

# **MANUAL DE PRÁCTICAS**

**Tecnología de Computadores**

**Grado en Ingeniería Informática**

**1<sup>er</sup> año, 2<sup>o</sup> cuatrimestre**

**uc3m** | **Universidad Carlos III** de Madrid

**DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA**

**Campus de Leganés**

Avenida del Universidad 30

28911 Leganés

**Date: 02/04/19**

## 1. Table of contents

---

1.	Table of contents .....	2
2.	Diseño de circuitos digitale .....	3
3.	Organización de las sesiones de prácticas .....	4
3.1	MATERIAL A ENTREGAR .....	4
3.2	evaluación de las prácticas .....	4
4.	Sesion : Juego de dados .....	6
4.1	Proceso de diseño y desarrollo del circuito .....	9
4.2	Material necesario.....	10
4.3	Montaje y pruebas .....	10
4.4	Informe final de práctica.....	11

## **2. Diseño de circuitos digitale**

---

Debido a la complejidad de los circuitos digitales actuales, el proceso de diseño de un circuito digital se caracteriza principalmente por:

- Utilización de herramientas CAD (simuladores, sintetizadores, análisis estático de tiempos, etc.)
- Utilización intensiva de técnicas, principalmente simulación, para comprobar el correcto funcionamiento del circuito.

Por ello, las prácticas de la asignatura se apoyarán en la utilización de herramientas de diseño. Para la realización de las prácticas los alumnos utilizarán la herramienta Quartus-II (Web Edition) de Altera para realizar la captura esquemática (dibujo del esquema de componentes en el ordenador) de los circuitos propuestos como solución, así como para comprobar su funcionamiento mediante simulación.

La herramienta Quartus-II (Web Edition) se puede descargar del portal <http://www.altera.com>. Se recomienda usar la versión 9.0 que puede encontrarse en la siguiente dirección: Support > Design Software > Download.

Esta herramienta es gratuita, sin embargo, para su utilización se debe de registrar (rellenando una serie de formularios) y seguir los pasos que se indican para su instalación.

Para obtener una información detallada sobre el funcionamiento del programa Quartus-II, puede consultar el manual básico del programa que se encuentra a disposición de los alumnos en Aula Global, o bien la propia ayuda del programa.

### **3. Organización de las sesiones de prácticas**

---

Cada sesión de prácticas tendrá una duración total de 1,5 horas.

Las sesiones de laboratorio no están destinadas a que los alumnos resuelvan el problema planteado, sino que deben asistir al laboratorio con la práctica resuelta en la mayor parte posible.

Los alumnos deberán traer al laboratorio (en un pendrive, disquete, etc.) los ficheros generados al realizar la captura esquemática con el programa Quartus-II, así como los ficheros necesarios para la simulación del circuito en el ordenador del laboratorio.

Durante las sesiones de laboratorio los profesores comprobarán que la solución propuesta por el alumno funciona correctamente. De no ser así, durante la sesión de prácticas el alumno podrá realizar sobre su diseño inicial las modificaciones que consideren oportunas, con el objeto de conseguir que el circuito funcione correctamente.

#### **3.1 MATERIAL A ENTREGAR**

Antes de la realización de la práctica cada grupo deberá realizar un estudio previo, que consistirá en la solución propuesta al problema de diseño planteado. El estudio previo se deberá subir a aula global antes de entrar al laboratorio. Se recomienda llevar una copia impresa con ella al laboratorio.

**NO se recogerá ningún estudio previo fuera de plazo.**

En los apartados en los que se describe cada una de las sesiones de prácticas se indica con más detalle el contenido del estudio previo que los alumnos deben entregar

#### **3.2 EVALUACIÓN DE LAS PRÁCTICAS**

El conjunto de las cuatro prácticas tiene una valoración del 15% sobre la nota final. La nota de prácticas se mantendrá durante el curso académico en las convocatorias ordinaria y extraordinaria.

En caso de que el alumno tenga que repetir la asignatura, la calificación de prácticas NO se guardará para los cursos siguientes.

Durante el transcurso de la sesión de prácticas, los profesores comprobarán que la solución propuesta por el alumno funciona correctamente, asignándole una calificación que dependerá del grado de funcionamiento del circuito y la sencillez de la solución propuesta.

## 4. Sesión 4: Juego de dados

El objetivo de esta práctica es que el alumno se familiarice con los circuitos de control digital, mediante el diseño de una máquina de estados que controla la ejecución de un juego. El juego emula la tirada de dados entre dos jugadores y la decisión de quién es el ganador.



Figura 1. Circuito del juego de dados: interfaz

En la Figura 1 se muestra la interfaz del circuito a realizar en la práctica 4 de la asignatura. Hay dos entradas de servicio, reloj de sincronismo x(Clk) y señal de inicialización asíncrona (Reset\_N, activa por nivel bajo) para todos los bloques secuenciales del circuito<sup>1</sup>. Así mismo, hay una señal de control, que deben pulsar los jugadores (Start) que permite la tirada del dado para cada jugador y el cálculo de la decisión de quién es el ganador.

El circuito tiene dos tipos de salidas. El primero consiste en el valor de la tirada de cada uno de los jugadores, que se conectará a los displays de 7 segmentos (presentes en el entrenador del banco de trabajo del laboratorio); y el segundo consiste en la decisión de quién es el ganador (Ganador1 ó Ganador2) o ha habido un empate. Estas salidas se conectarán a los LEDs del mencionado entrenador.

### Arquitectura del Juego de Dados

El circuito se compone de seis bloques que se muestran en la Figura 2.

1. Para el bloque “Dado Electrónico” usaremos el circuito LFSR realizado en la práctica anterior, alimentado a una frecuencia suficientemente alta para que sea prácticamente

<sup>1</sup> El reloj Clk se conectará al generador de reloj del entrenador del banco de trabajo del laboratorio, y la señal de inicialización asíncrona (Reset\_N) se conectará a uno de los interruptores de dicho entrenador (switches)

imposible predecir la tirada que se va a obtener. El valor de la *semilla* puede dejarse fija o conectarse a los interruptores del entrenador del banco de trabajo. Será necesario poner como salidas de este circuito los valores de los biestables (Q0, Q1, Q2 y Q3).

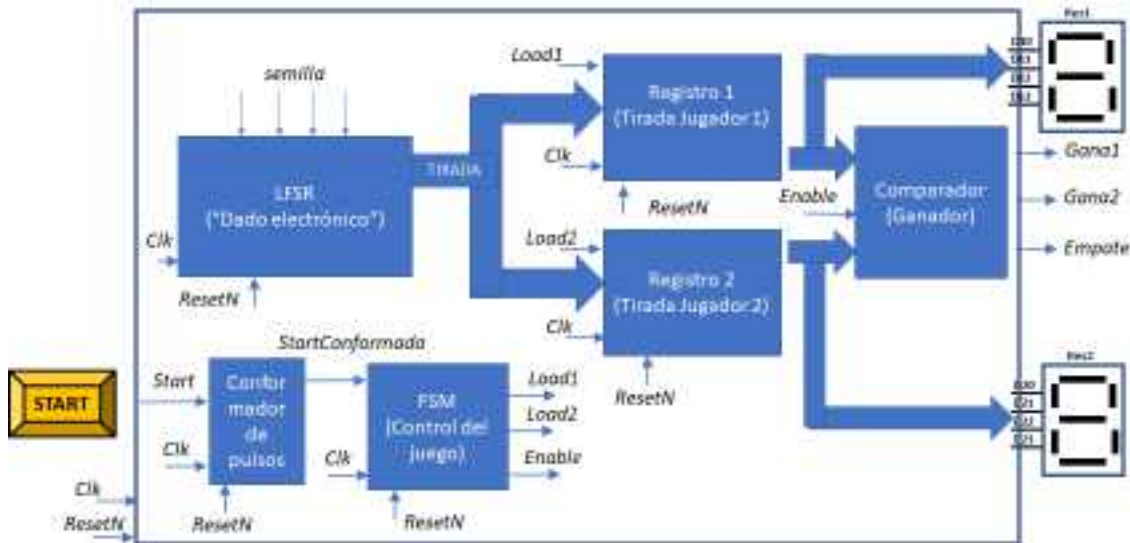


Figura 2. Diagrama de bloques del circuito “Juego de Dados”

2. Hay dos registros que almacenarán el valor de las tiradas de cada jugador. En el registro superior de la figura, se guardará la tirada del Jugador 1, y en el inferior la del Jugador 2. Estos registros son síncronos con la señal Clk, tienen una señal de inicialización asíncrona (Reset\_N) activa por nivel bajo y una señal de carga síncrona (LoadX). La señal que habilita la carga de cada una de las tiradas será respectivamente Load1 y Load2. Ambos registros almacenan 4 bits que corresponden con la salida del bloque LFSR.

3. El bloque Comparador-Ganador se encarga de decidir quién ha ganado, comparando el valor almacenado en el registro 1 y en el registro 2. Este bloque es el único del circuito que es puramente combinacional. Cuando la señal Enable lo permita (vale ‘1’) se activará alguna de sus tres salidas. Si Enable vale ‘0’ todas las salidas (Gana1, Gana2 y Empate) valen ‘0’. Si Enable vale ‘1’, se activará **Gana1** (se pone a ‘1’) cuando Registro1 > Registro2. Por el contrario, **Gana2** será ‘1’ cuando Registro1 < Registro2 y Enable valga ‘1’. Finalmente, **Empate** será ‘1’ cuando Registro1 = Registro2 y Enable valga ‘1’. En el resto de los casos, las salidas valdrán ‘0’.

4. Finalmente, el bloque de control es el denominado FSM (Finite State Machine), que es una máquina de estados que da el turno a los jugadores y activa la decisión de ganador.

La máquina de estados sólo tiene una señal de entrada de control. Esta señal procede del exterior y hay que acondicionarla (Conformador de pulsos).

4.1 Conformado de la entrada **Start**. Al estar conectada a un botón (interfaz humana) pero aplicarse a un circuito que opera a la frecuencia del reloj de sincronismo (Clk del orden de cientos de kHz), es necesario conformar esta entrada para que una pulsación de los jugadores se traduzca en un único pulso de la señal de reloj. En la **¡Error! No se**

**encuentra el origen de la referencia.** Figura 3 se muestra el resultado esperado de esta conformación de la señal procedente del botón que interactúa con los jugadores. Así mismo, en la Figura 4 se muestra el circuito que implementa esta funcionalidad.

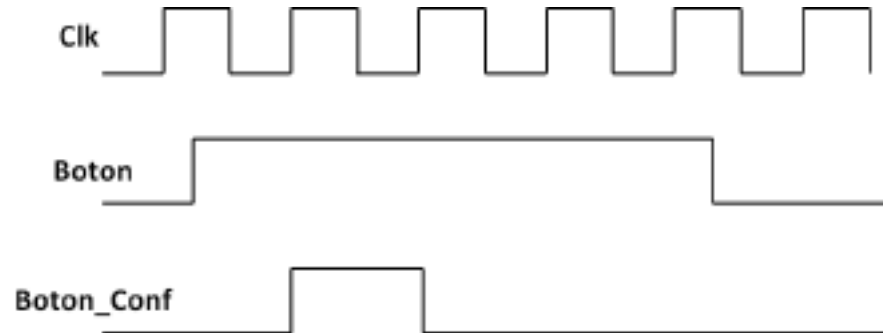


Figura 3. Conformación del botón de entrada

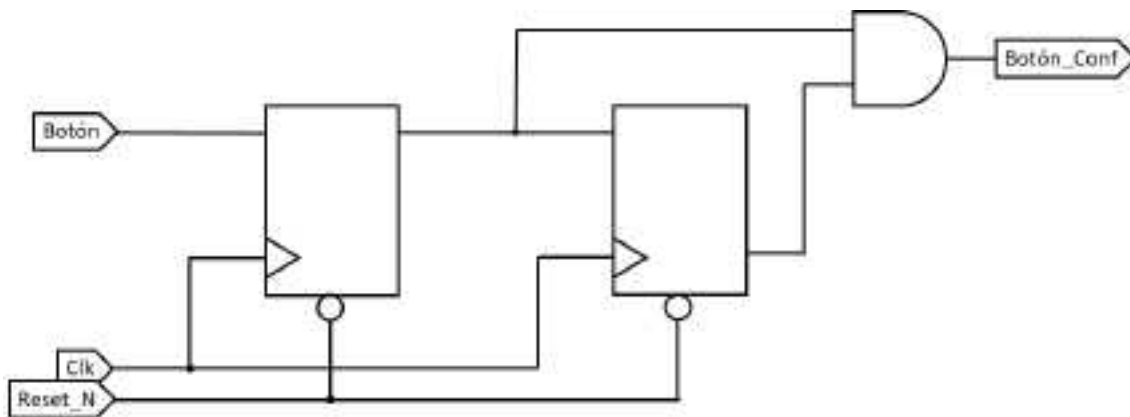


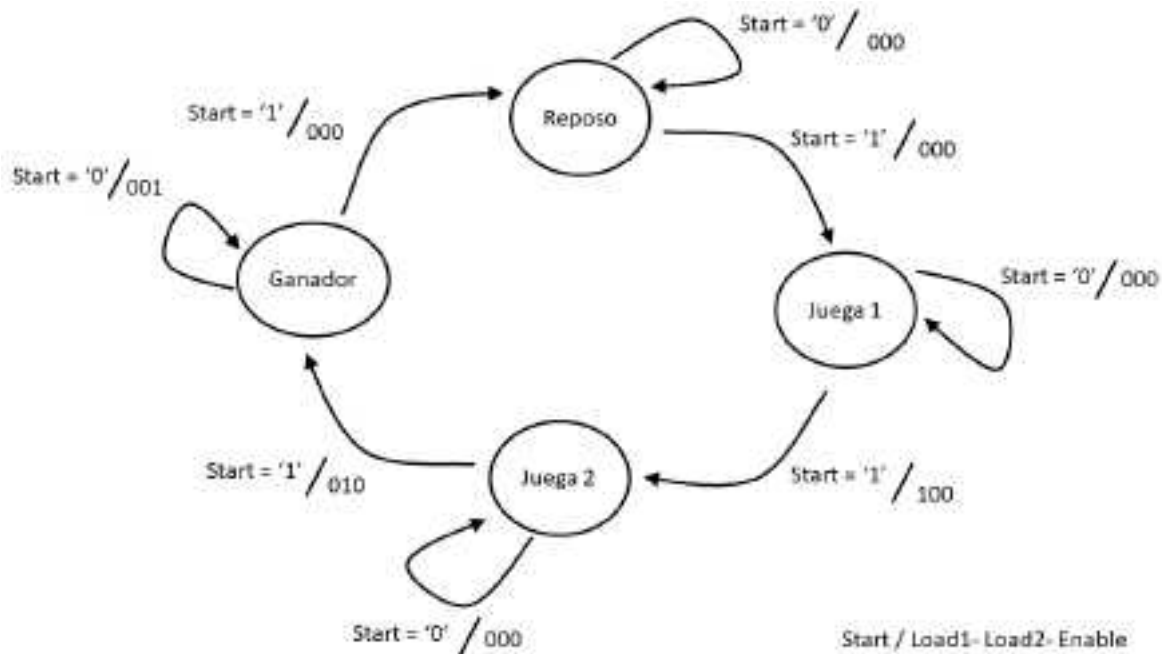
Figura 4. Circuito Conformador de Pulsos

#### 4.2 Máquina de estados de control del Juego de Dados.

El circuito de control que se propone consta de 4 estados y está diseñado como máquina de Mealy. En la Figura 5 se muestra el diagrama de estados de la propuesta.

La máquina de estados tiene una entrada de control (Señal de Start procedente del conformado del botón de entrada) y tres salidas (Load1, Load2 y Enable) que gobiernan los otros bloques del circuito.





**Figura 5. Diagrama de estados del circuito de control del Juego de Dados**

## **4.1 PROCESO DE DISEÑO Y DESARROLLO DEL CIRCUITO**

Previamente a la sesión de laboratorio, los alumnos realizarán un diseño jerárquico (En la herramienta QuartusII) con diferentes bloques funcionales en los que se incluyan todos los módulos anteriormente descritos. Así mismo, realizará un documento donde se incluirán todas las decisiones de diseño, tablas de verdad, mapas de Karnaugh, etc. que se han utilizado para el diseño.

1. Comparador de datos de 4 bits de entrada de datos a comparar, y señal de Enable (Activa por nivel alto) para activar las salidas. Este bloque produce 3 salidas (GanaA, GanaB y Empate) activas por nivel alto.
2. Registro de 4 bits de entrada (D) con señal de carga síncrona (Load), activa por nivel alto; reloj de sincronismo (Clk), activo por flanco de subida; y señal de inicialización asíncrona (Reset\_N) activa por nivel bajo.
3. Bloque LFSR con reloj de sincronismo (Clk), activo por flanco de subida; y señal de habilitación (Enable), activa por nivel alto; señal de carga (Load), activa por nivel alto; y semilla a elegir por los alumnos (k1, k2, k3 y k4).
4. Bloque conformador de botón (Start), con reloj de sincronismo (Clk), activo por flanco de subida; y señal de inicialización asíncrona (Reset\_N) activa por nivel bajo. Produce una salida StartConformada, activa por nivel alto, de un solo ciclo de reloj de duración.

5. Máquina de estados (FSM) con reloj de sincronismo (Clk), activo por flanco de subida; y señal de inicialización asíncrona (Reset\_N) activa por nivel bajo; entrada StartConformada, activa por nivel alto. Este bloque produce tres salidas, activas por nivel alto: Load1 (Para cargar la salida del bloque LFSR en el registro del jugador\_1) Load2 (Para cargar la salida del bloque LFSR en el registro del jugador\_2) y Enable (Para generar la salida del bloque comparador).

Los alumnos deberán llevar el diseño completo simulado a la sesión de laboratorio, en la que se corregirán los errores, si los hubiera, y se prototipará el circuito en la FPGA proporcionada durante la sesión. El dispositivo lógico que se va a programar pertenece a la familia MAX7000S, y su nombre es EPM7064SLC44-10.

## **4.2 MATERIAL NECESARIO**

Esta práctica incluye el montaje de un circuito y será necesario que los alumnos dispongan del siguiente material:

1. Cable rígido unipolar de sección de 0,5 mm<sup>2</sup>.
2. Alicates de corte y pelacables (opcional).

El resto del material necesario se proporcionará en el laboratorio.

## **4.3 MONTAJE Y PRUEBAS**

1. Compilar y sintetizar el circuito digital, con objeto de traducir el esquema del circuito a puertas lógicas. El dispositivo lógico que se va a programar pertenece a la familia MAX7000S, y su nombre es EPM7064SLC44-10.
2. Simular el circuito para comprobar su correcto funcionamiento. Generación del fichero de programación. Dicho fichero, cuya extensión es \*.pof, se genera de forma automática durante el proceso de compilación. Cada puesto de laboratorio programará su FPGA mediante el puerto USB.
3. Identificación de pines de entrada/salida del circuito. Se pretende conocer qué pin de la CPLD se ha asignado a cada señal de entrada/salida del circuito. Para ello se debe consultar en el fichero \*.rpt, también generado de forma automática durante el proceso de compilación.
4. Alimentación de la placa, y su conexión al entrenador. Una vez realizada la síntesis y programación, ya se dispone del circuito lógico dentro de la CPLD. Para probar su funcionamiento, es necesario conectar la placa al entrenador de la siguiente manera:
  - a. Alimentación de la placa a 5V, proporcionados por el entrenador. Cuidado al conectar los terminales de 5V (Vcc) y 0V (GND) y no ponerlos al revés.
  - b. NO CONECTAR A 5V TODOS LOS PINES DE LA CPLD QUE SE LLAMEN Vcc. Todos esos pines están conectados internamente con la alimentación, de manera que al alimentar la placa a 5V, ya están todos conectados a Vcc.

- c. La señal START y las claves ( $k_1$ - $k_4$ ) se conectarán a los interruptores del entrenador.
  - d. La señal CLK es una señal cuadrada de amplitud máxima cinco voltios. Dicha señal se obtendrá del oscilador del entrenador.
  - e. Las Salidas ( $D_{11}$ - $D_{14}$ ) (Jugador1) y ( $D_{21}$ - $D_{24}$ ) (Jugador2) irán conectadas a los displays de 7 segmentos del entrenador.
  - f. Las salidas (GanaA, GanaB y Empate) irán conectadas a los LEDs del entrenador.
  - g. Llamar al profesor de prácticas para la revisión del montaje antes de encender la fuente de alimentación.
5. Comprobación del funcionamiento del Juego de Dados.

#### **4.4 INFORME FINAL DE PRÁCTICA**

Los alumnos deberán entregar un informe final donde se detallen el diseño finalmente implementado, indicando:

- 1. Tamaño del diseño en la FPGA (% de ocupación en bloques secuenciales y combinacionales)
- 2. Listado de entradas y salidas y número de pin asignado a cada una de ellas.
- 3. Mejoras implementadas

Como sugerencias de mejora, el alumno puede plantearse el diseño de la FSM como Máquina de Moore o la optimización de la FSM propuesta en este manual. No obstante, cualquier iniciativa será bienvenida y apoyada por los profesores de la asignatura.