Documentación Ejercicio 3

Ej. 3: Decodificador hex-to-7-segments

Marco teórico

Multiplexor

El uso de un multiplexor consiste fundamentalmente en la selección de una de varias entradas en un dispositivo. Haciendo uso de un puerto de control de selección se determina con valor binario el número de la entrada que ha ser la salida del dispositivo. Para el caso en cuestión, el bloque de un multiplexor puede apreciarse en la Figura

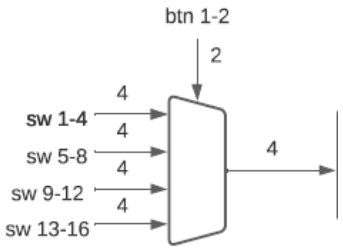

Figura_1 Funcionamiento de un multiplexor

Decodificador de 7 segmentos BCD

Un decodificador de 7 segmentos BCD convencional consiste en un dispositivo lógico que recibe un valor binario en sus entradas y da un resultado combinacional correspondiente a la misma, esta salida ha de corresponder a al número de la entrada para que pueda ser interpretado por un display de 7 segmentos.

Display de 7 segmentos.

Es un dispositivo electrónico que se utiliza convencionalmente para mostrar números haciendo uso de luces led. Existen 2 tipos, de ánodo común y de cátodo común. Cada uno con una lógica opuesta para su funcionamiento, uno funciona con ceros lógicos para encender una luz y el otro con unos lógicos para encender una luz. Un ejemplo de un display de 7 segmentos se puede apreciar en la Figura

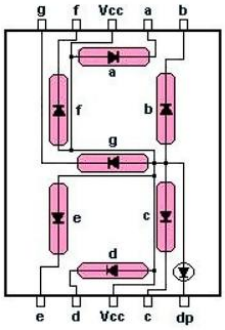

Figura_2 Funcionamiento de un display de 7 segmentos

Tabla de la verdad para un decodificador de 7 segmentos

Teniendo en cuenta el funcionamiento de un decodificador y display de 7 segmentos se puede realizar la siguiente tabla de la verdad con el fin apreciar el funcionamiento esperado principalmente por parte del decodificador BCD a 7 segmentos

	Entradas					Salidas						
N	D	C	B	A		a	b	c	d	e	f	g
0	0	0	0	0		1	1	1	1	1	1	0
1	0	0	0	1		0	1	1	0	0	0	0
2	0	0	1	0		1	1	0	1	1	0	1
3	0	0	1	1		1	1	1	1	0	0	1
4	0	1	0	0		0	1	1	0	0	1	1
5	0	1	0	1		1	0	1	1	0	1	1
6	0	1	1	0		1	0	1	1	1	1	1
7	0	1	1	1		1	1	1	0	0	0	0
8	1	0	0	0		1	1	1	1	1	1	1
9	1	0	0	1		1	1	1	0	0	1	1

Descripción del código a implementar

Se pretende ahora realizar una interconexión de distintos módulos tales que cumplan el siguiente funcionamiento, a continuación, se procede a hacer una descripción de cada uno de los módulos del código

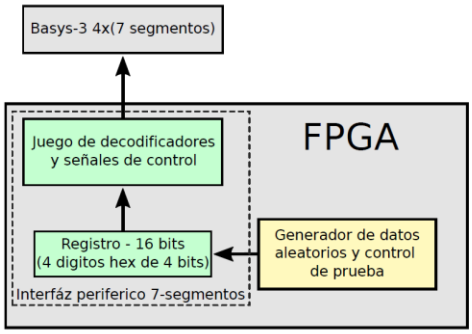

Figura 3: Diagrama conceptual de la interfaz de display de 7 segmentos.

Módulos 1: TOP

Este módulo contiene las entradas principales del sistema, los cables que conectan las distintas entradas y salidas de los módulos internos. En este módulo también se hacen las instancias del resto de módulos para luego ser probadas en la testbench. El módulo se puede apreciar a continuación.

```
module TOP (

    input   logic        rst,
    input   logic        clk,
    input   logic        write,
    input   logic        select,
    output  logic [15:0] aleatorio,
    output  logic [3:0] Hexa1,
    output  logic [3:0] Hexa2,
    output  logic [3:0] Hexa3,
    output  logic [3:0] Hexa4,
    output  logic [6:0] Result1,
    output  logic [6:0] Result2,
    output  logic [6:0] Result3,
    output  logic [6:0] Result4
);

// Cables

    wire [15:0] GenRes ;
    wire [3:0]  Display1 ;
    wire [3:0]  Display2 ;
    wire [3:0]  Display3 ;
    wire [3:0]  Display4 ;
```

Figura_4: Módulo TOP

Módulos 2: Generador

Este módulo tiene las entradas necesarias para generar el número pseudoaleatorio. Utiliza una asignación especial

```
    aleatorio_sal <= 16'hACE1;
else if(write && select)
        aleatorio_sal <= {aleatorio_sal[14:0],aleatorio_sal[15]^aleatorio_sal[13]^aleatorio_sal[12]^aleatorio_sal[10]};
```

Figura_5: Valor pseudoaleatorio

De manera que se seleccionen datos aleatorios a partir de un valor base, el módulo se puede apreciar en la Figura_6

```
module Generator (
    input   logic        rst,
    input   logic        clk,
    input   logic        write,
    input   logic        select,
    output  logic [15:0] aleatorio
);
```

Figura_6: Módulo generador

Módulos 3: HEX

Este módulo recibe el dato generado de manera aleatoria de 16 bits y lo divide en 4 números distintos de 4 bits cada uno, para concepto de la demostración solo se utiliza 1 de los 4 números. El módulo HEX se puede apreciar a continuación

```
module HEX (
    input  logic [15:0] binario,
    output logic  [3:0] numero1
    //output logic  [3:0] numero2,
    //output logic  [3:0] numero3,
    //output logic  [3:0] numero4,
    //output logic  [3:0] Hexa1,
    //output logic  [3:0] Hexa2,
    //output logic  [3:0] Hexa3,
    //output logic  [3:0] Hexa4
);
```

Figura_7: Módulo HEX

Módulos 4: DIS

Este módulo consiste a grandes rasgos en una maquina de estado para los display, en función de cada uno de los valores numéricos en cada entrada y siguiendo la tabla de la verdad definida en el marco teórico selecciona el valor del display de 7 segmentos correspondiente según el valor de la entrada, esto lo hace para cada uno de los 4 números, pero nuevamente, por conceptos de la demostración solo se selecciona una de las entradas para obtener el valor de 1 de los display. Este módulo se puede ver a continuación

```
module DIS (
    input              Enable,
    input  logic [3:0] Dis1,
    //input  logic [3:0] Dis2,
    //input  logic [3:0] Dis3,
    //input  logic [3:0] Dis4,

    output logic [6:0] Out1
    //output logic [6:0] Out2,
    //output logic [6:0] Out3,
    //output logic [6:0] Out4
);
```

Figura_8: Módulo DIS

Funcionamiento

El funcionamiento en este caso depende únicamente de las entradas de control, ya que los valores de los números se generan de manera aleatoria. Estos valores de control que se pueden apreciar en la Figura_9 determinan si el código arranca o no

```
initial begin
    Ena = 0;
    rst = 1;
    write = 1;
    select = 1;
    #10;
    Ena = 1;
    rst = 0;
    write = 1;
    select = 1;
    #70

    $finish();

end
```

Figura_9: Inicialización

Una vez realizada la inicialización y todos los datos son correctos para iniciar siguiendo las buenas prácticas, se espera que para la simulación se obtengan valores aleatorios en la salida “aleatorio”, mientras que en los dichos valores de Result# se obtengan los valores provenientes de la maquina de estados necesarios para activar los valores correspondientes en los display de 7 segmentos. Por otra parte el resultado obtenido en Hexa# corresponde al valor decimal para una mejor interpretación

```
# ____________________________________________________
# En el momento: 25000 | el valor generado aleatoriamente es: 0100101011011111
# En el momento: 25000 | Display1: 1001100 | Display2: 0001000 | Display3: 1000010 | Display4: 0111000
# En el momento: 25000 | Parte1: 4 | Parte2: a | Parte3: d | Parte4: f
# ____________________________________________________
# En el momento: 35000 | el valor generado aleatoriamente es: 1101100100110011
# En el momento: 35000 | Display1: 1000010 | Display2: 0001100 | Display3: 0000110 | Display4: 0000110
# En el momento: 35000 | Parte1: d | Parte2: 9 | Parte3: 3 | Parte4: 3
# ____________________________________________________
# En el momento: 45000 | el valor generado aleatoriamente es: 1010111000011110
# En el momento: 45000 | Display1: 0001000 | Display2: 0110000 | Display3: 1001111 | Display4: 0110000
# En el momento: 45000 | Parte1: a | Parte2: e | Parte3: 1 | Parte4: e
# ____________________________________________________
# En el momento: 55000 | el valor generado aleatoriamente es: 1110101001001100
# En el momento: 55000 | Display1: 0110000 | Display2: 0001000 | Display3: 1001100 | Display4: 0110001
# En el momento: 55000 | Parte1: e | Parte2: a | Parte3: 4 | Parte4: c
```

Figura_10: Resultados

Para una mejor visualización del funcionamiento ver el video de demostración