Proyecto

Objetivo: Implementar un programa mediante shell para manipular las entradas de un convertidor de código y ver el resultado mediante un laboratorio remoto.

El convertidor de código tiene 3 entradas, las cuales se representan mediante un vector de 3 bits, y 8 salidas, las cuales se representan mediante un vector de 8 bits. Esto se muestra en la figura 1.

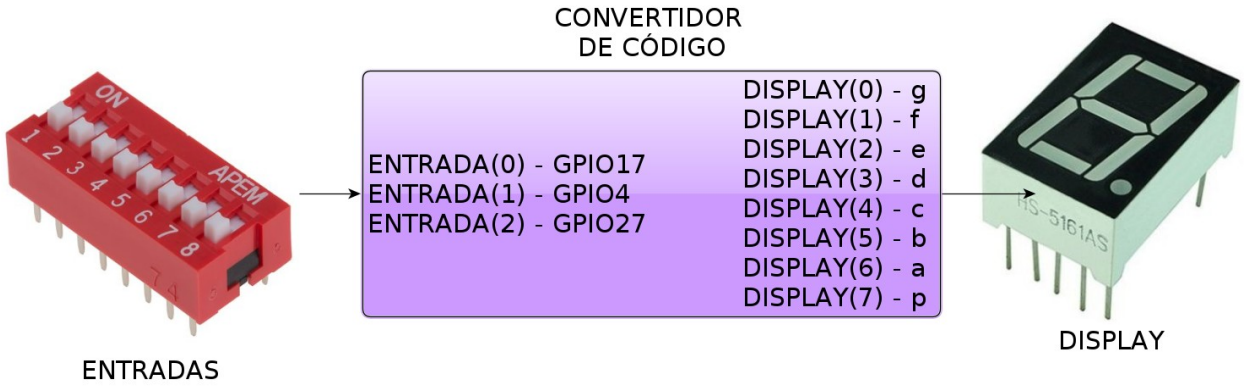

Figura 1: Arquitectura del convertidor de código.

El convertidor de código muestra en un display de ánodo común los dígitos del 0 al 7. Esto se muestra en la figura 2.

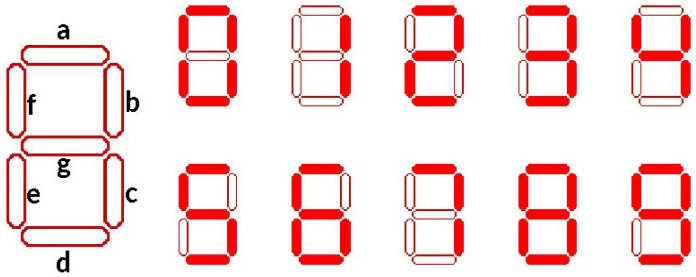

Figura 2: Dígitos que se pueden mostrar en el display

En este tipo de display se debe colocar un cero binario para encender los segmentos y un uno binario para apagar los segmentos.

El funcionamiento del convertidor de código se muestra en la tabla 1.

ENTRADAS			DISPLAY									
E(2) GPIO27	E(1) GPIO4	E(0) GPIO17	p	a	b	c	d	e	f	g	CÓDIGO	DIGITO
0	0	0	0	0	0	0	0	0	0	1	0X01	0
0	0	1	0	1	0	0	1	1	1	1	0X4F	1
0	1	0	0	0	0	1	0	0	1	0	0X12	2
0	1	1	0	0	0	0	0	1	1	0	0X06	3
1	0	0	0	1	0	0	1	1	0	0	0X4C	4
1	0	1	0	0	1	0	0	1	0	0	0X24	5
1	1	0	0	0	1	0	0	0	0	0	0X20	6
1	1	1	0	0	0	0	1	1	1	1	0X0F	7

Tabla 1: Operación del convertidor de código

El programa en VHDL que implementa este diseño se muestra en la figura 3.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity CONVCOD is
5      GENERIC( N : INTEGER := 8 );
6      Port ( ENTRADA : in STD_LOGIC_VECTOR (2 downto 0);
7             LEDS : OUT STD_LOGIC_VECTOR (2 downto 0);
8             AN : OUT STD_LOGIC_VECTOR (N-1 downto 0);
9             DISPLAY : out STD_LOGIC_VECTOR (N-1 downto 0));
10 end CONVCOD;
11
12 architecture PROGRAMA of CONVCOD is          --PABCDEFG
13 CONSTANT DIG0 : STD_LOGIC_VECTOR(N-1 DOWNTO 0) := X"01";
14 CONSTANT DIG1 : STD_LOGIC_VECTOR(N-1 DOWNTO 0) := X"4F";
15 CONSTANT DIG2 : STD_LOGIC_VECTOR(N-1 DOWNTO 0) := X"12";--
16 CONSTANT DIG3 : STD_LOGIC_VECTOR(N-1 DOWNTO 0) := X"06";
17 CONSTANT DIG4 : STD_LOGIC_VECTOR(N-1 DOWNTO 0) := X"4C";
18 CONSTANT DIG5 : STD_LOGIC_VECTOR(N-1 DOWNTO 0) := X"24";
19 CONSTANT DIG6 : STD_LOGIC_VECTOR(N-1 DOWNTO 0) := X"20";
20 CONSTANT DIG7 : STD_LOGIC_VECTOR(N-1 DOWNTO 0) := X"0F";
21 begin
22     LEDS <= ENTRADA;
23     AN <= X"FE";
24
25     DISPLAY <= DIG0 WHEN( ENTRADA = "000" )ELSE
26                DIG1 WHEN( ENTRADA = "001" )ELSE
27                DIG2 WHEN( ENTRADA = "010" )ELSE
28                DIG3 WHEN( ENTRADA = "011" )ELSE
29                DIG4 WHEN( ENTRADA = "100" )ELSE
30                DIG5 WHEN( ENTRADA = "101" )ELSE
31                DIG6 WHEN( ENTRADA = "110" )ELSE
32                DIG7;
33 end PROGRAMA;
```

Figura 3: Programa en VHDL del convertidor de código

Para probar este convertidor de código se deben manipular las entradas del diseño. Estas entradas están conectadas a tres GPIOS del sistema embebido que son GPIO17, GPIO4, GPIO27, tal como se muestra en la figura 1 y tabla 1. Para manipular los GPIOS se debe realizar un programa en shell, el cual debe realizar cuatro funciones básicas:

1. Habilitar el GPIO.
2. Configurar la dirección del GPIO.
3. Colocar los valores digitales deseados en el GPIO.
4. Deshabilitar el GPIO.

Estas funciones se realizan en el carpeta /*sys*/ de Linux. /sys/ es un directorio virtual el cual contiene información del sistema usada por el kernel de linux. Aqui se encuentra el acceso al device drivers de GPIOs.

Habilitar el GPIO.

Esto se realiza con la carpeta ubicada en /sys/class/gpio/export. “export” sirve para exportar el control del GPIO desde el kernel al espacio de usuario, escribiendo su número al archivo. Ejemplo:

```
$ echo “17” > /sys/class/gpio/export
```

Configurar la dirección del GPIO.

Después de habilitar el GPIO se crea la carpeta ubicada en /sys/class/gpio/gpio17/direction. Aquí se escribe “in” para configurar el GPIO como entrada y “out” para configurar el GPIO como salida. Ejemplo:

```
$ echo “out” > /sys/class/gpio/gpio17/direction
```

Colocar los valores digitales deseados en el GPIO.

Después de habilitar el GPIO se crea la carpeta ubicada en /sys/class/gpio/gpio17/value. Aquí se escribe “1” para enviar un uno lógico y “0” para enviar un cero lógico. Ejemplo:

```
$ echo “1” > /sys/class/gpio/gpio17/value
$ echo “0” > /sys/class/gpio/gpio17/value
```

Deshabilitar el GPIO.

Esto se realiza con la carpeta ubicada en /sys/class/gpio/unexport. “unexport”. Revierte el efecto de exportar al espacio de usuario, escribiendo su número al archivo. Ejemplo:

```
$ echo “17” > /sys/class/gpio/unexport
```

Un programa que realiza la configuración de los GPIOS: GPIO17, GPIO4, GPIO27 se muestra en la figura 4.

```
#!/bin/bash

echo "Habilitando terminales en export"
echo "17" > /sys/class/gpio/export
sleep 1
echo "4" > /sys/class/gpio/export
sleep 1
echo "27" > /sys/class/gpio/export
sleep 1

echo "Configurando terminales como salidas"
echo "out" > /sys/class/gpio/gpio17/direction
echo "out" > /sys/class/gpio/gpio4/direction
echo "out" > /sys/class/gpio/gpio27/direction

echo "Mandando valor de entrada al convertidor"
echo "1" > /sys/class/gpio/gpio17/value
echo "0" > /sys/class/gpio/gpio4/value
echo "1" > /sys/class/gpio/gpio27/value
sleep 5

echo "Deshabilitando terminales en unexport"
echo "17" > /sys/class/gpio/unexport
echo "4" > /sys/class/gpio/unexport
echo "27" > /sys/class/gpio/unexport

exit 0
```

Figura 4: Script de shell para configurar los GPIOS

Para probar el programa de figura 4, primero debemos programar el archivo convcod.bit en el sistema de desarrollo usado en el laboratorio remoto. Esto se hace de la siguiente forma:

```
$ progFPGA.sh convcod.bit
```

Posteriormente se accede al servidor de video, esto se realiza colocando en el navegador la dirección IP del servidor con el puerto 8081. Ejemplo, para la dirección IP 148.204.58.65 se coloca:

http://148.204.58.65:8081

Después se introduce el usuario y contraseña del servidor que se muestra en la tabla 2.

Esta contraseña es la misma para todos los sistemas.

	Servidor de video
Código QR:	
Usuario:	escom
Contraseña:	labremoto

Tabla 2: Credenciales del servidor de video

Finalmente el estudiante debe completar el programa de la figura 4 para enviar todos los valores de entrada, mostrados en la tabla 1, para visualizar los dígitos del 0 al 7 en el display.