2 =>
903                 RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
904                 FunctionSet3;
905
906             when FunctionSet3 =>
907                 RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
908                 FunctionSet4;
909
910             when FunctionSet4 =>
911                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
912                 FunctionSet5;
913
914             when FunctionSet5 =>
915                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
916                 FunctionSet6;
917
918             when FunctionSet6 =>
919                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
920                 FunctionSet7;
921
922             when FunctionSet7 =>
923                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
924                 FunctionSet8;
925
926             when FunctionSet8 =>
927                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
928                 FunctionSet9;
929
930             when FunctionSet9 =>
931                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
932                 FunctionSet10;
933
934             when FunctionSet10 =>
935                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
936                 FunctionSet11;
937
938             when FunctionSet11 =>
939                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
940                 FunctionSet12;
941
942             when FunctionSet12 =>
943                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
944                 FunctionSet13;
945
946             when FunctionSet13 =>
947                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
948                 FunctionSet14;
949
950             when FunctionSet14 =>
951                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
952                 FunctionSet15;
953
954             when FunctionSet15 =>
955                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
956                 FunctionSet16;
957
958             when FunctionSet16 =>
959                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
960                 FunctionSet17;
961
962             when FunctionSet17 =>
963                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
964                 FunctionSet18;
965
966             when FunctionSet18 =>
967                 RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=

```

```

951                               FunctionSet19;
952
953      when FunctionSet19 =>
954          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
955          ClearDisplay ;
956
957      when ClearDisplay =>
958          RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
959          DisplayControl;
960
961      when DisplayControl =>
962          RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
963          EntryMode;
964
965      when EntryMode =>
966          RS <= '0'; RW <= '0'; DB <= "00000110"; nx_state <=
967          WriteData1;
968
969      when WriteData1 =>
970          RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
971          SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
972
973      when SetAddress1 =>
974          RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
975          WriteData2; --'COMANDO PARA POSICIONAR O CURSOR NA LINHA
976          0 COLUNA 3
977
978      when WriteData2 =>
979          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
980          WriteData3; --'F'
981
982      when WriteData3 =>
983          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
984          WriteData4; --'r'
985
986      when WriteData4 =>
987          RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
988          WriteData5; --'e'
989
990      when WriteData5 =>
991          RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
992          WriteData6; --'q'
993
994      when WriteData6 =>
995          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
996          WriteData7; --':'
997
998      when WriteData7 =>
999          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1000         WriteData8; --' '
1001
1002      when WriteData8 =>
1003          RS <= '1'; RW <= '0'; DB <= X"33"; nx_state <=
1004          WriteData9; --'3'
1005
1006      when WriteData9 =>
1007          RS <= '1'; RW <= '0'; DB <= X"35"; nx_state <=
1008          WriteData10; --'5'
1009
1010      when WriteData10 =>
1011          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1012          WriteData11; --' '
1013
1014      when WriteData11 =>
1015          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
1016          WriteData12; --'H'
1017
1018      when WriteData12 =>
1019          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
1020          ReturnHome; --'z'
1021
1022      when ReturnHome =>
1023          RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
1024

```

```

1005                               WriteData1;
1006
1007      when others =>
1008          RS <= '0'; RW <= '0';
1009
1010     end case;
1011
1012 -- =====
1013 when "1000" =>
1014
1015     case pr_state is
1016
1017         when FunctionSet1 =>
1018             RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
1019             FunctionSet2;
1020
1021         when FunctionSet2 =>
1022             RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
1023             FunctionSet3;
1024
1025         when FunctionSet3 =>
1026             RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
1027             FunctionSet4;
1028
1029         when FunctionSet4 =>
1030             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1031             FunctionSet5;
1032
1033         when FunctionSet5 =>
1034             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1035             FunctionSet6;
1036
1037         when FunctionSet6 =>
1038             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1039             FunctionSet7;
1040
1041         when FunctionSet7 =>
1042             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1043             FunctionSet8;
1044
1045         when FunctionSet8 =>
1046             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1047             FunctionSet9;
1048
1049         when FunctionSet9 =>
1050             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1051             FunctionSet10;
1052
1053         when FunctionSet10 =>
1054             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1055             FunctionSet11;
1056
1057         when FunctionSet11 =>
1058             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1059             FunctionSet12;
1060
1061         when FunctionSet12 =>
1062             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1063             FunctionSet13;
1064
1065         when FunctionSet13 =>
1066             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1067             FunctionSet14;
1068
1069         when FunctionSet14 =>
1070             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1071             FunctionSet15;
1072
1073         when FunctionSet15 =>
1074             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1075             FunctionSet16;

```

```

1062      when FunctionSet16 =>
1063          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1064              FunctionSet17;
1065
1066      when FunctionSet17 =>
1067          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1068              FunctionSet18;
1069
1070      when FunctionSet18 =>
1071          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1072              FunctionSet19;
1073
1074      when FunctionSet19 =>
1075          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1076              ClearDisplay ;
1077
1078      when ClearDisplay =>
1079          RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
1080              DisplayControl;
1081
1082      when DisplayControl =>
1083          RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
1084              EntryMode;
1085
1086      when EntryMode =>
1087          RS<= '0'; RW<= '0'; DB <= "00000110"; nx_state <=
1088          WriteData1;
1089
1090      when WriteData1 =>
1091          RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
1092          SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
1093
1094      when SetAddress1 =>
1095          RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
1096          WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
1097          0 COLUNA 3
1098
1099      when WriteData2 =>
1100          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
1101          WriteData3; --'F'
1102
1103      when WriteData3 =>
1104          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
1105          WriteData4; --'r'
1106
1107      when WriteData4 =>
1108          RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
1109          WriteData5; --'e'
1110
1111      when WriteData5 =>
1112          RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
1113          WriteData6; --'q'
1114
1115      when WriteData6 =>
1116          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
1117          WriteData7; --';'
1118
1119      when WriteData7 =>
1120          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1121          WriteData8; --'
1122
1123      when WriteData8 =>
1124          RS <= '1'; RW <= '0'; DB <= X"34"; nx_state <=
1125          WriteData9; --'4'
1126
1127      when WriteData9 =>
1128          RS <= '1'; RW <= '0'; DB <= X"30"; nx_state <=
1129          WriteData10; --'0'
1130
1131      when WriteData10 =>
1132          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1133          WriteData11; --'

```

```

1116      when WriteData11 =>
1117          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
1118          WriteData12; --'H'
1119
1120      when WriteData12 =>
1121          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
1122          ReturnHome; -- 'z'
1123
1124      when ReturnHome =>
1125          RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
1126          WriteData1;
1127
1128      when others =>
1129          RS <= '0'; RW <= '0';
1130
1131  end case;
1132
1133  -- =====
1134  when "1001" =>
1135
1136      case pr_state is
1137
1138          when FunctionSet1 =>
1139              RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
1140              FunctionSet2;
1141
1142          when FunctionSet2 =>
1143              RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
1144              FunctionSet3;
1145
1146          when FunctionSet3 =>
1147              RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
1148              FunctionSet4;
1149
1150          when FunctionSet4 =>
1151              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1152              FunctionSet5;
1153
1154          when FunctionSet5 =>
1155              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1156              FunctionSet6;
1157
1158          when FunctionSet6 =>
1159              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1160              FunctionSet7;
1161
1162          when FunctionSet7 =>
1163              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1164              FunctionSet8;
1165
1166          when FunctionSet8 =>
1167              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1168              FunctionSet9;
1169
1170          when FunctionSet9 =>
1171              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1172              FunctionSet10;
1173

```

```

1174                               FunctionSet14;
1175
1176      when FunctionSet14 =>
1177          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1178          FunctionSet15;
1179
1180      when FunctionSet15 =>
1181          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1182          FunctionSet16;
1183
1184      when FunctionSet16 =>
1185          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1186          FunctionSet17;
1187
1188      when FunctionSet17 =>
1189          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1190          FunctionSet18;
1191
1192      when FunctionSet18 =>
1193          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1194          FunctionSet19;
1195
1196      when FunctionSet19 =>
1197          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1198          ClearDisplay ;
1199
1200      when ClearDisplay =>
1201          RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
1202          DisplayControl;
1203
1204      when DisplayControl =>
1205          RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
1206          EntryMode;
1207
1208      when EntryMode =>
1209          RS <= '0'; RW <= '0'; DB <= "00000110"; nx_state <=
1210          WriteData1;
1211
1212      when WriteData1 =>
1213          RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
1214          SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
1215
1216      when SetAddress1 =>
1217          RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
1218          WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
1219          0 COLUNA 3
1220
1221      when WriteData2 =>
1222          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
1223          WriteData3; --'F'
1224
1225      when WriteData3 =>
1226          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
1227          WriteData4; --'r'
1228
1229      when WriteData4 =>
1230          RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
1231          WriteData5; --'e'
1232
1233      when WriteData5 =>
1234          RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
1235          WriteData6; --'q'
1236
1237      when WriteData6 =>
1238          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
1239          WriteData7; --':'
1240
1241      when WriteData7 =>
1242          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1243          WriteData8; --' '
1244
1245      when WriteData8 =>
1246          RS <= '1'; RW <= '0'; DB <= X"34"; nx_state <=

```

```

1228                               WriteData9; --'4'
1229
1230      when WriteData9 =>
1231          RS <= '1'; RW <= '0'; DB <= X"35"; nx_state <=
1232          WriteData10; --'5'
1233
1234      when WriteData10 =>
1235          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1236          WriteData11; --' '
1237
1238      when WriteData11 =>
1239          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
1240          WriteData12; --'H'
1241
1242      when WriteData12 =>
1243          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
1244          ReturnHome; -- 'z'
1245
1246      when others =>
1247          RS <= '0'; RW <= '0';
1248
1249  end case;

1250
1251  -- =====
1252  when "1010" =>
1253
1254      case pr_state is
1255
1256          when FunctionSet1 =>
1257              RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
1258              FunctionSet2;
1259
1260          when FunctionSet2 =>
1261              RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
1262              FunctionSet3;
1263
1264          when FunctionSet3 =>
1265              RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
1266              FunctionSet4;
1267
1268          when FunctionSet4 =>
1269              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1270              FunctionSet5;
1271
1272          when FunctionSet5 =>
1273              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1274              FunctionSet6;
1275
1276          when FunctionSet6 =>
1277              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1278              FunctionSet7;
1279
1280          when FunctionSet7 =>
1281              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1282              FunctionSet8;
1283
1284          when FunctionSet8 =>
1285              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1286              FunctionSet9;
1287
1288          when FunctionSet9 =>
1289              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1290              FunctionSet10;
1291
1292          when FunctionSet10 =>
1293              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1294              FunctionSet11;

```

```

1285     when FunctionSet11 =>
1286         RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1287             FunctionSet12;
1288
1289     when FunctionSet12 =>
1290         RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1291             FunctionSet13;
1292
1293     when FunctionSet13 =>
1294         RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1295             FunctionSet14;
1296
1297     when FunctionSet14 =>
1298         RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1299             FunctionSet15;
1300
1301     when FunctionSet15 =>
1302         RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1303             FunctionSet16;
1304
1305     when FunctionSet16 =>
1306         RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1307             FunctionSet17;
1308
1309     when FunctionSet17 =>
1310         RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1311             FunctionSet18;
1312
1313     when FunctionSet18 =>
1314         RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1315             FunctionSet19;
1316
1317     when FunctionSet19 =>
1318         RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1319             ClearDisplay;
1320
1321     when ClearDisplay =>
1322         RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
1323             DisplayControl;
1324
1325     when DisplayControl =>
1326         RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
1327             EntryMode;
1328
1329     when EntryMode =>
1330         RS <= '0'; RW <= '0'; DB <= "00000110"; nx_state <=
1331             WriteData1;
1332
1333     when WriteData1 =>
1334         RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
1335             SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
1336
1337     when SetAddress1 =>
1338         RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
1339             WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
1340                 0 COLUNA 3
1341
1342     when WriteData2 =>
1343         RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
1344             WriteData3; --'F'
1345
1346     when WriteData3 =>
1347         RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
1348             WriteData4; --'r'
1349
1350     when WriteData4 =>
1351         RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
1352             WriteData5; --'e'
1353
1354     when WriteData5 =>
1355         RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
1356             WriteData6; --'q'
1357
1358

```

```

1339      when WriteData6 =>
1340          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
1341              WriteData7; --'1'
1342
1343      when WriteData7 =>
1344          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1345              WriteData8; --'1'
1346
1347      when WriteData8 =>
1348          RS <= '1'; RW <= '0'; DB <= X"35"; nx_state <=
1349              WriteData9; --'5'
1350
1351      when WriteData9 =>
1352          RS <= '1'; RW <= '0'; DB <= X"30"; nx_state <=
1353              WriteData10; --'0'
1354
1355      when WriteData10 =>
1356          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1357              WriteData11; --'1'
1358
1359      when WriteData11 =>
1360          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
1361              WriteData12; --'H'
1362
1363      when WriteData12 =>
1364          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
1365          ReturnHome; --'z'
1366
1367      when ReturnHome =>
1368          RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
1369          WriteData1;
1370
1371      when others =>
1372          RS <= '0'; RW <= '0';
1373
1374  end case;

1375
1376  -- =====
1377  when "1011" =>
1378
1379      case pr_state is
1380
1381          when FunctionSet1 =>
1382              RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
1383                  FunctionSet2;
1384
1385          when FunctionSet2 =>
1386              RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
1387                  FunctionSet3;
1388
1389          when FunctionSet3 =>
1390              RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
1391                  FunctionSet4;
1392
1393          when FunctionSet4 =>
1394              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1395                  FunctionSet5;
1396
1397          when FunctionSet5 =>
1398              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1399                  FunctionSet6;
1400
1401          when FunctionSet6 =>
1402              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1403                  FunctionSet7;
1404
1405          when FunctionSet7 =>
1406              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1407                  FunctionSet8;
1408
1409          when FunctionSet8 =>
1410              RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=

```

```

1397                               FunctionSet9;
1398
1399      when FunctionSet9 =>
1400          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1401              FunctionSet10;
1402
1403      when FunctionSet10 =>
1404          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1405              FunctionSet11;
1406
1407      when FunctionSet11 =>
1408          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1409              FunctionSet12;
1410
1411      when FunctionSet12 =>
1412          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1413              FunctionSet13;
1414
1415      when FunctionSet13 =>
1416          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1417              FunctionSet14;
1418
1419      when FunctionSet14 =>
1420          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1421              FunctionSet15;
1422
1423      when FunctionSet15 =>
1424          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1425              FunctionSet16;
1426
1427      when FunctionSet16 =>
1428          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1429              FunctionSet17;
1430
1431      when FunctionSet17 =>
1432          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1433              FunctionSet18;
1434
1435      when FunctionSet18 =>
1436          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1437              FunctionSet19;
1438
1439      when FunctionSet19 =>
1440          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1441              ClearDisplay;
1442
1443      when ClearDisplay =>
1444          RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
1445              DisplayControl;
1446
1447      when DisplayControl =>
1448          RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
1449              EntryMode;
1450
1451      when EntryMode =>
1452          RS <= '0'; RW <= '0'; DB <= "00000110"; nx_state <=
1453              WriteData1;
1454
1455      when WriteData1 =>
1456          RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
1457              SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
1458
1459      when SetAddress1 =>
1460          RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
1461              WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
1462                  0 COLUNA 3;
1463
1464      when WriteData2 =>
1465          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
1466              WriteData3; --'F';
1467
1468      when WriteData3 =>
1469          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
1470

```

```

1451                               WriteData4; --'r'
1452
1453 when WriteData4 =>
1454     RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
1455     WriteData5; --'e'
1456
1457 when WriteData5 =>
1458     RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
1459     WriteData6; --'q'
1460
1461 when WriteData6 =>
1462     RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
1463     WriteData7; --':'
1464
1465 when WriteData7 =>
1466     RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1467     WriteData8; --' '
1468
1469 when WriteData8 =>
1470     RS <= '1'; RW <= '0'; DB <= X"35"; nx_state <=
1471     WriteData9; --'5'
1472
1473 when WriteData9 =>
1474     RS <= '1'; RW <= '0'; DB <= X"35"; nx_state <=
1475     WriteData10; --'5'
1476
1477 when WriteData10 =>
1478     RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1479     WriteData11; --' '
1480
1481 when WriteData11 =>
1482     RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
1483     WriteData12; --'H'
1484
1485 when WriteData12 =>
1486     RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
1487     ReturnHome; -- 'z'
1488
1489 when ReturnHome =>
1490     RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
1491     WriteData1;
1492
1493 when others =>
1494     RS <= '0'; RW <= '0';
1495
1496 end case;
1497
1498 -- =====
1499 when "1100" =>
1500
1501     case pr_state is
1502
1503         when FunctionSet1 =>
1504             RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
1505             FunctionSet2;
1506
1507         when FunctionSet2 =>
1508             RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
1509             FunctionSet3;
1510
1511         when FunctionSet3 =>
1512             RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
1513             FunctionSet4;
1514
1515         when FunctionSet4 =>
1516             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1517             FunctionSet5;
1518
1519         when FunctionSet5 =>
1520             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1521             FunctionSet6;
1522
1523         when FunctionSet6 =>

```

```

1508             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1509             FunctionSet7;
1510
1511         when FunctionSet7 =>
1512             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1513             FunctionSet8;
1514
1515         when FunctionSet8 =>
1516             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1517             FunctionSet9;
1518
1519         when FunctionSet9 =>
1520             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1521             FunctionSet10;
1522
1523         when FunctionSet10 =>
1524             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1525             FunctionSet11;
1526
1527         when FunctionSet11 =>
1528             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1529             FunctionSet12;
1530
1531         when FunctionSet12 =>
1532             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1533             FunctionSet13;
1534
1535         when FunctionSet13 =>
1536             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1537             FunctionSet14;
1538
1539         when FunctionSet14 =>
1540             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1541             FunctionSet15;
1542
1543         when FunctionSet15 =>
1544             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1545             FunctionSet16;
1546
1547         when FunctionSet16 =>
1548             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1549             FunctionSet17;
1550
1551         when FunctionSet17 =>
1552             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1553             FunctionSet18;
1554
1555         when FunctionSet18 =>
1556             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1557             FunctionSet19;
1558
1559         when FunctionSet19 =>
1560             RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1561             ClearDisplay ;
1562
1563         when ClearDisplay =>
1564             RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
1565             DisplayControl;
1566
1567         when DisplayControl =>
1568             RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
1569             EntryMode;
1570
1571         when EntryMode =>
1572             RS<= '0'; RW<= '0'; DB <= "00000110"; nx_state <=
1573             WriteData1;
1574
1575         when WriteData1 =>
1576             RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
1577             SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
1578
1579         when SetAddress1 =>
1580             RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=

```

```

        WriteData2; --COMANDO PARA POSICIONAR O CURSOR NA LINHA
        0 COLUNA 3

1563
1564
1565      when WriteData2 =>
1566          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
1567          WriteData3; --'F'
1568
1569      when WriteData3 =>
1570          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
1571          WriteData4; --'r'
1572
1573      when WriteData4 =>
1574          RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
1575          WriteData5; --'e'
1576
1577      when WriteData5 =>
1578          RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
1579          WriteData6; --'q'
1580
1581      when WriteData6 =>
1582          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
1583          WriteData7; --';'
1584
1585      when WriteData7 =>
1586          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1587          WriteData8; --' '
1588
1589      when WriteData8 =>
1590          RS <= '1'; RW <= '0'; DB <= X"36"; nx_state <=
1591          WriteData9; --'6'
1592
1593      when WriteData9 =>
1594          RS <= '1'; RW <= '0'; DB <= X"30"; nx_state <=
1595          WriteData10; --'0'
1596
1597      when WriteData10 =>
1598          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1599          WriteData11; --' '
1600
1601      when WriteData11 =>
1602          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
1603          WriteData12; --'H'
1604
1605      when WriteData12 =>
1606          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
1607          ReturnHome; --'z'
1608
1609      when ReturnHome =>
1610          RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
1611          WriteData1;
1612
1613      when others =>
1614          RS <= '0'; RW <= '0';
1615
1616  end case;

1617
1618  =====
1619  when others =>
1620
1621      case pr_state is
1622
1623          when FunctionSet1 =>
1624              RS<= '0'; RW<= '0'; DB<= "00111000"; nx_state <=
1625              FunctionSet2;
1626
1627          when FunctionSet2 =>
1628              RS<= '0'; RW<= '0'; DB <= "00111000"; nx_state <=
1629              FunctionSet3;
1630
1631          when FunctionSet3 =>
1632              RS <= '0'; RW <='0'; DB <= "00111000"; nx_state <=
1633              FunctionSet4;

```

```

1619
1620      when FunctionSet4 =>
1621          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1622          FunctionSet5;
1623
1624      when FunctionSet5 =>
1625          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1626          FunctionSet6;
1627
1628      when FunctionSet6 =>
1629          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1630          FunctionSet7;
1631
1632      when FunctionSet7 =>
1633          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1634          FunctionSet8;
1635
1636      when FunctionSet8 =>
1637          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1638          FunctionSet9;
1639
1640      when FunctionSet9 =>
1641          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1642          FunctionSet10;
1643
1644      when FunctionSet10 =>
1645          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1646          FunctionSet11;
1647
1648      when FunctionSet11 =>
1649          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1650          FunctionSet12;
1651
1652      when FunctionSet12 =>
1653          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1654          FunctionSet13;
1655
1656      when FunctionSet13 =>
1657          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1658          FunctionSet14;
1659
1660      when FunctionSet14 =>
1661          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1662          FunctionSet15;
1663
1664      when FunctionSet15 =>
1665          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1666          FunctionSet16;
1667
1668      when FunctionSet16 =>
1669          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1670          FunctionSet17;
1671
1672      when FunctionSet17 =>
1673          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
          FunctionSet18;
1674
1675      when FunctionSet18 =>
1676          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1677          FunctionSet19;
1678
1679      when FunctionSet19 =>
1680          RS <= '0'; RW <= '0'; DB <= "00111000"; nx_state <=
1681          ClearDisplay ;
1682
1683      when ClearDisplay =>
1684          RS <= '0'; RW <= '0'; DB <= "00000001"; nx_state <=
1685          DisplayControl;
1686
1687      when DisplayControl =>
1688          RS <= '0'; RW <= '0'; DB <= "00001100"; nx_state <=
1689          EntryMode;

```

```

1674
1675      when EntryMode =>
1676          RS<= '0'; RW<= '0'; DB <= "00000110"; nx_state <=
1677              WriteData1;
1678
1679      when WriteData1 =>
1680          RS <= '1'; RW <= '0'; DB <= "00100000"; nx_state <=
1681              SetAddress1; --'ESCREVE UM ESPAÇO EM BRANCO
1682
1683      when SetAddress1 =>
1684          RS <= '0'; RW <= '0'; DB <= "10000010"; nx_state <=
1685              WriteData2; --'COMANDO PARA POSICIONAR O CURSOR NA LINHA
1686                  0 COLUNA 3
1687
1688      when WriteData2 =>
1689          RS <= '1'; RW <= '0'; DB <= X"46"; nx_state <=
1690              WriteData3; --'F'
1691
1692      when WriteData3 =>
1693          RS <= '1'; RW <= '0'; DB <= X"72"; nx_state <=
1694              WriteData4; --'r'
1695
1696      when WriteData4 =>
1697          RS <= '1'; RW <= '0'; DB <= X"65"; nx_state <=
1698              WriteData5; --'e'
1699
1700      when WriteData5 =>
1701          RS <= '1'; RW <= '0'; DB <= X"71"; nx_state <=
1702              WriteData6; --'q'
1703
1704      when WriteData6 =>
1705          RS <= '1'; RW <= '0'; DB <= X"3A"; nx_state <=
1706              WriteData7; --':'
1707
1708      when WriteData7 =>
1709          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1710              WriteData8; --' '
1711
1712      when WriteData8 =>
1713          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1714              WriteData9; --' '
1715
1716      when WriteData9 =>
1717          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1718              WriteData10; --' '
1719
1720      when WriteData10 =>
1721          RS <= '1'; RW <= '0'; DB <= X"20"; nx_state <=
1722              WriteData11; --' '
1723
1724      when WriteData11 =>
1725          RS <= '1'; RW <= '0'; DB <= X"48"; nx_state <=
1726              WriteData12; --'H'
1727
1728      when WriteData12 =>
1729          RS <= '1'; RW <= '0'; DB <= X"7A"; nx_state <=
1730              ReturnHome; -- 'z'
1731
1732      when ReturnHome =>
1733          RS <= '0'; RW <= '0'; DB <= "10000000"; nx_state <=
1734              WriteData1;
1735
1736      when others =>
1737          RS <= '0'; RW <= '0';
1738
1739      end case;
1740
1741  end case;
1742
1743  end process;
1744
1745 end hardware;

```


Set Frequencia

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity set_freq is
port(
    signal clk_50_MHz      : in std_logic;
    signal increment        : in std_logic;
    signal decrement        : in std_logic;
    signal freq_setada      : out std_logic_vector(3 downto 0);
    signal clk_button_aux   : in std_logic
);
end set_freq;

architecture logica of set_freq is
    signal inc_button_ans  : std_logic;
    signal dec_button_ans  : std_logic;
    signal barramento       : std_logic_vector(15 downto 0) := "0000000000000000";
begin
    component button
    port(
        signal in_button      : in std_logic;
        signal clk_button     : in std_logic;
        signal out_pin_button : out std_logic
    );
    end component;

begin
    inc_button : button port map( clk_button => clk_button_aux, in_button => increment, out_pin_button => inc_button_ans );
    dec_button : button port map( clk_button => clk_button_aux, in_button => decrement, out_pin_button => dec_button_ans );

process(clk_50_MHz)
    variable contador_base   : integer := 0;
    variable contador_lus    : integer := 0;
    variable freq_count_aux : std_logic_vector(15 downto 0) := "0000000000000000";
    variable delay           : integer := 0;
    variable inicializacao  : integer := 0;
begin
    if (rising_edge(clk_50_MHz)) then
        contador_base := contador_base + 1;
        if(contador_base = 50) then
            contador_lus := contador_lus + 1;
            contador_base := 0;
        end if;
        if (inicializacao = 0) then
            case freq_count_aux is
                when "0000000000000000" =>
                    if(contador_lus = 417000) then
                        freq_count_aux <= "0001000000000000";
                        contador_lus := 0;
                    end if;
                when "0001000000000000" =>
                    if(contador_lus = 417000) then
                        freq_count_aux <= "0010000000000000";
                        contador_lus := 0;
                    end if;
                when "0010000000000000" =>
                    if(contador_lus = 417000) then
                        freq_count_aux <= "0011000000000000";
                        contador_lus := 0;
                    end if;
            end case;
        end if;
    end if;
end process;
```

```

when "0011000000000000" =>
    if(contador_lus = 417000) then
        freq_count_aux <= "0100000000000000";
        contador_lus := 0;
    end if;

when "0100000000000000" =>
    if(contador_lus = 417000) then
        freq_count_aux <= "0101000000000000";
        contador_lus := 0;
    end if;

when "0101000000000000" =>
    if(contador_lus = 417000) then
        freq_count_aux <= "0110000000000000";
        contador_lus := 0;
    end if;

when "0110000000000000" =>
    if(contador_lus = 417000) then
        freq_count_aux <= "0111000000000000";
        contador_lus := 0;
    end if;

when "0111000000000000" =>
    if(contador_lus = 417000) then
        freq_count_aux <= "1000000000000000";
        contador_lus := 0;
    end if;

when "1000000000000000" =>
    if(contador_lus = 417000) then
        freq_count_aux <= "1001000000000000";
        contador_lus := 0;
    end if;

when "1001000000000000" =>
    if(contador_lus = 417000) then
        freq_count_aux <= "1010000000000000";
        contador_lus := 0;
    end if;

when "1010000000000000" =>
    if(contador_lus = 417000) then
        freq_count_aux <= "1011000000000000";
        contador_lus := 0;
    end if;

when "1011000000000000" =>
    if(contador_lus = 417000) then
        freq_count_aux <= "1100000000000000";
        contador_lus := 0;
    end if;

when "1100000000000000" =>
    if(contador_lus = 417000) then
        inicializacao := 1;
        contador_lus := 0;
    end if;

when others =>
    inicializacao := 1;
    contador_lus := 0;

end case;

else

    if(contador_lus = 100) then
        if (inc_button_ans = '0') then
            if(freq_count_aux < "1100000000000000") then

```

```

        end case;

    else

        if(contador_lus = 100) then

            if (inc_button_ans = '0') then
                if(freq_count_aux < "1100000000000000") then
                    freq_count_aux := freq_count_aux + '1';
                end if;
            end if;

            if (dec_button_ans = '0') then
                if(freq_count_aux > "0000000000000000") then
                    freq_count_aux := freq_count_aux - '1';
                end if;
            end if;

            contador_lus := 0;
        end if;

    end if;

    barramento <= freq_count_aux;
end process;

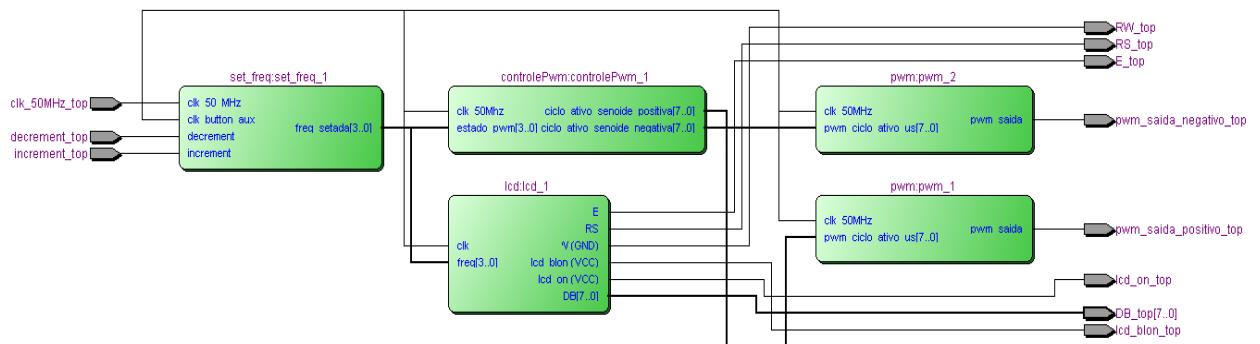
freq_setada(3) <= barramento(15);
freq_setada(2) <= barramento(14);
freq_setada(1) <= barramento(13);
freq_setada(0) <= barramento(12);

end logica;

```

Anexo C

RTL Viewer



Considerações finais

Nosso código criado funcionou com perfeição, no entanto o circuito não funcionou completamente como desejávamos. O problema foi no dimensionamento de um dos capacitores, que não tinha a tensão máxima adequada para o funcionamento do circuito e acabou por queimar.

Referências

- <http://www.ni.com/white-paper/6984/pt/>
- <http://pdf1.alldatasheet.com/datasheet-pdf/view/14624/PANJIT/1N4007.html>
- <https://leifrautn.blogspot.com.br/2014/10/inversor-monofasico-dcac-de-pwm-senoidal.html>