



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE COMPUTO



Práctica 4

“Piano con Mux y Decodificador”

Diseño de Sistemas Digitales

Rene Baltazar Jiménez Ruiz

Equipo 7

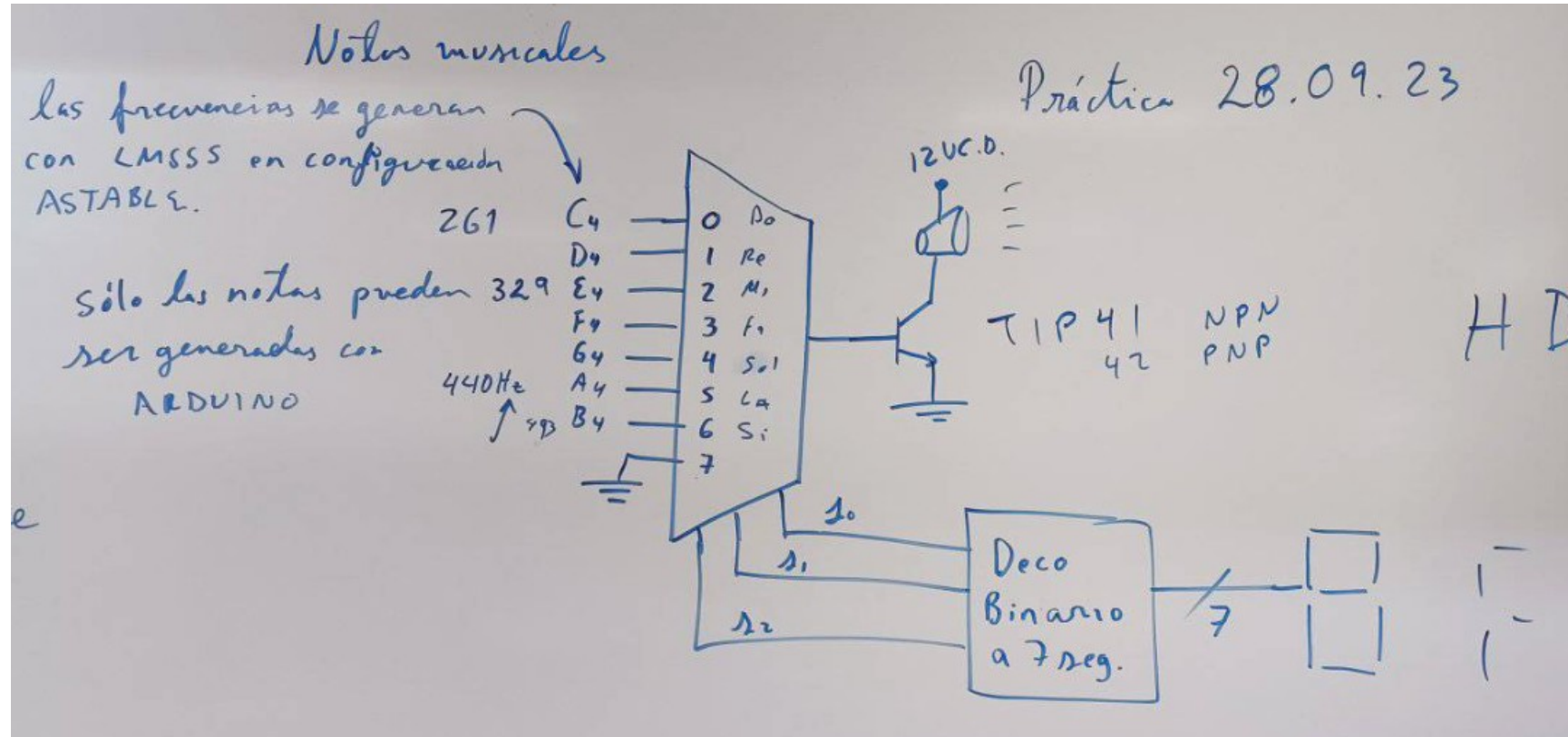
Hernández Hernández Roberto Issac

Navarrete Piña Arath Antonio

Yescas García Ulysses Aarón

4CV2

DIAGRAMA



CODIGO VHDL

```
1  LIBRARY IEEE;
2  USE IEEE.STD_LOGIC_1164.ALL;
3  USE IEEE.STD_LOGIC_UNSIGNED.ALL;
4
5  ENTITY NotasMus IS
6
7  PORT ( CLK,D:IN STD_LOGIC;
8        S: IN STD_LOGIC_VECTOR (2 DOWNTO 0);
9        SEG: INOUT STD_LOGIC_VECTOR (6 DOWNTO 0);
10       MOS: INOUT STD_LOGIC_VECTOR (3 DOWNTO 0);
11       SONIDO : INOUT STD_LOGIC
12       );
13  END NotasMus;
14
15  ARCHITECTURE GeneraSonido OF NotasMus IS
16  ----- FRECUENCIA LA -----
17  SIGNAL CONT : STD_LOGIC_VECTOR (16 DOWNTO 0);
18  CONSTANT CONT_MAX: STD_LOGIC_VECTOR := "11011101111100100"; --113636(DEC)
19  ----- FRECUENCIA DO -----
20  SIGNAL CONT1 : STD_LOGIC_VECTOR (17 DOWNTO 0);
21  CONSTANT CONT_MAX1: STD_LOGIC_VECTOR := "101110110001010010"; --191570(DEC)
22  ----- FRECUENCIA MI -----
23  SIGNAL CONT2 : STD_LOGIC_VECTOR (17 DOWNTO 0);
24  CONSTANT CONT_MAX2: STD_LOGIC_VECTOR := "100101000110100111"; --151975(DEC)
25  ----- FRECUENCIA SOL -----
26  SIGNAL CONT3 : STD_LOGIC_VECTOR (16 DOWNTO 0);
27  CONSTANT CONT_MAX3: STD_LOGIC_VECTOR := "11111001000111111"; --127551(DEC)
28  ----- FRECUENCIA RE -----
29  SIGNAL CONT4 : STD_LOGIC_VECTOR (17 DOWNTO 0);
30  CONSTANT CONT_MAX4: STD_LOGIC_VECTOR := "101001100100011001"; --170265(DEC)
31  ----- FRECUENCIA FA -----
32  SIGNAL CONT5 : STD_LOGIC_VECTOR (17 DOWNTO 0);
33  CONSTANT CONT_MAX5: STD_LOGIC_VECTOR := "100010111101001000"; --143176(DEC)
34  ----- FRECUENCIA SI -----
35  SIGNAL CONT6 : STD_LOGIC_VECTOR (16 DOWNTO 0);
36  CONSTANT CONT_MAX6: STD_LOGIC_VECTOR := "11000101101110111"; --101239(DEC)
37
```

Se establecen las librerías y la entidad (las entradas y salidas que se utilizaran para el Código).

En la primera parte de la arquitectura se establece la frecuencia de cada una de las notas musicales, generamos una señal en cada una y determinamos el tamaño de su vector al conseguir el valor de la frecuencia que debe utilizarse.

Sabemos que obtenemos dicho valor...

Frecuencia de la nota = Frecuencia del CLK de la FPGA ÷ frecuencia octava 4 de la nota musical.

$$50,000,000,000 \div 440,000 = 113,636$$

$$113,636_2 = 11011101111100100$$

Dicho valor nos ayudara para establecer el el valor máximo que llegara cada nota en el divisor de frecuencia.

```

38 BEGIN
39 ----- LA -----
40
41 PROCESS (CLK)
42 BEGIN
43 IF CLK'EVENT AND CLK = '1' THEN
44     CONT <= CONT + 1;
45     IF CONT = CONT_MAX THEN
46         CONT <= "000000000000000000";
47     END IF;
48 END IF;
49 END PROCESS;
50
51 ----- DO -----
52
53 PROCESS (CLK)
54 BEGIN
55 IF CLK'EVENT AND CLK = '1' THEN
56     CONT1 <= CONT1 + 1;
57     IF CONT1 = CONT_MAX1 THEN
58         CONT1 <= "000000000000000000";
59     END IF;
60 END IF;
61 END PROCESS;
62
63 ----- MI -----
64
65 PROCESS (CLK)
66 BEGIN
67 IF CLK'EVENT AND CLK = '1' THEN
68     CONT2 <= CONT2 + 1;
69     IF CONT2 = CONT_MAX2 THEN
70         CONT2 <= "000000000000000000";
71     END IF;
72 END IF;
73 END PROCESS;
74
75 ----- SOL -----
76
77 PROCESS (CLK)
78 BEGIN
79 IF CLK'EVENT AND CLK = '1' THEN
80     CONT3 <= CONT3 + 1;
81     IF CONT3 = CONT_MAX3 THEN
82         CONT3 <= "000000000000000000";
83     END IF;
84 END IF;
85 END PROCESS;
86
87 ----- RE -----
88
89 PROCESS (CLK)
90 BEGIN
91 IF CLK'EVENT AND CLK = '1' THEN
92     CONT4 <= CONT4 + 1;
93     IF CONT4 = CONT_MAX4 THEN
94         CONT4 <= "000000000000000000";
95     END IF;
96 END IF;
97 END PROCESS;
98
99 ----- FA -----
100
101 PROCESS (CLK)
102 BEGIN
103 IF CLK'EVENT AND CLK = '1' THEN
104     CONT5 <= CONT5 + 1;
105     IF CONT5 = CONT_MAX5 THEN
106         CONT5 <= "000000000000000000";
107     END IF;
108 END IF;
109 END PROCESS;
110
111 ----- SI -----
112
113 PROCESS (CLK)
114 BEGIN
115 IF CLK'EVENT AND CLK = '1' THEN
116     CONT6 <= CONT6 + 1;
117     IF CONT6 = CONT_MAX6 THEN
118         CONT6 <= "000000000000000000";
119     END IF;
120 END IF;
121 END PROCESS;
122

```

En esta parte se genera el divisor de frecuencia para cada uno de los “CLK” correspondiente a su nota musical en donde oscilaran constantemente cuando el CLK = ‘0’ (dada las características de la FPGA esta invertido por eso se muestra que CLK = 1) hasta llegar a su valor máximo, después de llegar a su valor máximo se reiniciara en “0” (dependiendo del tamaño del vector) y volverán a ejecutarse de la misma manera.


```

122
123 -----SONIDO EN LA FPGA -----
124 --Tenemos logica negativa por lo cual todos los botones y swiches estan en 1 y los al precionar o activar los se cambian a '0'---
125 PROCESS (SONIDO, SEG)
126 BEGIN
127     MOS<="1110";
128     IF S="000" THEN --DO
129         SONIDO <= CONT1(17);
130         SEG <="0110001";
131     ELSIF S = "001" THEN --RE
132         SONIDO <= CONT4(17);
133         SEG <="1000010";
134     ELSIF S = "010" THEN --MI
135         SONIDO <= CONT2(17);
136         SEG <="0110000";
137     ELSIF S = "011" THEN --FA
138         SONIDO <= CONT5(17);
139         SEG <="0111000";
140     ELSIF S = "100" THEN --SOL
141         SONIDO <= CONT3(16);
142         SEG <="0100000";
143     ELSIF S = "101" THEN --LA
144         SONIDO <= CONT(16);
145         SEG <="0001000";
146     ELSIF S = "110" THEN --SI
147         SONIDO <= CONT6(16);
148         SEG <="1100000";
149     ELSE
150         SONIDO <= '1';
151         SEG <="1111111";
152     END IF;
153 END PROCESS;
154
155
156 END GeneraSonido;

```

Finalmente se establece el selector “S” y dada la combinación del selector da como salida cada una de las frecuencias de las notas musicales y su letra correspondiente mostrado en el display.

En cada una de las opciones del selector se tiene también la cadena de bits del display de 7 segmentos de cada una de las notas para poder mostrar la letra correspondiente de cada nota