



INSTITUTO POLITECNICO
NACIONAL



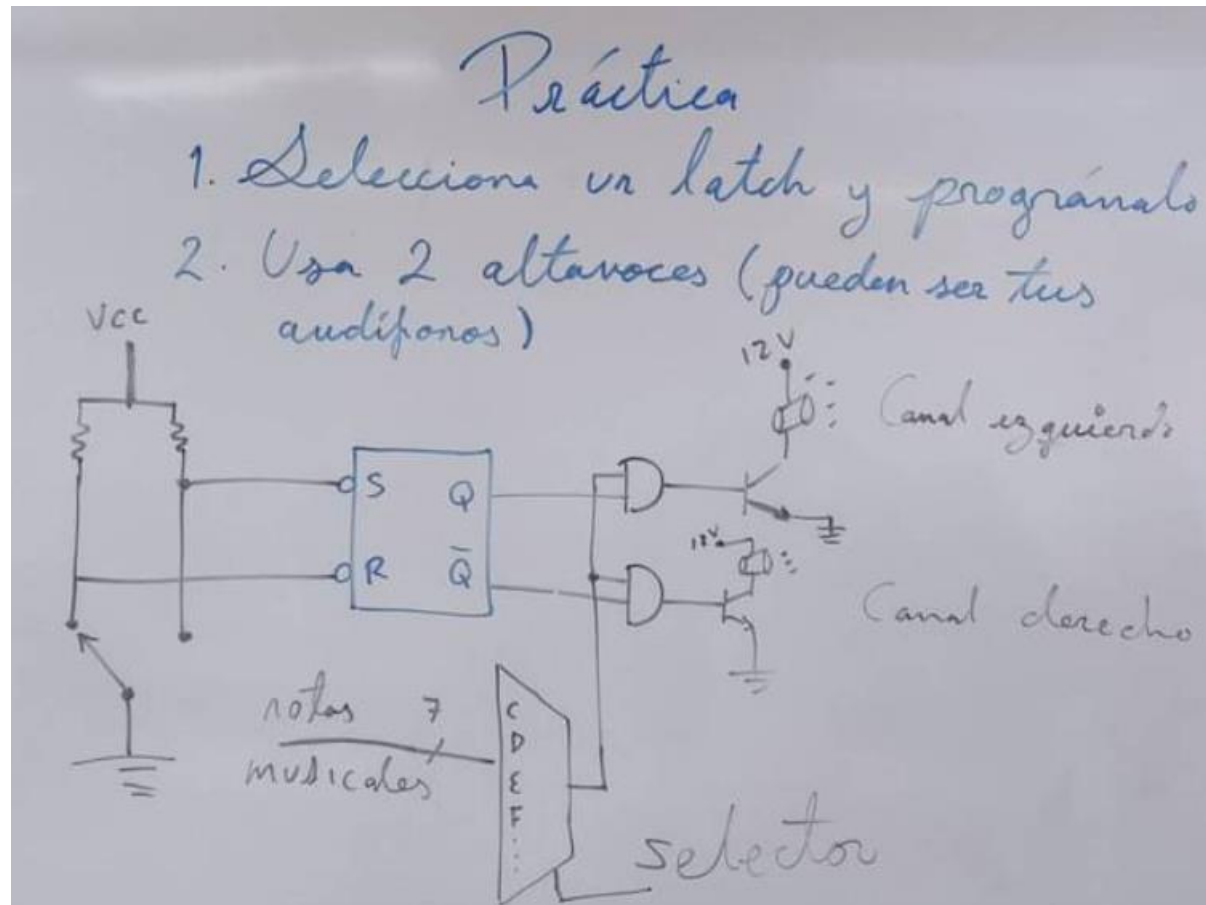
ESCUELA SUPERIOR DE COMPUTO

Práctica 5
“Piano con latch”

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 MusicaLatch IS
6
7  PORT ( CLK,D,R: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       Q,NQ: INOUT STD_LOGIC;
12       SONIDOI, SONIDOD : INOUT STD_LOGIC
13     );
14  END MusicaLatch;
15
16  ARCHITECTURE GeneraSonido OF MusicaLatch IS
17     ----- FRECUENCIA LA -----
18     SIGNAL CONT : STD_LOGIC_VECTOR (16 DOWNTO 0);
19     CONSTANT CONT_MAX: STD_LOGIC_VECTOR := "11011101111100100"; --113636(DEC)
20     ----- FRECUENCIA DO -----
21     SIGNAL CONT1 : STD_LOGIC_VECTOR (17 DOWNTO 0);
22     CONSTANT CONT_MAX1: STD_LOGIC_VECTOR := "101110110001010010"; --191570(DEC)
23     ----- FRECUENCIA MI -----
24     SIGNAL CONT2 : STD_LOGIC_VECTOR (17 DOWNTO 0);
25     CONSTANT CONT_MAX2: STD_LOGIC_VECTOR := "100101000110100111"; --151975(DEC)
26     ----- FRECUENCIA SOL -----
27     SIGNAL CONT3 : STD_LOGIC_VECTOR (16 DOWNTO 0);
28     CONSTANT CONT_MAX3: STD_LOGIC_VECTOR := "11111001000111111"; --127551(DEC)
29     ----- FRECUENCIA RE -----
30     SIGNAL CONT4 : STD_LOGIC_VECTOR (17 DOWNTO 0);
31     CONSTANT CONT_MAX4: STD_LOGIC_VECTOR := "101001100100011001"; --170265(DEC)
32     ----- FRECUENCIA FA -----
33     SIGNAL CONT5 : STD_LOGIC_VECTOR (17 DOWNTO 0);
34     CONSTANT CONT_MAX5: STD_LOGIC_VECTOR := "100010111101001000"; --143176(DEC)
35     ----- FRECUENCIA SI -----
36     SIGNAL CONT6 : STD_LOGIC_VECTOR (16 DOWNTO 0);
37     CONSTANT CONT_MAX6: STD_LOGIC_VECTOR := "11000101101110111"; --101239(DEC)
38

```

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.

```

39 BEGIN
40 ----- LA -----
41
42 PROCESS (CLK)
43 BEGIN
44 IF CLK'EVENT AND CLK = '1' THEN
45     CONT <= CONT + 1;
46     IF CONT = CONT_MAX THEN
47         CONT <= "000000000000000000";
48     END IF;
49 END IF;
50 END PROCESS;
51
52 ----- DO -----
53
54 PROCESS (CLK)
55 BEGIN
56 IF CLK'EVENT AND CLK = '1' THEN
57     CONT1 <= CONT1 + 1;
58     IF CONT1 = CONT_MAX1 THEN
59         CONT1 <= "000000000000000000";
60     END IF;
61 END IF;
62 END PROCESS;
63
64 ----- MI -----
65
66 PROCESS (CLK)
67 BEGIN
68 IF CLK'EVENT AND CLK = '1' THEN
69     CONT2 <= CONT2 + 1;
70     IF CONT2 = CONT_MAX2 THEN
71         CONT2 <= "000000000000000000";
72     END IF;
73 END IF;
74 END PROCESS;
75

```

```

76 ----- SOL -----
77
78 PROCESS (CLK)
79 BEGIN
80 IF CLK'EVENT AND CLK = '1' THEN
81     CONT3 <= CONT3 + 1;
82     IF CONT3 = CONT_MAX3 THEN
83         CONT3 <= "000000000000000000";
84     END IF;
85 END IF;
86 END PROCESS;
87
88 ----- RE -----
89
90 PROCESS (CLK)
91 BEGIN
92 IF CLK'EVENT AND CLK = '1' THEN
93     CONT4 <= CONT4 + 1;
94     IF CONT4 = CONT_MAX4 THEN
95         CONT4 <= "000000000000000000";
96     END IF;
97 END IF;
98 END PROCESS;
99
100 ----- FA -----
101
102 PROCESS (CLK)
103 BEGIN
104 IF CLK'EVENT AND CLK = '1' THEN
105     CONT5 <= CONT5 + 1;
106     IF CONT5 = CONT_MAX5 THEN
107         CONT5 <= "000000000000000000";
108     END IF;
109 END IF;
110 END PROCESS;
111

```

```

112 ----- SI -----
113
114 PROCESS (CLK)
115 BEGIN
116 IF CLK'EVENT AND CLK = '1' THEN
117     CONT6 <= CONT6 + 1;
118     IF CONT6 = CONT_MAX6 THEN
119         CONT6 <= "000000000000000000";
120     END IF;
121 END IF;
122 END PROCESS;

```

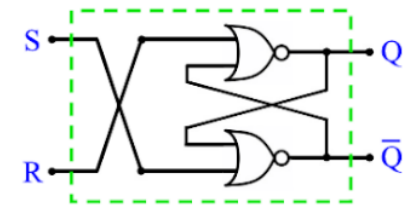
En esta parte se genera el divisor de frecuencia para cada uno de los “CLK” correspondiente a su nota musical en donde oscilaran constantemente cuando el CLK = ‘0’ (dada las características de la FPGA esta invertido por eso se muestra que CLK = 1) hasta llegar a su valor máximo, después de llegar a su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

```

123 -----LATCH SR-----
124
125 PROCESS (R, Q, NQ)
126 BEGIN
127     Q<= NQ NOR R;
128     NQ<= Q NOR NOT (R);
129
130 END PROCESS;
131

```

En esta parte se establece la creación del “latch SR” el cual nos servirá para determinar la salida del sonido (izquierda o derecha) conforme a la tabla de verdad... en este caso se utiliza el “latch SR” que se conecta con “NOR”.



| R | S | Q | Q' |
|---|---|--------------------|----|
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 0 | Q | Q' |
| 1 | 1 | Estado no-definido | |

```

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

```

Aquí es donde se establecen dos condicionales... uno es quien determina la salida del sonido dependiendo del valor de Q (se representa como el switch de un polo y dos disparos que se encuentra en el diagrama dada por el profesor), si Q = ‘1’ la salida del sonido se ejecutara a la izquierda. Posteriormente el segundo condicional es el selector “S” ya que 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.

```

163 ELSE
164     IF (S="000") THEN --DO
165         SONIDOD <= CONT1(17);
166         SEG <="0110001";
167     ELSIF S = "001" THEN --RE
168         SONIDOD <= CONT4(17);
169         SEG <="1000010";
170     ELSIF S = "010" THEN --MI
171         SONIDOD <= CONT2(17);
172         SEG <="0110000";
173     ELSIF S = "011" THEN --FA
174         SONIDOD <= CONT5(17);
175         SEG <="0111000";
176     ELSIF S = "100" THEN --SOL
177         SONIDOD <= CONT3(16);
178         SEG <="0100000";
179     ELSIF S = "101" THEN --LA
180         SONIDOD <= CONT (16);
181         SEG <="0001000";
182     ELSIF S = "110" THEN --SI
183         SONIDOD <= CONT6(16);
184         SEG <="1100000";
185     ELSE
186         SONIDOD <= '1';
187         SEG <="1111111";
188     END IF;
189
190 END IF;
191
192 END PROCESS;
193
194
195 END GeneraSonido;

```

Finalmente, si Q = ‘0’ la salida del sonido se ejecutara a la derecha. Posteriormente el segundo condicional es el mismo selector “S” (que se encuentra en ambos casos sin importar el valor de Q) ya que 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.