

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  package archivo_componentes is
5  Component CLOCK_DIV_10_MHz is
6  Port
7  (  CLOCK_10MHz      :IN   STD_LOGIC;
8    CLOCK_1MHz       :OUT  STD_LOGIC;
9    CLOCK_100KHz     :OUT  STD_LOGIC;
10   CLOCK_10KHz      :OUT  STD_LOGIC;
11   CLOCK_1KHz       :OUT  STD_LOGIC;
12   CLOCK_100Hz      :OUT  STD_LOGIC;
13   CLOCK_10Hz       :OUT  STD_LOGIC;
14   CLOCK_1Hz        :OUT  STD_LOGIC);
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 end Component;
21
22 Component comparador_4bits is
23 port(A,B: in std_logic_vector(3 downto 0);
24      igual: out std_logic);
25 end Component;
26
27 Component comparador_24bits is
28 port(A,B: in std_logic_vector(23 downto 0);
29      igual: out std_logic);
30 end Component;
31
32 component mux_3s
33 Port (num1: IN std_logic_vector (3 downto 0);
34       num2: IN std_logic_vector (3 downto 0);
35       num3: IN std_logic_vector (3 downto 0);
36       num4: IN std_logic_vector (3 downto 0);
37       num5: IN std_logic_vector (3 downto 0);
38       num6: IN std_logic_vector (3 downto 0);
39       selector: IN std_logic_vector (2 downto 0);
40       s: OUT std_logic_vector (3 downto 0));
41 end component;
42
43 Component mux_8s is
44 Port (num1: IN std_logic_vector (23 downto 0);
45       num2: IN std_logic_vector (23 downto 0);
46       num3: IN std_logic_vector (23 downto 0);
47       num4: IN std_logic_vector (23 downto 0);
48       num5: IN std_logic_vector (23 downto 0);
49       num6: IN std_logic_vector (23 downto 0);
50       num7: IN std_logic_vector (23 downto 0);
51       num8: IN std_logic_vector (23 downto 0);
52       selector: IN std_logic_vector (7 downto 0);
53       s: OUT std_logic_vector (23 downto 0));
54 end Component;
55
56 Component mux_2selector_restador is
57 Port (num1: IN std_logic_vector (3 downto 0);
58       num2: IN std_logic_vector (3 downto 0);
59       num3: IN std_logic_vector (3 downto 0);
60       selector: IN std_logic_vector (1 downto 0);
61       s: OUT std_logic_vector (3 downto 0));
62 end Component;
63
64 Component mux_1selector is
65 Port (num1: IN std_logic_vector (3 downto 0);
66       num2: IN std_logic_vector (3 downto 0);
67       selector: IN std_logic;
68       s: OUT std_logic_vector (3 downto 0));
69 end Component;
70
71 Component contador_1_4bits is
72 PORT (
73     clk_cont,reset_cont : IN STD_LOGIC;
74     suma: OUT STD_LOGIC_VECTOR(3 DOWNTO 0)
75 );
76 end Component;
77
78 Component flip_flop_D is
79 port (clk, reset: in std_logic;
80       d: in std_logic;
81       q: out std_logic);

```

```

82     end Component;
83
84     component RAM is
85         port (clock      : in  std_logic;
86              we          : in  std_logic;
87              address     : in  std_logic_vector (3 downto 0);
88              datain      : in  std_logic_vector (23 downto 0);
89              dataout     : out std_logic_vector (23 downto 0));
90     end component;
91
92     Component registro_sostenimiento is
93         PORT(clock,reset,enable: IN STD_LOGIC;
94              Ent : IN STD_LOGIC_VECTOR (3 downto 0);
95              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,
100             igual_oc,dir_hist_igual,igual_hist1,
101             REALIZAR_LLAMADA,FINALIZAR_LLAMADA,icuenta0,icuenta1,icuenta2,icuenta3,icuenta4,
102             icuenta5,u1,u2,u3,u4,u5,u6,u7,u8,
103             Fu_ad1,Fu_ad2,Fu_ad3,Fu_ad4,Fu_ad5,Fu_ad6,Fu_ad7,Fu_ad8,iusuario1,iusuario2,
104             iusuario3,iusuario4,iusuario5,iusuario6,
105             iusuario7,iusuario8,iusuario_ant1,iusuario_ant2,iusuario_ant3,iusuario_ant4,
106             iusuario_ant5,iusuario_ant6,iusuario_ant7,
107             iusuario_ant8: IN std_logic;
108
109             incrementa,reset_reg,cuenta_a_0,incremento_u1,incremento_u2,incremento_u3,
110             incremento_u4,incremento_u5,incremento_u6,
111             incremento_u7,incremento_u8,we_u1,we_u2,we_u3,we_u4,we_u5,we_u6,we_u7,we_u8,
112             reset_oc,OCUPADO,reset_hist,disp_select,rhist_u1,
113             rhist_u2,rhist_u3,rhist_u4,rhist_u5,rhist_u6,rhist_u7,rhist_u8,ihist_u1,ihist_u2,
114             ,ihist_u3,ihist_u4,ihist_u5,ihist_u6,ihist_u7,
115             ihist_u8,led1,led2,led3,led4,led5,led6,led7,led8,clock_ff_d: OUT std_logic;
116
117             selec_u1,selec_u2,selec_u3,selec_u4,selec_u5,selec_u6,selec_u7,selec_u8: OUT
std_logic_vector (1 downto 0);
118
119             enable_reg: OUT std_logic_vector (5 downto 0));
120     end Component;
121
122 end archivo_componentes;

```