Universidad de Guadalajara

Seminario de problemas de programación de sistemas reconfigurables

Proyecto 2

Diseño de un circuito combinacional que genere un bit de paridad impar, utilizando compuertas *or exclusivas* y/o *nor exclusiva*

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*.

Utilizar hojas de datos de las familias lógicas.

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

Diseñar e implementar una función utilizando compuertas *orex* y *norex*. Por ejemplo:

- Un detector de paridad par y/o impar.

- Un sumador o restador de 4 bits.

2. Material

- Protoboard.
- Fuente VCC (5V).
- Led verde.
- Resistencias de 200Ω y 2$k\Omega$.
- 1 dip switch de 8 bits.
- 1 display de 7 segmentos.
- 1 barra de leds de 8 bits.
- 2 74LS04 $\rightarrow$ inversor o compuerta lógica *not*.
- 2 74LS08 $\rightarrow$ compuerta lógica *and* de 2 entradas.
- 3 74LS86 $\rightarrow$ compuerta lógica *xor* de 2 entradas.
- 1 74LS48 $\rightarrow$ decodificador BCD-7 segmentos.

3. Marco teórico

BIT de paridad par: Es el bit que se agrega para que la información en binario tenga un numero par de 1's.

BIT de paridad impar: Es el bit que se agrega para que la información en binario tenga un numero impar de 1's.

Diseñar un *generador de paridad impar* para la salida de un decodificador BCD-7 segmentos.

3.1. Generador de paridad impar para los números del 0 al 15

En la tabla 1 se muestra la tabla de verdad de un generador impar de 4 entradas.

	A	B	C	D	F1
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

Tabla 1: Generador de paridad impar

$$F1 = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C\,D + \overline{A}\,B\,\overline{C}\,D + \overline{A}\,B\,C\,\overline{D} + A\,\overline{B}\,\overline{C}\,D + A\,\overline{B}\,C\,\overline{D} + A\,B\,\overline{C}\,\overline{D} + A\,B\,C\,D \rightarrow xor$$

$$F1 = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C\,D + (\overline{A}\,B\,(C \oplus D)) + (A\,\overline{B}\,(C \oplus D)) + A\,B\,\overline{C}\,\overline{D} + A\,B\,C\,D \rightarrow 12a$$

$$F1 = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C\,D + C \oplus D\,(\overline{A}\,B + A\,\overline{B}) + A\,B\,\overline{C}\,\overline{D} + A\,B\,C\,D \rightarrow xor$$

$$F1 = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C\,D + (C \oplus D)\,(A \oplus B) + A\,B\,\overline{C}\,\overline{D} + A\,B\,C\,D \rightarrow 12a$$

$$F1 = C\,D\,(\overline{A}\,\overline{B} + A\,B) + \overline{C}\,\overline{D}\,(\overline{A}\,\overline{B} + A\,B) + (A \oplus B)\,(C \oplus D) \rightarrow xnor$$

$$F1 = C\,D\,(\overline{A \oplus B}) + \overline{C}\,\overline{D}\,(\overline{A \oplus B}) + (A \oplus B)\,(C \oplus D) \rightarrow 12a$$

$$F1 = (\overline{A \oplus B})\,(\overline{C}\,\overline{D} + C\,D) + (A \oplus B)\,(C \oplus D) \rightarrow xnor$$

$$F1 = (\overline{A \oplus B})\,(\overline{C \oplus D}) + (A \oplus B)\,(C \oplus D) \rightarrow xnor$$

$$\boxed{F1 = \overline{(A \oplus B) \oplus (C \oplus D)}}$$

A partir de la ecuación resultante, se puede intuir que para el generador de paridad impar del decodificador se pueden utilizar solo compuertas *xnor* o *xor* y usar un inversor al final.

3.2. Multiplicador 2x2 bits

$$M3 = (W\,Y)\,(X\,Z)$$
$$M2 = W\,Y\,\overline{X\,Z}$$
$$M1 = X\,Y \oplus W\,Z$$
$$M0 = X\,Z$$

4. Procedimiento

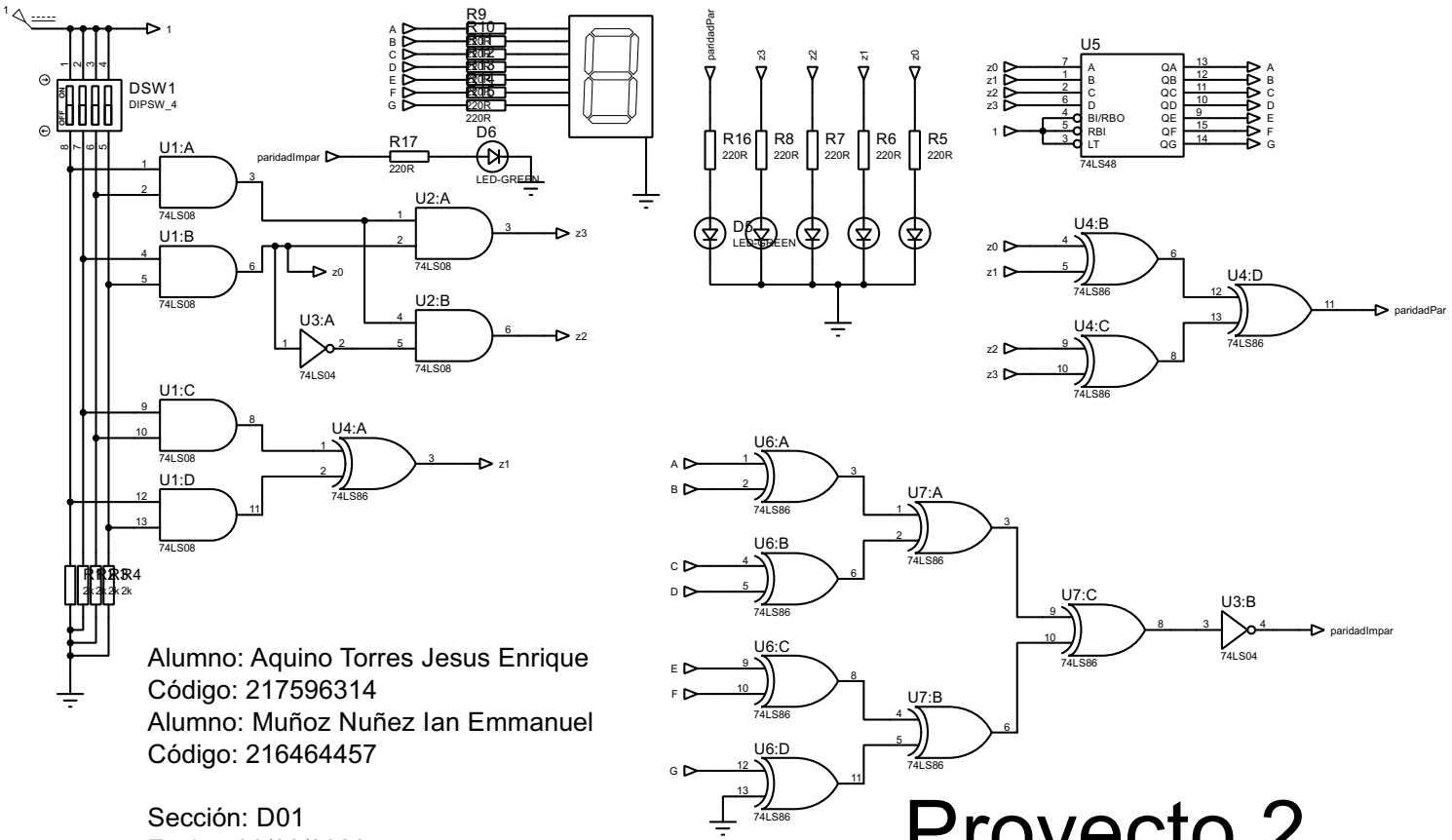
Para realizar el proyecto, primero se hizo el multiplicador de 2x2 bits y se probo que funcionara. Luego se creo el generador de paridad par para el el multiplicador.

Luego de tener el generador de paridad par se hicieron las conexiones del decodificador con el multiplicador y con el display de 7 segmentos. Después de probar el decodificador se hizo el generador de paridad impar para las salidas de este. Ya que se encontraron problemas con las compuertas *xnor*, se decidió por usar compuertas *xor* y negar la salida del generador.

Los materiales que se utilizaron para este proyecto son: 1 dipswitch de 8 bits, 1 led, 13 resistencias de 220Ω y 4 resistencias de $2k\Omega$, y display de 7 segmentos, una barra de leds de 8 bits, 2 inversores o compuertas *not*, 2 compuertas *and* de 2 entradas, 3 compuertas *xor* de 2 entradas y un decodificador BCD-7 segmentos.

4.1. Simulación

En la siguiente página se puede ver el diseño de la simulación hecha en *Proteus™*



Alumno: Aquino Torres Jesus Enrique
Código: 217596314
Alumno: Muñoz Nuñez Ian Emmanuel
Código: 216464457

Sección: D01
Fecha: 30/08/2022
Materia: Seminario de problemas de programación de sistemas reconfigurables
Maestra: Maria Patricia Ventura Nuñez

Proyecto 2

Diseño de un circuito combinacional que genere un bit de paridad, utilizando compuertas xor y/o xnor

4.2. Protoboard

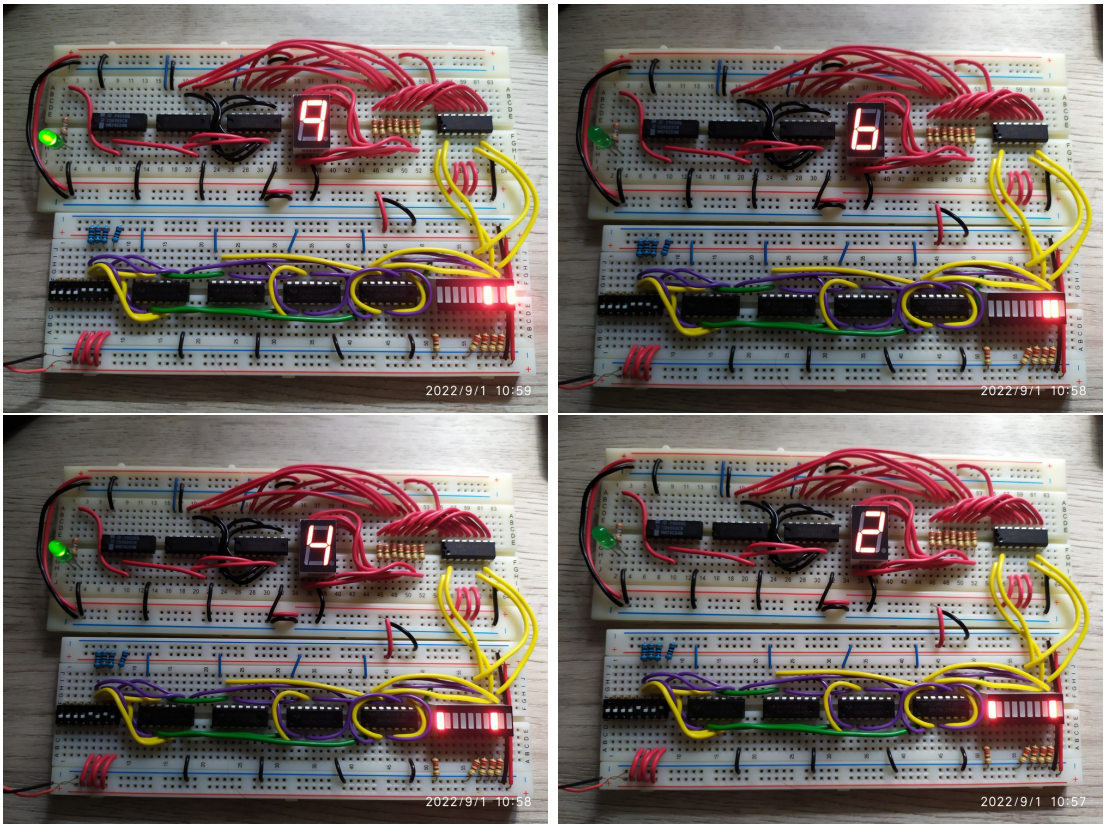


Figura 1: Circuito en protoboard

5. Conclusión

Considero que saber como funcionan los generadores de paridad es algo muy importante, pero hubo problemas al realizar el generador de paridad impar, pues la idea original era usar las compuertas lógicas *xnor* para realizar el generador, pero se tuvieron problemas con estas, tanto las de la familia *TTL* como de la *CMOS*, por lo que mejor se decicidio usar compuertas *xor* y solamente negar la salida del generador.