## PROYECTO CALL CENTER

## DESCRIPCIÓN DE BLOQUES

Nombre: Carlos Neira

Nombre: Martín Herrera

Fecha: 2/08/2018

Docente: Ing. Leonardo Muñoz

```
1
      LIBRARY IEEE;
      USE
            IEEE.STD_LOGIC_1164 all;
 3
           IEEE.STD_LOGIC_UNSIGNED all;
 4
      entity MSS_Call_Center is
    Port (clock, resetn, start, boton, BORRAR, Fu, HISTORIAL_LLAMADAS, cont6, Re_oc, Re_des,
 6
      igual_oc,dir_hist_igual,igual_hist1,
                  REALIZAR_LLAMADA , FINALIZAR_LLAMADA , icuenta0 , icuenta1 , icuenta2 , icuenta3 , icuenta4 ,
 7
      icuenta5,u1,u2,u3,u4,u5,u6,u7,u8,
Fu_ad1,Fu_ad2,Fu_ad3,Fu_ad4,Fu_ad5,Fu_ad6,Fu_ad7,Fu_ad8,iusuario1,iusuario2,
 8
      iusuario3,iusuario4,iusuario5,iusuario6,
                  iusuario7,iusuario8,iusuario_ant1,iusuario_ant2,iusuario_ant3,iusuario_ant4,
 9
      iusuario_ant5,iusuario_ant6,iusuario_ant7,
10
                  iusuario_ant8: IN std_logic;
11
                  \verb|incrementa|, \verb|reset_reg|, \verb|cuenta_a_0|, \verb|incremento_u1|, \verb|incremento_u2|, \verb|incremento_u3|, \\
12
      incremento_u4,incremento_u5,incremento_u6,
      incremento_u7,incremento_u8,we_u1,we_u2,we_u3,we_u4,we_u5,we_u6,we_u7,we_u8,
reset_oc,OCUPADO,reset_hist,disp_select,rhist_u1,
13
      14
16
      selec_u1,selec_u2,selec_u3,selec_u4,selec_u5,selec_u6,selec_u7,selec_u8: OUT
std_logic_vector (1 downto 0);
17
18
19
20
21
22
23
24
                  enable_reg: OUT std_logic_vector (5 downto 0));
      end MSS_Call_Center;
      ARCHITECTURE sol OF MSS_Call_Center IS
25
      Type estado is (Ta,Tb,Tc,Td,Te,Tf,Tg,Th,Ti,Tj,Tk,Tl,Tm,Tr,Ts,Tt,Tu,Tv,Tw,Tx,Ty,T_oc,
T_oc_apagado,T_hist,T_hist1,T_hi,Tl);--falta revisar qué estados cambié
26
27
28
29
      Signal y: estado;
      Begin
          Process (clock, resetn)
          Begin
              if resetn='0' then y<=Ta;
elsif (clock'event and clock='1') then</pre>
30
31
32
33
34
35
36
37
38
40
41
42
43
44
45
                  case y is
                     when Ta => if start='1' then y<=Tb; else y<=Ta; end if; when Tb => if start='1' then y<=Tb; else y<=Tc; end if; when Tc => if start='1' then y<=Td;
                                    elsif boton='1' then y<=Tf;
                                    elsif boton= 1 then y<=11,
elsif (BORRAR='1') then y<=Tr;
elsif Fu='1' then y<=Tk;
elsif HISTORIAL_LLAMADAS='1' then y<=T_hist;
                                   else y<=Tc; end if;
if start='1' then y<=Td; else y<=Ta; end if;</pre>
                     when Td =>
                                   y<=Te;
                      when
                            Τf
                                    if boton='1' then y<=Te; else y<=Tg; end if;</pre>
                     When Te =>
                      when Tg => if cont6='0' then y<=Th; else y<=Ti; end if;
                                   y<=Tc;--Por el incremento, es una salida if Re_oc='1' then y<=T_oc; elsif Re_des='1' then y<=Tj;
                     when Th =>
46
47
                     when Ti =>
48
                                    elsif BORRAR='1' then y<=Tr;
49
                                    else y<=Ti; end if;</pre>
50
51
                      when T_oc => if (igual_oc='1') then y<=T_oc_apagado; else y<=T_oc; end if;
      --estado de transicion para poder apagar la señal de ocupado
52
53
                     when T_oc_apagado => y<=Tc;
when T_hist => if HISTORIAL_LLAMADAS='1' then y<=T_hist; else y<=T_hist1; end</pre>
       if:
54
                     when T_hist1 => if dir_hist_igual='1' then y<=Tc;--se sobre entiende que
      sólo un usuario está seleccionado antes de que presione el botón del historial
                     else y<=T_hi;end if;
when T_hi => if igual_hist1='1' then y<=T1;
55
56
57
58
59
60
61
62
63
64
65
66
                                    else y \le T_hi; end if;
                      when T1 \Rightarrow y <= T_hist1;
                      when Tj => if REALIZAR_LLAMADA = '1' then y<=Tj; else y<=Tl; end if;
                     when T1 => y<=Ts; --Porque hay una salida (que el led se encienda)
                     when Tk => if FINALIZAR_LLAMADA = '1' then y <= Tk; else y <= Tm; end if;
                     when Tm => y<=Tt; --Porque hay una salida (que el led se apague)
                      when Tr => if BORRAR='1' then y<=Tr; else y<=Ty; end if;
67
68
                      When Ts => y <= Tu;
69
                     when Tu => y<=Tw;
```

```
when Tw => y <= Tc;
 71
                                 when
                                          Tt \Rightarrow y \leq Tv;
  72
                                 when Tv \Rightarrow y <= Tx;
 73
                                 when Tx => y<=Tc;
 74
75
                                 when Ty => y<=Tc;
  76
                           end case;
 77
                      end if;
 78
                end Process;
 79
                Process(y)
 80
                Begin
 81
                      case y is
 82
                                 when Ta => incrementa<='0';reset_reg<='1';cuenta_a_0<='0';--Condicion</pre>
           83
          84
           incremento_u8 <= '0'; we_u1 <= '0'; --de pasar a otro estado todo</pre>
                                                     we_u2<='0';we_u3<='0';we_u4<='0';we_u5<='0';we_u6<='0';we_u7<='0';
 85
          86
 87
 88
           89
          90
          91
          92
 93
 94
           ihist_u1<='0';ihist_u2<='0';ihist_u3<='0';ihist_u4<='0';ihist_u5<=
'0';ihist_u6<='0';ihist_u7<='0';ihist_u8<='0';

when Tc => incrementa<='0';reset_reg<='1';cuenta_a_0<='0';--Condicion
 95
 96
           inicial, si en un estado cambia esa condicion
 97
                                                      incremento_u1 <= '0';incremento_u2 <= '0';incremento_u3 <= '0';</pre>
           98
           99
          100
           reset_oc<='1'; OCUPADO<='0'; reset_hist<='1'; disp_select<='0'; rhist_u1<='1'; rhist_u2<='1'; rhist_u3<='1'; rhist_u4<='1'; rhist_u5<= '1'; rhist_u6<='1'; rhist_u7<='1'; rhist_u8<='1'; ihist_u1<='0'; ihist_u2<='0'; ihist_u3<='0'; ihist_u4<='0'; ihist_u5<= '0'; ihist_u6<='0'; ihist_u7<='0'; ihist_u8<='0';
101
102
103
                                 104
105
106
                                                      elsif icuenta2='1' then enable_reg<="000100"
107
                                                                icuenta3='1'
                                                                                        then enable_reg<="001000"
108
                                                      elsif
                                                                                        then enable_reg<="010000";
                                                     elsif icuenta4='1' then enable_reg<="010000";
elsif icuenta5='1' then enable_reg<="100000";end if;
109
110
                                 when Te => enable_reg<="000000";</pre>
111
                                 when Tg =>
112
                                          Th => incrementa<='1';</pre>
                                 When
114
                                 when Ti =>
115
                                 when T_oc => reset_oc<='0'; OCUPADO<='1';</pre>
116
          when T_oc_apagado => OCUPADO<='0';
when T_hist => selec_u1<="10"; selec_u2<="10"; selec_u3<="10"; selec_u4<="10";
selec_u5<="10"; selec_u6<="10"; selec_u7<="10"; selec_u8<="10";
ihist_u1<='0'; ihist_u2<='0'; ihist_u3<='0'; ihist_u4<='0';
ihist_u5<='0'; ihist_u6<='0'; ihist_u1<='0'; ihist_u2<='0'; ihist_u3<='0'; ihist_u4<='0';
ihist_u5<='0'; ihist_u5
117
118
119
           rhist_u1<= '0';rhist_u3<='0';rhist_u3<='0';rhist_u1<='0';
rhist_u5<='0';rhist_u6<='0';rhist_u7<='0';rhist_u8<='0';--solamente en estos estados el contador para mostrar el historial when T hist1 => colora 1 """
120
           when T_hist1 => selec_u1<="10";selec_u2<="10";selec_u3<="10";selec_u4<="10";
selec_u5<="10";selec_u6<="10";selec_u7<="10";selec_u8<="10";--debe functionar por eso es</pre>
121
           '0', ya que con '1' el reset se activa

rhist_u1<='0';rhist_u2<='0';rhist_u3<='0';rhist_u1<='0';rhist_u5<=
'0';rhist_u6<='0';rhist_u7<='0';rhist_u8<='0';
ihist_u1<='0';ihist_u2<='0';ihist_u3<='0';ihist_u4<='0';ihist_u5<=
'0';ihist_u6<='0';ihist_u7<='0';ihist_u8<='0';disp_select<='1';
122
123
```

```
124
125
         126
127
128
          empieza en 0
129
                             when T1 => if u1='1' then ihist_u1<='1';--incremento para cada adress de la
          ram
130
                                                elsif u2='1' then ihist_u2<='1';
                                                elsif u3='1'
                                                                     then ihist_u3<='1
131
                                                         u4='<mark>1</mark>'
132
                                                                     then ihist_u4<='
                                                elsif
                                                elsif u5='1'
                                                                     then ihist_u5<='1'
133
                                                elsif u6='1' then ihist_u6<='1'
134
                                                elsif u7='1' then ihist_u7<='1
135
136
                                                elsif u8='1'
                                                                     then ihist_u8<='1';end if;disp_select<='1';</pre>
137
                             when Tj => clock_ff_d<='0';
    if (u1='1') then we_u1<='1'
        elsif (u2='1') then we_u2<=</pre>
138
139
                                                          (u2='1') then we_u2<='(u3='1') then we_u2<='
141
                                                elsif
                                                          (u4='1') then we_u4<='
142
                                                         (u5='1') then we_u5<='1'
(u6='1') then we_u6<='1'
(u7='1') then we_u7<='1'
143
                                                elsif
144
                                                elsif
          when Tk => if Fu_ad1='1' then incremento_u1<='1';--puedo omitir
"FINALIZAR_LLAMADA='1'" porque se sobre entiende</pre>
145
146
147
                                                porque se sobre entiende elsif Fu_ad2='1' then incremento_u2<='1';--Solo cuando finalizo
148
         la llamada debo avanzar al siguiente adress de la ram
elsif Fu_ad3='1' then incremento_u3<='1';--Podria yo cambiar y
149
         establecer 2 clocks, uno para encendido y otro para apagado
elsif Fu_ad4='1' then incremento_u4<='1';--con eso evitaría que
150
         salgan las 'x's . La solucion es cuando un led enciende

elsif Fu_ad5='1' then incremento_u5<='1';--escribir que los demas
151
          sean cero, pero con dos clocks evito que alguno que
                                                elsif Fu_ad6='1' then incremento_u6<='1';--esté encendido se apague elsif Fu_ad7='1' then incremento_u7<='1';
152
153
                                              elsif Fu_ad7='1' then incremento_u7<='1';
elsif Fu_ad8='1' then incremento_u8<='1';end if;clock_ff_d<='0';
if (u1='1') then led1<='1';--condicion para cada usuario encender
elsif (u2='1') then led2<='1';
elsif (u3='1') then led3<='1';
elsif (u4='1') then led4<='1';
elsif (u5='1') then led5<='1';
155
                             when T1 =>
156
157
                                               elsiT (u5='1') then led5<='1';
elsif (u6='1') then led6<='1';
elsif (u7='1') then led7<='1';
elsif (u8='1') then led8<='1'; end if; clock_ff_d<='0';
if (u1='1') then led1<='0'; --condicion para cada usuario apagar
elsif (u2='1') then led2<='0';
elsif (u4='1') then led3<='0';
elsif (u4='1') then led4<='0';
elsif (u5='1') then led4<='0';
158
161
163
                             when Tm =>
164
165
                                                          (u5='1')
                                                elsif
                                                                        then led5<='0'
167
                                               elsif (u6='1') then led6<='0';
elsif (u7='1') then led7<='0';
elsif (u8='1') then led8<='0'; end if;clock_ff_d<='0';
168
169
170
171
                             when Tr =>
                              when Ts => if (u1='1') then led1<='1';--Para que mantenga el valor anterior
172
         y el clock del flipflop funcione
173
                                                elsif
elsif
                                                          (u2='1') then led2<='
                                                          (u3='1')
                                                                        then led3<='1
174
                                                         (u4='1') then led4<='1
(u5='1') then led5<='1
(u6='1') then led6<='1
(u7='1') then led7<='1
175
                                                elsif
176
                                                elsif
177
                                                elsif
178
                                                elsif
                                               elsif (u8='1') then led8<='1'; end if;clock_ff_d<='1'; if (u1='1') then led1<='0';--Para que mantenga el valor anterior
179
180
                              when Tt =>
         y el clock del flipflop funcione
                                                         (u2='1') then led2<='0'
(u3='1') then led3<='0'
(u4='1') then led4<='0'
                                                elsif
                                                                        then led3<='0'
                                                elsif
elsif
182
                                                                                 led4<='0'
183
                                                          (u5='1')
                                                                        then led5<='0'
184
                                                elsif
         elsif (u5='1') then led5<='0';
elsif (u6='1') then led6<='0';
elsif (u7='1') then led7<='0';
elsif (u8='1') then led8<='0'; end if;clock_ff_d<='1';
elsif (u8='1') then led8<='0'; selec_u3<="01";selec_u4<="01";selec_u4<="01";selec_u4<="01";selec_u4<="01";
selec_u5<="01";selec_u6<="01";selec_u7<="01";selec_u8<="01";
when Tu => if (iusuario1='1') then led1<='1';--condicion para cada usuario
185
186
187
188
189
```

```
para encender el otro led
                                                                elsif (iusuario2='1') then led2<='1';
elsif (iusuario3='1') then led3<='1';</pre>
190
191
                                                                elsif (iusuario4='1') then led4<='1';
elsif (iusuario5='1') then led5<='1';
elsif (iusuario6='1') then led6<='1';
elsif (iusuario7='1') then led7<='1';
192
193
194
            elsit (lusuario7='1') then led7<='1';
    elsif (iusuario8='1') then led8<='1'; end if;clock_ff_d<='0';
    when Tv => clock_ff_d<='0';--condicion para cada usuario apagar el otro led
        selec_u1<="01";selec_u2<="01";selec_u3<="01";selec_u4<="01";
    selec_u5<="01";selec_u6<="01";selec_u7<="01";selec_u8<="01";
        if iusuario_ant1='1' then led1<='0';
        elsif iusuario_ant2='1' then led3<='0';
        elsif iusuario_ant3='1' then led3<='0';
        elsif iusuario_ant4-'1' then led3<='0';</pre>
195
196
197
198
199
200
201
                                                                elsif iusuario_ant3= 1 then led4<='0';
elsif iusuario_ant4='1' then led4<='0';
elsif iusuario_ant5='1' then led6<='0';
elsif iusuario_ant7='1' then led7<='0';
202
203
204
205
                                                                elsif iusuario_ant8='1' then led8<='0';end if;
206
                                        when Tw => if (iusuario1='1') then led1<='1';--Para que mantenga el valor
207
             del otro led
                                                                elsif (iusuario2='1') then led2<='1';
elsif (iusuario3='1') then led3<='1';
elsif (iusuario4='1') then led4<='1';
elsif (iusuario5='1') then led5<='1';</pre>
208
209
210
211
            elsif (iusuario7='1') then led6<='1';
elsif (iusuario7='1') then led7<='1';
elsif (iusuario8='1') then led8<='1'; end if;clock_ff_d<='1';
reset_reg<='0';cuenta_a_0<='1';
                                                                elsif (iusuario6='1') then led6<='1'
212
213
214
            215
216
217
218
219
220
                                                                elsif iusuario_ant5='1' then led5<='0'
221
                                       elsif iusuario_ant5= 1 then led5<= 0;
elsif iusuario_ant6='1' then led6<='0';
elsif iusuario_ant7='1' then led7<='0';
elsif iusuario_ant8='1' then led8<='0';end if;
when Ty => reset_reg<='0';cuenta_a_0<='1';
222
223
224
225
226
227
                          end case;
                   end Process;
             end sol;
228
```

```
library ieee;
use ieee.std_logic_1164.all;
 4
        package archivo_componentes is
 5
            Component CLOCK_DIV_10_MHz is
 6
              Port
 7
                   CLOCK_10MHz
                                            : TN
                                                      STD_LOGIC:
                  CLOCK_1MHZ
CLOCK_100KHZ
CLOCK_10KHZ
 8
                                           : OUT
                                                    STD_LOGIC;
 9
                                           : OUT
                                                    STD_LOGIC
10
                                           :OUT
                                                    STD_LOGIC:
11
                   CLOCK_1KHZ
                                           :OUT
                                                    STD_LOGIC;
12
                   CLOCK_100Hz
CLOCK_10Hz
                                                    STD_LOGIC;
                                           :OUT
13
                                           :OUT
                                                    STD_LOGIC
14
                   CLOCK_1Hz
                                                    STD_LOGIC);
                                           :OUT
15
            end Component;
16
17
            Component encoder_Dec_BCD is
18
                 Port ( Ent: in STD_LOGIC_VECTOR (9 downto 0);
19
                             Salida : OUT STD_LOGIC_VECTOR (3 downto 0));
20
21
             end Component;
22
            Component comparador_4bits is
23
24
25
26
27
28
29
31
33
34
35
36
37
38
39
                 port(A,B:_in std_logic_vector(3 downto 0);
                         igual: out std_logic);
            end Component;
            Component comparador_24bits is
  port(A,B: in std_logic_vector(23 downto 0);
  igual: out std_logic);
             end Component;
             component mux_3s
                 Port (num1: IN std_logic_vector (3 downto 0);
                           num2: IN std_logic_vector (3 downto 0);
num3: IN std_logic_vector (3 downto 0);
num4: IN std_logic_vector (3 downto 0);
num5: IN std_logic_vector (3 downto 0);
                           num6: IN std_logic_vector (3 downto 0);
                           selector: IN std_logic_vector (2 downto 0);
s: OUT std_logic_vector (3 downto 0));
40
41
42
43
            end component;
            Component mux_8s is
44
45
                 Port (num1: IN std_logic_vector (23 downto 0);
                           num2: IN std_logic_vector
                                                                    (23 downto 0);
46
47
48
49
50
51
52
53
54
55
56
57
58
60
61
                           num3: IN std_logic_vector
num4: IN std_logic_vector
num5: IN std_logic_vector
                                                                    (23 downto 0);
                                                                    (23 downto 0);
(23 downto 0);
                           num6: IN std_logic_vector
                                                                    (23 downto 0);
                           num7: IN std_logic_vector (23 downto 0);
num8: IN std_logic_vector (23 downto 0);
selector: IN std_logic_vector (7 downto 0);
                           s: OUT std_logic_vector (23 downto 0));
            end Component;
            Component mux_2selector_restador is
                 Port (num1: IN std_logic_vector (3 downto 0);
    num2: IN std_logic_vector (3 downto 0);
    num3: IN std_logic_vector (3 downto 0);
    selector: IN std_logic_vector (1 downto 0);
    s: OUT std_logic_vector (3 downto 0));
62
63
             end Component;
64
             Component mux_1selector is
                 Port (num1: IN std_logic_vector (3 downto 0);
    num2: IN std_logic_vector (3 downto 0);
    selector: IN std_logic;
65
66
67
68
69
71
72
73
74
75
76
77
78
                           s: OUT std_logic_vector (3 downto 0));
            end Component;
            Component contador_1_4bits is
                         clk_cont,reset_cont : IN STD_LOGIC;
suma: OUT STD_LOGIC_VECTOR (3 DOWNTO 0)
                   );
            end Component;
            80
                             q: out std_logic);
```

```
82
         end Component;
 83
 84
         component RAM is
 85
           port (clock
                          : in
                               std_logic;
                                std_logic;
std_logic_vector (3 downto 0);
std_logic_vector (23 downto 0)
 86
                            in
                 we
 87
                 address
                           in
                          : in
 88
                                                  (23 downto 0);
                 datain
 89
                 dataout : out std_logic_vector (23 downto 0));
 90
         end component;
 91
 92
93
94
95
         Component registro_sostenimiento is
            PORT(clock,reset,enable: IN STD_LOGIC;
                Ent : IN STD_LOGIC_VECTOR (3 downto 0);
Q : OUT STD_LOGIC_VECTOR (3 downto 0));
 96
         end Component;
 97
 98
         Component MSS_Call_Center is
 99
         Port (clock, resetn, start, boton, BORRAR, Fu, HISTORIAL_LLAMADAS, cont6, Re_oc, Re_des,
      igual_oc,dir_hist_igual,igual_hist1,
               REALIZAR_LLAMADA, FINALIZAR_LLAMADA, icuenta0, icuenta1, icuenta2, icuenta3, icuenta4,
100
      icuenta5,u1,u2,u3,u4,u5,u6,u7,u8,
Fu_ad1,Fu_ad2,Fu_ad3,Fu_ad4,Fu_ad5,Fu_ad6,Fu_ad7,Fu_ad8,iusuario1,iusuario2,
101
      102
103
               iusuario_ant8: IN std_logic;
104
               incrementa, reset_reg, cuenta_a_0, incremento_u1, incremento_u2, incremento_u3,
105
      incremento_u4, incremento_u5, incremento_u6
      incremento_u7,incremento_u8,we_u1,we_u2,we_u3,we_u4,we_u5,we_u6,we_u7,we_u8,
reset_oc,OCUPADO,reset_hist,disp_select,rhist_u1,
106
      107
108
109
               selec_u1,selec_u2,selec_u3,selec_u4,selec_u5,selec_u6,selec_u7,selec_u8: OUT
110
      std_logic_vector (1 downto 0);
111
               enable_reg: OUT std_logic_vector (5 downto 0));
112
113
114
         end Component;
115
116
117
      end archivo_componentes;
```

```
68
            --Numero del usuario 5
           signal num_u5: std_logic_vector(23 downto 0):="01100111001010010110000011";
 69
 70
            --Numero del usuario 6
           Signal num_u6: std_logic_vector(23 downto 0):="0110011100100011010110";
--Numero del usuario 7
 71
 72
73
           Signal num_u7: std_logic_vector(23 downto 0):="011001110001010000111000";
 74
            --Numero del usuario 8
           Signal num_u8: std_logic_vector(23 downto 0):="011001110100010001010011";
 75
           Begin
 77
                 uf \leftarrow ((u1) \text{ and } not(u2) \text{ and } not(u3) \text{ and } not(u4) \text{ and } not(u5) \text{ and } not(u6) \text{ and } not(u7)
           and not(u8)) or
 78
                               (not (u1) and (u2) and not(u3) and not(u4) and not(u5) and not(u6) and not(u7)
           and not(u8)) or
 79
                               (not (u1) and not(u2) and (u3) and not(u4) and not(u5) and not(u6) and not(u7)
           and not(u8)) or
 80
                                (not (u1) and not(u2) and not(u3) and (u4) and not(u5) and not(u6) and not(u7)
           and not(u8)) or
                                (not (u1) and not(u2) and not(u3) and not(u4) and (u5) and not(u6) and not(u7)
 81
           and not(u8) or
 82
                               (not (u1) and not(u2) and not(u3) and not(u4) and not(u5) and (u6) and not(u7)
           and not(u8)) or
 83
                               (not (u1) and not(u2) and not(u3) and not(u4) and not(u5) and not(u6) and (u7)
           and not(u8)) or
 84
                                (not (u1) and not(u2) and not(u3) and not(u4) and not(u5) and not(u6) and not(
           u7) and (u8)) )
                 usuario<=u8&u7&u6&u5&u4&u3&u2&u1;--usuario seleccionado
 86
                 numero_actual <= sal_reg_1&sal_reg_2&sal_reg_3&sal_reg_4&sal_reg_5&sal_reg_6;--numero
           actual
                 num1hist<=numero_ram(23)&numero_ram(22)&numero_ram(21)&numero_ram(20);--numero 1 del
 87
           historial que se obtiene de la RAM
                 \label{lem:num2hist} num2hist <= numero\_ram (19) & numero\_ram (18) & \underline{numero\_ram (17)} & numero\_ram (16); --el numero de label | numero\_ram (18) & \underline{numero\_ram (17)} & \underline{numero\_ram (16)}; --el numero | numero\_ram (18) & \underline{numero\_ram (17)} & \underline{numero\_ram (18)} & \underline{numero\_ram (18)
 88
           la ram puede usarse para mostrar el historial
 89
                 num3hist<=numero_ram(15)&numero_ram(14)&numero_ram(13)&numero_ram(12);--o para
           conocer el numero anterior
                 num4hist<=numero_ram(11)&numero_ram(10)&numero_ram(9)&numero_ram(8);
num5hist<=numero_ram(7)&numero_ram(6)&numero_ram(5)&numero_ram(4);</pre>
 90
 91
                 num6hist<=numero_ram(3)&numero_ram(2)&numero_ram(1)&numero_ram(0);
 92
 93
                 boton <= (num_bcd(3) \text{ or } num_bcd(2) \text{ or } num_bcd(1) \text{ or } num_bcd(0)) \text{ and } uf;--además de
           presionar un botón un usuario está seleccionado
 94
                 alguna_llamada <= (LLAMADA_EN_PROGRESO(0) or LLAMADA_EN_PROGRESO(1) or
           LLAMADA_EN_PROGRESO(2) or
 95
                                                    LLAMADA_EN_PROGRESO (3) or LLAMADA_EN_PROGRESO (4) or
           LLAMADA_EN_PROGRESO (5) or
                                                    LLAMADA_EN_PROGRESO (6) or LLAMADA_EN_PROGRESO (7));
 96
                 cont6 <= (not cuenta (3) and cuenta(2) and not cuenta(1) and cuenta(0));
Fu <= (uf and alguna_llamada and FINALIZAR_LLAMADA);--usuario seleccionado y alguna</pre>
 97
 98
           llamada en progreso, sólo ahí puedo finalizar
num_existe <= (iusuario1 or iusuario2 or iusuario3 or iusuario4 or iusuario5 or
 99
           iusuario6 or iusuario7 or iusuario8);--si el numero actual es igual a al menos un número
           en el registro
                 Re_oc <= (num_existe and alguna_llamada and REALIZAR_LLAMADA); -- Para mostrar el
100
           ocupado, si el numero escrito existe y ademas existe alguna llamáda y presiono realizar
           11amada
                 Re_des <= (num_existe and not (alguna_llamada) and REALIZAR_LLAMADA);--para realizar
101
           la llamada el numero escrito debe existir, no debe haber ninguna llamada y debo
           presionar realizar llamada
           dir_hist_igual <= (igual_h1 or igual_h2 or igual_h3 or igual_h4 or igual_h5 or igual_h6 or igual_h7 or igual_h8);--si la direccion de alguno de los historiales ya se igualo a la direccion de las ram en las que voy guardando
102
                 Fu_ad1 <= (uf and u1 and LLAMADA_EN_PROGRESO(0)); --solo finalizo la llamada si hay
103
           una en progreso y de paso el adress de la ram debe aumentar a la siguiente para guardar Fu_ad2 <= (uf and u2 and LLAMADA_EN_PROGRESO(1));
                                           and u2 and LLAMADA_EN_PROGRESO(1));
and u3 and LLAMADA_EN_PROGRESO(2));
104
                                     (uf
105
                 Fu_ad3 <=
106
                                     (uf
                                            and u4 and LLAMADA_EN_PROGRESO (3));
                 Fu_ad4 <=
                                            and u5 and LLAMADA_EN_PROGRESO (4));
and u6 and LLAMADA_EN_PROGRESO (5));
                                     (uf
107
                 Fu_ad5 <=
108
                                     (uf
                 Fu_ad6 <=
                 Fu_ad7 <=
                                     (uf
                                            and u7 and LLAMADA_EN_PROGRESO (6));
109
110
                 Fu_ad8 \ll (uf
                                            and u8 and LLAMADA_EN_PROGRESO (7));
111
                 DISPLAY_1n <= sal_mux_1; -- tomar en cuenta para la particion funcional
                 DISPLAY_2n <= sal_mux_2;
DISPLAY_3n <= sal_mux_3;
112
113
114
                 DISPLAY_4n <= sal_mux_4;
115
                 DISPLAY_5n <= sall_mux_
116
                 DISPLAY_6n <= sal_mux_6
117
                 GO: encoder_Dec_BCD port map (TECLADO, num_bcd);
118
                 G1_1: contador_1_4bits port map (incrementa, cuenta_a_0, cuenta);--contador para
           cuando presiono una tecla
119
                  --Contador para el adress del usuario n
                 G1_u1: contador_1_4bits port map (incremento_u1, reset_u1, direc_u1);
120
                 G1_u2: contador_1_4bits port map (incremento_u2, reset_u2, direc_u2);
G1_u3: contador_1_4bits port map (incremento_u3, reset_u3, direc_u3);
122
```

Date: August 02, 2018

```
G1_u4: contador_1_4bits
G1_u5: contador_1_4bits
123
                                         port map (incremento_u4, reset_u4, direc_u4);
124
           G1_u5:
                                         port map
                                                    (incremento_u5, reset_u5, direc_u5)
                                                    (incremento_u6, reset_u6, direc_u6);
125
           G1_u6: contador_1_4bits
                                         port map
           G1_u7: contador_1_4bits
                                                    (incremento_u7, reset_u7, direc_u7);
126
                                         port map
           G1_u8: contador_1_4bits port map (incremento_u8, --Contador para el historial adress del usuario n
127
                                         port map (incremento_u8, reset_u8, direc_u8);
128
                                                           (ihist_u1,rhist_u1,dhist_1);
129
           G1_hist_u1: contador_1_4bits
                                               port map
           G1_hist_u2: contador_1_4bits
G1_hist_u3: contador_1_4bits
G1_hist_u4: contador_1_4bits
130
                                               port map
                                                           (ihist_u2,rhist_u2,dhist_2)
                                                           (ihist_u3,rhist_u3,dhist_
131
                                               port map
132
                                                           (ihist_u4,rhist_u4,dhist_4);
                                               port map
           G1_hist_u5: contador_1_4bits
133
                                               port map
                                                           (ihist_u5,rhist_u5,dhist_5)
134
           G1_hist_u6: contador_1_4bits
G1_hist_u7: contador_1_4bits
                                               port map
                                                           (ihist_u6,rhist_u6,dhist_6);
                                                           (ihist_u7,rhist_u7,dhist_7);
(ihist_u8,rhist_u8,dhist_8);
135
                                               port map
           G1_hist_u8: contador_1_4bits
                                               port map
136
137
           G1_ocupado: contador_1_4bits
                                               port map
                                                           (CLOCK_1Hz,reset_oc,cuenta_oc);--cuenta hasta 2
       segundos
138
           G1_historial: contador_1_4bits port map (CLOCK_1Hz, reset_hist, cuenta_hist); -- cuenta 4
       segundos
       G2_resta_u1: mux_2selector_restador port map (direc_u1,direc_u1,dhist_1,selec_u1,adress_u1);--la RAM puede trabajar de 3 maneras G2_resta_u2: mux_2selector_restador port map (direc_u2,direc_u2,dhist_2,selec_u2,
139
140
       adress_u2);--quardando el numero y aumentando el adress
141
           adress_u3);--verificando la direccion anterior (ultimo numero guardado para saber a que
       numero llamo determinado usuario)
       G2_resta_u4: mux_2selector_restador port map (direc_u4,direc_u4,dhist_4,selec_u4, adress_u4);--y un conteo para mostral el historial en el display
142
143
           G2_resta_u5: mux_2selector_restador port map (direc_u5,direc_u5,dhist_5,selec_u5,
       adress_u5);
144
           G2_resta_u6: mux_2selector_restador port map (direc_u6,direc_u6,dhist_6,selec_u6,
       adress_u6);
145
           G2_resta_u7: mux_2selector_restador    port map (direc_u7,direc_u7,dhist_7,selec_u7,
       adress_u7);
146
           adress_u8);
147
           G2_h1: mux_8s port map (historial_u1, historial_u2, historial_u3, historial_u4,
       historial_u5, historial_u6, historial_u7, historial_u8, usuario, numero_ram); --depende del
       usuario que quiera ver el historial, entonces solo ahi se va a verificar el ultimo
       numero guardado en la ram de dicho usuario
148
           G2_sal_d1: mux_1selector port map (sal_reg_1,num1hist,disp_select,sal_mux_1);
        --seleccionar en el display:
           G2_sal_d2: mux_1selector port map (sal_reg_2,num2hist,disp_select,sal_mux_2);--el
149
       numero escrito actual o
150
           G2_sal_d3: mux_1selector port map (sal_reg_3,num3hist,disp_select,sal_mux_3);
         -mostrar el historial
151
           G2_sal_d4: mux_1selector
                                          port map
                                                      (sal_reg_4,num4hist,disp_select,sal_mux_4);
           G2_sal_d5: mux_1selector
G2_sal_d6: mux_1selector
                                                      (sal_reg_5,num5hist,disp_select,sal_mux_5);
(sal_reg_6,num6hist,disp_select,sal_mux_6);
152
                                          port map
153
                                          port map
154
           G3_h1: comparador_4bits
                                                    (direc_u1,dhist_1,igual_h1);
                                         port map
                   comparador_4bits
comparador_4bits
comparador_4bits
           G3_h2:
                                                    (direc_u2,dhist_2,igual_h2);
155
                                         port map
156
                                                    (direc_u3,dhist_3,igual_h3);
(direc_u4,dhist_4,igual_h4);
           G3_h3:
                                         port map
157
           G3 h4:
                                         port map
158
           G3_h5:
                   comparador_4bits
                                         port map
                                                    (direc_u5,dhist_5,igual_h5);
                   comparador_4bits
comparador_4bits
           G3_h6:
G3_h7:
                                                    (direc_u6,dhist_6,igual_h6);
159
                                         port map
                                         port map (direc_u7,dhist_7,igual_h7);
port map (direc_u8,dhist_8,igual_h8);
160
                   comparador_4bits
           G3_h8:
161
162
           G3_nu1: comparador_24bits port map (numero_actual,num_u1,iusuario1);--Si el numero
       actual le pertenece al usuario 1
G3_nu2: comparador_24bits por
163
                                           port map
                                                      (numero_actual,num_u2,iusuario2);--Si el numero
       actual le pertenece al usuario 2
           G3_nu3: comparador_24bits
G3_nu4: comparador_24bits
G3_nu5: comparador_24bits
164
                                           port map
                                                       (numero_actua],num_u3,iusuario3);
                                                       (numero_actual,num_u4,iusuario4);
(numero_actual,num_u5,iusuario5);
165
                                           port map
166
                                           port map
                                                       (numero_actual, num_u6, iusuario6);
           G3_nu6: comparador_24bits
167
                                           port map
       G3_nu7: comparador_24bits por G3_nu8: comparador_24bits por actual le pertenece al usuario 8
                                                       (numero_actual,num_u7,iusuario7);
(numero_actual,num_u8,iusuario8);--Si el numero
168
                                           port map
169
                                           port map
170
           G3_hu1: comparador_24bits
                                           port map
                                                      (numero_ram,num_u1,iusuario_ant1);--Si el numero
       anterior de la ram (para finalizar la llamada) le pertenece al usuario 1
171
                                                 map (numero_ram,num_u2,iusuario_ant2);--Si el numero
la llamada) le pertenece al usuario 2
           G3_hu2: comparador_24bits
                                           port map
       anterior de la ram (para finalizar
           G3_hu3: comparador_24bits
172
                                                       (numero_ram,num_u3,iusuario_ant3)
                                           port map
           G3_hu4: comparador_24bits
G3_hu5: comparador_24bits
G3_hu6: comparador_24bits
                                                       (numero_ram,num_u4,iusuario_ant4)
173
                                           port map
174
                                           port map
                                                       (numero_ram,num_u5,iusuario_ant5)
175
                                           port map
                                                       (numero_ram,num_u6,jusuarjo_ant6)
           G3_hu7: comparador_24bits
176
                                           port map
                                                       (numero_ram,num_u7,iusuario_ant7)
       G3_hu8: comparador_24bits port map (numero_ram,num_u8,iusuario_ant8);--Si el numero anterior de la ram (para finalizar la llamada) le pertenece al usuario 8
177
           G3_oc: comparador_4bits port map (dos,cuenta_oc,igual_oc);--agregado G3_cuentn1: comparador_4bits port map (cuenta,cero,icuenta0);--si la cuenta está en
178
179
       cero
```

```
Project: Call_Center
          G3_cuentn2: comparador_4bits G3_cuentn3: comparador_4bits
                                             port map (cuenta,uno,icuenta1); --si la cuenta está en uno
port map (cuenta,dos,icuenta2); --si la cuenta está en dos
180
181
          G3_cuentn4: comparador_4bits port map (cuenta,tres,icuenta3);--si la cuenta está en
182
       tres
183
          G3_cuentn5: comparador_4bits port map (cuenta,cuatro,icuenta4);--si la cuenta está en
       cuatro
184
          G3_cuentn6: comparador_4bits port map (cuenta,cinco,icuenta5);--si la cuenta está en
       cinco
185
          G3_hist: comparador_4bits port map (cuatro,cuenta_hist,iqual_hist1);--si ha llegado a
       los 4 segundos
186
          G4: flip_flop_D port map (clock_ff_d,reset_ff_d,led1,LLAMADA_EN_PROGRESO(0));
187
          G5: flip_flop_D
                             port map (clock_ff_d, reset_ff_d, led2, LLAMADA_EN_PROGRESO (1));
188
          G6: flip_flop_D
                              port map
                                        (clock_ff_d, reset_ff_d, led3, LLAMADA_EN_PROGRESO (2));
                                        (clock_ff_d,reset_ff_d,led4,LLAMADA_EN_PROGRESO(3));
          G7: flip_flop_D
189
                             port map
          G8: flip_flop_D port map (clock_ff_d,reset_ff_d,led5,LLAMADA_EN_PROGRESO (4));
G9: flip_flop_D port map (clock_ff_d,reset_ff_d,led6,LLAMADA_EN_PROGRESO (5));
G10: flip_flop_D port map (clock_ff_d,reset_ff_d,led7,LLAMADA_EN_PROGRESO (6));
G11: flip_flop_D port map (clock_ff_d,reset_ff_d,led8,LLAMADA_EN_PROGRESO (7));
G12: flip_flop_D port map (clock_ff_d,reset_ff_d,led8,LLAMADA_EN_PROGRESO (7));
190
191
192
193
          G12_1: registro_sostenimiento port map (clock,reset_reg,enable_reg(0),num_bcd.
194
       sal_reg_1);
195
          G12_2: registro_sostenimiento port map (clock, reset_reg, enable_reg(1), num_bcd,
       sal_reg_2);
G12_3: registro_sostenimiento port map (clock,reset_reg,enable_reg(2),num_bcd,
196
       sal_reg_3);
G12_4: registro_sostenimiento port map (clock, reset_reg, enable_reg(3), num_bcd,
197
       sal_reg_4);
198
                  registro_sostenimiento port map (clock,reset_reg,enable_reg(4),num_bcd,
          G12_5:
       sal_reg_5);
199
          G12_6: registro_sostenimiento port map (clock, reset_reg, enable_reg(5), num_bcd,
       sal_reg_6);
200
          G14_u1: RAM port map (clock,we_u1,adress_u1,numero_actual,historial_u1);--usuario 1
       historial
201
          G14_u2: RAM port map (clock,we_u2,adress_u2,numero_actual,historial_u2);--usuario 2
       historial
202
          G14_u3: RAM port map (clock, we_u3, adress_u3, numero_actual, historial_u3); --usuario 3
       historial
203
          G14_u4: RAM port map (clock, we_u4, adress_u4, numero_actual, historial_u4); --usuario 4
       historial
204
          G14_u5: RAM port map (clock, we_u5, adress_u5, numero_actual, historial_u5); --usuario 5
       historial
205
          G14_u6: RAM port map (clock, we_u6, adress_u6, numero_actual, historial_u6); --usuario 6
       historial
206
          G14_u7: RAM port map (clock, we_u7, adress_u7, numero_actual, historial_u7); --usuario 7
       historial
207
          G14_u8: RAM port map (clock, we_u8, adress_u8, numero_actual, historial_u8); --usuario 8
       historial
208
209
          MSS: MSS_Call_Center port map
210
                  clock,resetn,start,boton,BORRAR,Fu,HISTORIAL_LLAMADAS,cont6,Re_oc,Re_des,
       igual_oc,dir_hist_igual,igual_hist1,
REALIZAR_LLAMADA,FINALIZAR_LLAMADA,icuenta0,icuenta1,icuenta2,icuenta3,icuenta4,
icuenta5,u1,u2,u3,u4,u5,u6,u7,u8,
211
212
                  Fu_ad1,Fu_ad2,Fu_ad3,Fu_ad4,Fu_ad5,Fu_ad6,Fu_ad7,Fu_ad8,iusuario1,iusuario2,
       213
       iusuario_ant5,iusuario_ant6,iusuario_ant7
214
                  iusuario_ant8,incrementa,reset_reg,cuenta_a_0,incremento_u1,incremento_u2,
       incremento_u3, incremento_u4, incremento_u5, incremento_u6, incremento_u6, incremento_u7, incremento_u8, we_u1, we_u2, we_u3, we_u4, we_u5, we_u6, we_u7, we_u8, reset_oc,OCUPADO, reset_hist, disp_select, rhist_u1,
215
                  rhist_u2, rhist_u3, rhist_u4, rhist_u5, rhist_u6, rhist_u7, rhist_u8, ihist_u1, ihist_u2
216
       217
       selec_u3, selec_u4, selec_u5, selec_u6, selec_u7, selec_u8,
218
                  enable_reg);
```

219

end uno;

