

Universidad de Guadalajara



Seminario de problemas de programación de
sistemas reconfigurables

Diseño e implementación de un contador asíncrono

Nombre:

Muñoz Nuñez Ian Emmanuel

Sección: D01

Código: 216464457

Maestra:

María Patricia Ventura Nuñez

Ingeniería Robótica

1. Objetivo

Solucionar problemas de diseño utilizando las herramientas aprendidas en programación de sistemas reconfigurables.

Simular circuitos digitales en programas de diseño como *Proteus®* e implementarlos físicamente.

Diseño e implementación de un contador asíncrono, por ejemplo, contadores de código *BCD*'s.

2. Material

- 2 Protoboard.
- Fuente VCC (5V).
- Resistencias de 200Ω y $2k\Omega$.
- Dip-switch de 4 bits.
- 2 Display's de 7 segmentos.
- 2 decodificadores *BCD a 7 segmentos*.
- 1 *GAL22v10D*.

3. Marco teórico

3.1. Secuencias

$$x = 0 \rightarrow 1, 2, 3, 4, 6, 8, 9, 12, 13, 14, 15$$

$$x = 1 \rightarrow 15, 13, 11, 9, 7, 5, 3, 1$$

3.2. Código desarrollado

Listing 1: Código desarrollado en *WinCupl*

```
1 Name      contadorEstados ;
2 PartNo    00 ;
3 Date      10/18/2022 ;
4 Revision  01 ;
5 Designer  ian ;
6 Company   UdeG ;
7 Assembly  None ;
8 Location  ;
9 Device    g22v10 ;
10
11 /* CONTADOR X=0 --> 0-12, SI X = 1 --> IMPARES */
12
13 /* ENTRADAS */
14
15 PIN 1=CK;
16 PIN 2=X;
17
18 /* SALIDAS */
19 PIN 23=QA;
20 PIN 22=QB;
21 PIN 21=QC;
22 PIN 20=QD;
23 PIN 19=QE;
24
25 FIELD ESTADOS=[QA, QB, QC, QD, QE];
26 $DEFINE S0 'b' 00000
27 $DEFINE S1 'b' 00001
28 $DEFINE S2 'b' 00010
29 $DEFINE S3 'b' 00011
30 $DEFINE S4 'b' 00100
31 $DEFINE S5 'b' 00101
32 $DEFINE S6 'b' 00110
33 $DEFINE S7 'b' 00111
34 $DEFINE S8 'b' 01000
35 $DEFINE S9 'b' 01001
36 $DEFINE S11 'b' 10001
37 $DEFINE S12 'b' 10010
38 $DEFINE S13 'b' 10011
39 $DEFINE S14 'b' 10100
40 $DEFINE S15 'b' 10101
41
42 SEQUENCE ESTADOS {
43 PRESENT S0
44 IF !X NEXT S1;
45 IF X NEXT S15;
46
```

```
47 PRESENT S1
48 IF !X NEXT S2;
49 IF X NEXT S15;
50
51 PRESENT S2
52 IF !X NEXT S3;
53 IF X NEXT S1;
54
55 PRESENT S3
56 IF !X NEXT S4;
57 IF X NEXT S1;
58
59 PRESENT S4
60 IF !X NEXT S6;
61 IF X NEXT S1;
62
63 PRESENT S5
64 IF !X NEXT S1;
65 IF X NEXT S3;
66
67 PRESENT S6
68 IF !X NEXT S8;
69 IF X NEXT S1;
70
71 PRESENT S7
72 IF !X NEXT S1;
73 IF X NEXT S5;
74
75 PRESENT S8
76 IF !X NEXT S9;
77 IF X NEXT S1;
78
79 PRESENT S9
80 IF !X NEXT S12;
81 IF X NEXT S7;
82
83 PRESENT S11
84 IF !X NEXT S1;
85 IF X NEXT S9;
86
87 PRESENT S12
88 IF !X NEXT S13;
89 IF X NEXT S1;
90
91 PRESENT S13
92 IF !X NEXT S14;
93 IF X NEXT S11;
94
95 PRESENT S14
96 IF !X NEXT S15;
```

```
97 IF X NEXT S1;  
98  
99 PRESENT S15  
100 IF !X NEXT S0;  
101 IF X NEXT S13;  
102 }
```

4. Procedimiento

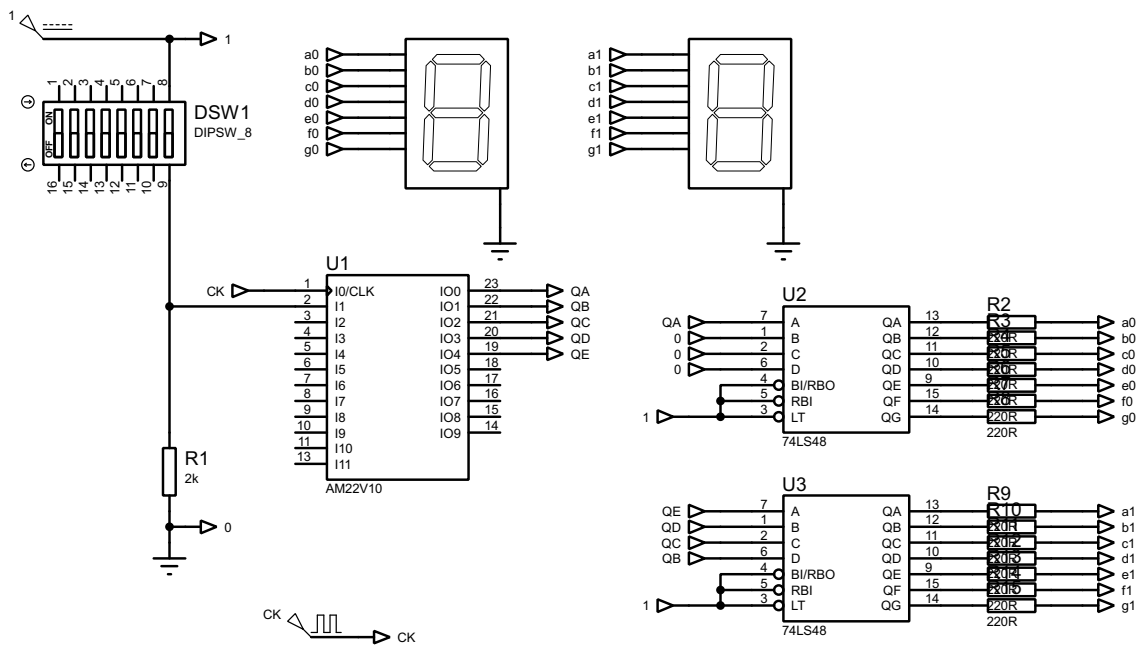
Para desarrollar este proyecto primero se hicieron las conexiones entre la *GAL*, los decodificadores y los display's, para después hacer el código en *WinCupl* y cargarlo a la *GAL*.

Los materiales utilizados para este proyecto son: 1 dip-switch de 4 bits, 14, resistencias de 220Ω y una de $2k\Omega$, dos decodificadores *BCD* a 7 segmentos, 2 display's de 7 segmentos y una *GAL22v10D*.

5. Circuito a implementar

5.1. Simulación

En la siguiente página se muestra el diseño del circuito en simulación.



Contador de estados con gal22v10

Materia: Seminario de problemas de programacion de sistemas reconfigurables

Maesta: Maria Patricia Ventura Nunez

Seccion: D01

Alumno: Munoz Nunez Ian Emmanuel

Codigo: 216464457

Alumno: Ruvalcaba Ramos Sergio Alejandro

Codigo: 216577855

5.2. Protoboard

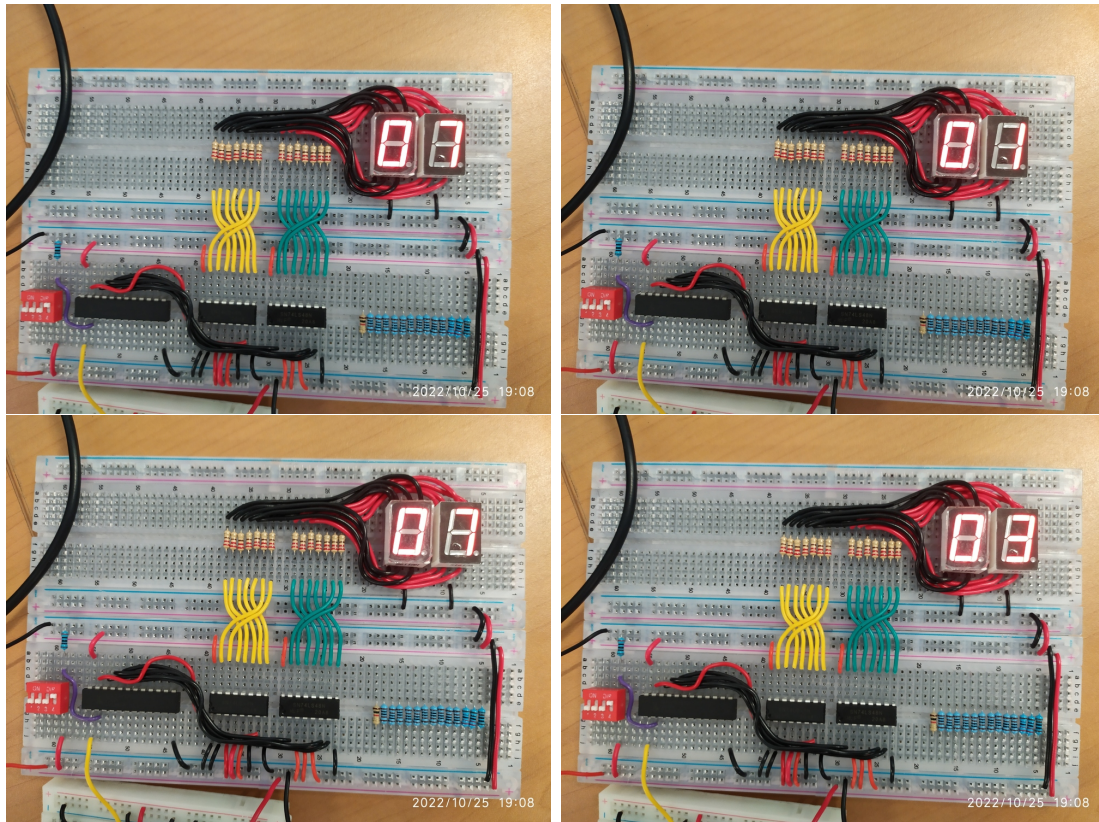


Figura 1: Circuito en protoboard

6. Conclusión

Usar una *GAL* en lugar de *Flip-Flop's* hace más fáciles las cosas, pues no ocupa mucho espacio y todo se realiza a través de software, pero al principio es un poco complicado entender como funcionan los estados en la *GAL*, pero al final es mejor desarrollar los estados con software que con hardware.