**LABORATORIO DE SISTEMAS ELECTRÓNICOS INDUSTRIALES**

**CURSO 2021/22**

**DESCIFRA EL CÓDIGO**

**Integrantes:**

Gómez-Pamo González-Cela, Jorge 54637

González Alday, Javier Pío 54639

González Denia, Adrián 54647

**Grupo de clase:** A404

**ÍNDICE**

[Introducción 4](#_Toc91704698)

[Explicación y normas del juego 4](#_Toc91704699)

[Componentes del sistema 4](#_Toc91704700)

[Fotos del proyecto 4](#_Toc91704701)

[Estructura del código 5](#_Toc91704702)

[Jerarquía de ficheros del proyecto 5](#_Toc91704703)

[DESCIFRA\_EL\_CODIGO\_TOP 6](#_Toc91704704)

[Descripción del componente 6](#_Toc91704705)

[Estructura interna del componente (diagrama RTL) 6](#_Toc91704706)

[BOTON\_TOP 7](#_Toc91704707)

[Descripción del componente 7](#_Toc91704708)

[Estructura interna del componente (diagrama RTL) 7](#_Toc91704709)

[Componentes internos 7](#_Toc91704710)

[FSM\_TOP 8](#_Toc91704711)

[Descripción del componente 8](#_Toc91704712)

[Estructura interna del componente (diagrama RTL) 8](#_Toc91704713)

[Componentes internos 8](#_Toc91704714)

[FSM\_MASTER 8](#_Toc91704715)

[FSM\_INCHECK 8](#_Toc91704716)

[FSM\_SHOWSEQ 8](#_Toc91704717)

[FSM\_LFSR 8](#_Toc91704718)

[FSM\_SLAVE\_TIMER 8](#_Toc91704719)

[VISUALIZER\_TOP 9](#_Toc91704720)

[Descripción del componente 9](#_Toc91704721)

[Estructura interna del componente (diagrama RTL) 9](#_Toc91704722)

[Componentes internos 9](#_Toc91704723)

[Conclusiones e impresiones 10](#_Toc91704724)

# Introducción

## Explicación y normas del juego

## Componentes del sistema

## Fotos del proyecto

# Estructura del código

## Jerarquía de ficheros del proyecto

A continuación se muestra el árbol de ficheros del proyecto. En el se muestran todos los componentes que constituyen el sistema del juego:

Texto

Descripción generada automáticamente con confianza baja

Como puede comprobarse, la entidad general es DESCIFRA\_EL\_CODIGO\_TOP. Esta está definida internamente por otros tres componentes diferentes:

* **BOTON\_TOP:** Entidad encargada de la recepción y coordinación de la señales producidas por las pulsaciones de los botones de la placa. Al utilizar los 5 botones disponibles, cada uno es tratado por separado por una instancia diferente de la misma entidad, BOTON\_DEBOUNCER.
* **FSM\_TOP:** Entidad dedicada a la coordinación de los eventos y fases del juego. Está compuesta por una máquina de estado maestra (FSM\_MASTER) encargada de determinar las fases del juego, y varias esclavas encargadas de coordinar los eventos e interacciones con el jugador.
* **VISUALIZER\_TOP:** Entidad encargada de la muestra de mensajes a través de los displays de 7 segmentos integrados en la placa. Estos mensajes le indicarán al jugador en la fase en la que se encuentra (inicio del juego, si ha ganado o perdido, etc…).

A lo largo del los siguientes puntos se realizará una descripción más detallada de cada componente por separado.

## DESCIFRA\_EL\_CODIGO\_TOP

### Descripción del componente

Como se ha indicado anteriormente, esta entidad es la encargada de juntar de forma general todos los componentes necesarios para el funcionamiento del juego. De esta forma, es posible simplificar la estructura general del sistema para considerarlo una “caja negra” con las siguientes entradas y salidas:

* **BTN:** Vector de 5 elementos de entrada. Cada elemento se corresponde con el valor de la señal de cada uno de los 5 botones de la placa (OK, UP, DOWN, LEFT, RIGHT).
* **CLK100MHZ:** Entrada la de señal de reloj de 100 MHz. Utilizada por todos los componentes del sistema.
* **CPU\_RESETN:** Entrada de la señal de RESET del sistema. Es activa a nivel bajo.
* **anode:** Salida de un vector de 8 elementos. Utilizado para indicar el display de 7 segmentos en el que se desea encender. Cada elemento corresponde a un display diferente.
* **segment:** Salida de un vector de 7 elementos. Utilizado para indicar los segemntos ha encender en cada display según el mensaje que se quiera mostrar.
* **JA:** Salida de un vector de 4 elementos. Estos corresponden a 4 de los puertos digitales laterales dela placa. Estos son utilizados para controlar el encendido y apagado de los LEDs externos que muestra la secuencia que el jugador tendría que haber introducido para ganar.

### Estructura interna del componente (diagrama RTL)

El código de esta entidad es del estilo estructural, ya que se interconectan los distintos componentes, entradas y salidas mediante señales. El código es el siguiente:

De esta forma, el diagrama de bloques RTL de la entidad quedará de la siguiente forma:

Gráfico

Descripción generada automáticamente

## BOTON\_TOP

### Descripción del componente

### Estructura interna del componente (diagrama RTL)

El código de esta entidad es del estilo estructural, ya que se interconectan los distintos componentes, entradas y salidas mediante señales. El código es el siguiente:

De esta forma, el diagrama de bloques RTL de la entidad quedará de la siguiente forma:

### Componentes internos

A continuación se describen todos los componentes internos de la BOTON\_TOP.

#### DEBOUNCER

#### EDGEDTCTR

#### SYNCHRNZR

## FSM\_TOP

### Descripción del componente

El componente FSM\_TOP es el encargado de juntar todos los elementos necesarios

### Estructura interna del componente (diagrama RTL)

El código de esta entidad es del estilo estructural, ya que se interconectan los distintos componentes, entradas y salidas mediante señales. El código es el siguiente:

De esta forma, el diagrama de bloques RTL de la entidad quedará de la siguiente forma:

Diagrama, Esquemático

Descripción generada automáticamente

### Componentes internos

A continuación se describen todos los componentes internos de la FSM\_TOP.

#### FSM\_MASTER

Esta entidad consiste en una máquina de estados maestra encargada de coordinar las distintas fases del juego y las máquinas de estado y componentes esclavos:

El diagrama de estados de la máquina FSM\_MASTER es el siguiente:

Diagrama

Descripción generada automáticamente

A continuación se describe el significado de los diferentes estados:

* **S\_STBY**: El juego aún no se ha iniciado. Para iniciarlo, se espera una pulsación del botón central (OK) de los 5 de la placa. Hasta que no se inicie, se muestra un mensaje de “START” por los displays.
* **S0**: Estado de generación de la secuencia a adivinar por el jugador. El componente LFSR está continuamente generando nuevas secuencias aleatorias a adivinar. Cuando se recibe una señal de DONE\_LFSR = ‘1’, se guarda la nueva secuencia a adivinar en una variable, y se continua con el desarrollo del juego.
* **S1**: Disparo del temporizador durante 2 segundos para mostrar un mensaje.
* **S1\_WT**: Estado de espera de muestra del mensaje de “GO”, indicándole al jugador que comience a introducir la secuencia que considere.
* **S2**: Disparo de la máquina de estados esclava de INCHECK.
* **S2\_WT**: Estado de espera de la máquina maestra mientras la máquina esclava INCHECK está en funcionamiento. Existen dos posibilidades: Que se finalice el juego ganando (se recibe un DONE\_INCHECK = “01”), o que se finalice el juego perdiendo (se recibe un DONE\_INCHECK = “10”).
* **S3**: Disparo del temporizador durante 2 segundos para mostrar un mensaje.
* **S3\_WT**: En caso de ganar, se muestra un mensaje de “WIN” por los dispalys.
* **S4**: Disparo del temporizador durante 2 segundos para mostrar un mensaje.
* **S4\_WT**: En caso de perder, se muestra un mensaje de “GAME OVER” por los dispalys.
* **S5**: Disparo de la máquina de estados esclava de SHOWSEQ.
* **S5\_WT**: Estado de espera hasta la finalización de SHOWSEQ

Para la programación de esta máquina de estados se ha declarado un tipo especial denominado **STATE\_MASTER\_T**. En el quedan incluidos todos los estados mostrados anteriormente.

Además, cabe destacar que se ha utilizado un modelo de una máquina de estados a partir de tres procesos diferentes:

* **state\_register**: Encargado de actualizar el estado de la máquina de estados a cada pulso de reloj. En caso de llegar una señal de RESET, se vuelve la S\_STBY.
* **nxt\_state\_decoder**: En este proceso quedan reflejadas todas las transiciones entre los diferentes estados, producidas por las entradas a la máquinas de estados.
* **output\_decoder**: Proceso dedicado a, según el estado actual de la máquina de estados, activar unas salidas u otras.

#### FSM\_INCHECK

Este componente consiste en una máquina de estados esclava encargada de la recepción y comprobación de los valores introducidos por el jugador. Cada vez que se realiza una pulsación, FSM\_INCHECK comprueba si el valor introducido coincide con el de la posición correspondiente de la secuencia generada aleatoriamente por LFSR.

Cabe destacar que los valores de la secuencia a adivinar solo pueden tomar

El diagrama de estados de la máquina FSM\_INCHECK:

Diagrama

Descripción generada automáticamente

#### FSM\_SHOWSEQ

#### Diagrama Descripción generada automáticamente

#### LFSR

#### FSM\_SLAVE\_TIMER

## VISUALIZER\_TOP

### Descripción del componente

### Estructura interna del componente (diagrama RTL)

El código de esta entidad es del estilo estructural, ya que se interconectan los distintos componentes, entradas y salidas mediante señales. El código es el siguiente:

De esta forma, el diagrama de bloques RTL de la entidad quedará de la siguiente forma:

### Componentes internos

A continuación se describen todos los componentes internos de la VISUALIZER\_TOP.

#### NATURAL\_DECODER

#### DISPLAY\_CONTROLLER

# Conclusiones e impresiones

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente con confianza media

Diagrama, Esquemático

Descripción generada automáticamente