



## **PRÁCTICA 7**

### **DISEÑO Y REALIZACIÓN DE SISTEMAS DIGITALES BASADOS EN FPGA II**

#### **1. INTRODUCCIÓN**

En esta práctica los alumnos deben realizar un sistema digital completo incluyendo el diseño, la simulación y la implementación en la placa de desarrollo DE0.

El sistema a realizar está basado en la metodología de la Práctica 6, la cual se debe aplicar para resolver el sistema secuencial correspondiente a cada grupo en esta práctica.

#### **2. OBJETIVOS DE APRENDIZAJE**

- Resolver problemas de electrónica digital.
- Implementar circuitos secuenciales síncronos.
- Realizar esquemas electrónicos y descripciones funcionales en VHDL.
- Utilizar recursos periféricos conectados a una FPGA. Realización del mapa de entradas y salidas. Asignación de pines de la FPGA.
- Realización de circuitos combinacionales.
- Simulación de componentes.
- Realización de sistemas.

#### **3. TAREAS PREVIAS AL LABORATORIO**

En esta práctica el alumno debe practicar los conocimientos aprendidos en las prácticas anteriores. Por ello se plantea como un trabajo independiente, es decir, no guiado por el profesor.

Cada grupo debe realizar el diseño y simulación de su sistema antes de asistir al laboratorio y debe realizar las consultas, sobre las dificultades que encuentre, en las tutorías de la asignatura.

#### 4. DESCRIPCIÓN DEL SISTEMA A REALIZAR

Diseñar y simular un circuito digital secuencial que implemente una máquina de estados de tipo Moore. Las salidas serán el propio estado interno del sistema **ESTADO[2..0]** y la señal **salida** de un bit. La entrada es la señal **entrada**.

Además, debe tener una entrada de inicialización asíncrona **RESET** y una de sincronización **reloj**. La evolución del estado interno se corresponde con la siguiente tabla de estados:

estado actual		estado siguiente																		
ESTADO	entrada																			salida
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
0	0	3	1	7	1	0	4	5	1	1	1	1	1	3	2	1	1	2	0	
	1	1	2	1	1	1	1	1	1	1	3	0	2	1	1	1	1	1		
1	0	2	2	1	2	2	1	7	2	2	3	2	2	0	4	2	2	2	0	
	1	2	2	2	4	5	2	2	2	2	2	7	2	2	3	3	3	3		
2	0	3	3	3	3	3	6	3	3	3	3	3	0	1	6	3	3	4	1	
	1	3	4	3	5	3	3	3	3	5	3	2	3	3	5	3	6	5		
3	0	2	5	4	6	4	3	0	4	4	5	4	7	4	7	4	4	4	1	
	1	4	5	4	6	4	4	0	4	4	4	1	4	0	3	7	2	5		
4	0	3	5	4	3	0	5	3	5	2	5	5	3	5	4	5	5	6	0	
	1	5	3	5	5	2	5	3	5	2	5	4	5	5	7	7	0	5		
5	0	7	6	6	6	6	4	4	7	6	7	6	4	6	7	6	6	6	0	
	1	6	1	7	6	6	6	6	6	1	6	3	6	6	5	7	4	7		
6	0	6	7	5	7	7	6	0	0	7	7	7	5	7	6	7	7	0	0	
	1	7	7	5	7	7	7	7	1	7	7	6	7	7	7	7	7	7		
7	0	0	0	3	7	0	7	6	1	5	1	0	6	3	0	0	0	0	1	
	1	0	0	0	0	5	0	1	0	5	1	5	3	3	0	0	5	1		

El estado se debe codificar en binario con un vector de 3 bits. Si se activa la señal **RESET** el estado debe ser el 0.

El bloque de memoria se debe implementar con un registro de entrada y salida en paralelo de tres bits. Por tanto, las señales de salida del combinacional que realimenta el estado son las tres entradas en paralelo del registro y la entrada de habilitación (cargar un nuevo dato '1', o mantener el que tiene memorizado '0').

Cada alumno resolverá el ejercicio con la columna del estado siguiente que le corresponda según el número que grupo (A\*-1 al A\*-12).

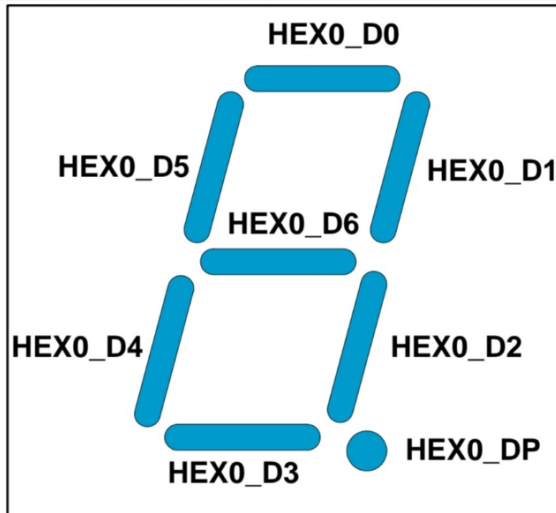
##### Tareas:

1. A partir de la tabla de estados realizar el correspondiente grafo de estados. Copiar y pegar el grafo en el ejercicio Moodle.
2. Completar la tabla de estados con las señales de entrada al elemento de memoria y obtener la tabla de verdad del circuito combinacional. Copiar y pegar la tabla en el ejercicio Moodle.
3. Describir en VHDL los módulos básicos necesarios y el de nivel superior en el que se instancian los básicos. Introducirlos en el orden que se pide en el ejercicio Moodle.
4. Programar un banco de pruebas que pruebe todas las transiciones posibles entre los estados. General el cronograma con ModelSim e incorporarlo al ejercicio Moodle..
5. Asignar los terminales de entrada a los interruptores (entrada y reset) y a un pulsador (reloj); y los tres bits de salida del **ESTADO** a los tres diodos LED de la derecha de la placa y la **salida** al de la izquierda. Recordar que para ver el ESTADO, que es un 'signal' interna de la arquitectura, hay que añadir salidas en la entidad superior de la máquina de estados que lo conecten con los visualizadores

6. Añadir un bloque básico decodificador binario-7 segmentos que pase de los tres bits del estado a los 7 bits de salida para representar el estado en un display. Realizar las asignaciones de terminales de la FPGA necesarias.
7. Realizar los apartados 4 al 6 describiendo la máquina de estado en VHDL comportamiento.

El alumno (en grupos de 2) debe realizar, como mínimo, las tareas 1, 2, 3 y 4 con anterioridad a la asistencia al laboratorio y traerlas en el formato necesario para pegarlas al ejercicio Moodle.

Debe mostrar las tareas previas a su profesor de prácticas y demostrar el correcto funcionamiento del sistema completo en el laboratorio.



Combinación	7 segmentos → Activo a nivel bajo
000	1000000
001	1111001
010	0100100
011	0110000
100	0011001
101	0010010
110	0000010
111	1111000

## 5. CONEXIONES ENTRE LOS PERIFÉRICOS Y LOS TERMINALES DE LA FPGA.

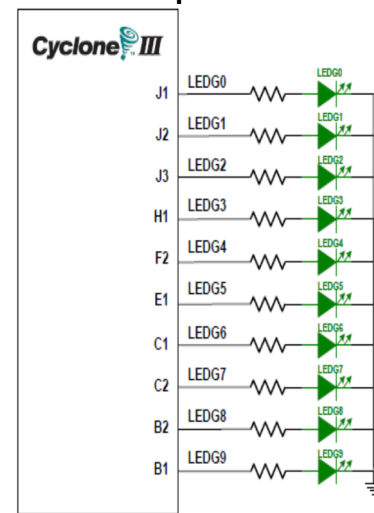
### Interruptores

Signal Name	FPGA Pin No.
SW[0]	PIN_J6
SW[1]	PIN_H5
SW[2]	PIN_H6
SW[3]	PIN_G4
SW[4]	PIN_G5
SW[5]	PIN_J7
SW[6]	PIN_H7
SW[7]	PIN_E3
SW[8]	PIN_E4
SW[9]	PIN_D2

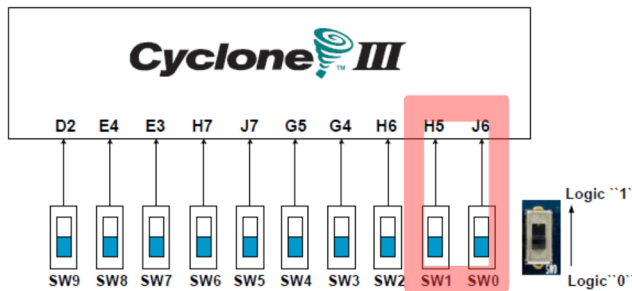
### Diodos LED

Signal Name	FPGA Pin No.
LEDG[0]	PIN_J1
LEDG[1]	PIN_J2
LEDG[2]	PIN_J3
LEDG[3]	PIN_H1
LEDG[4]	PIN_F2
LEDG[5]	PIN_E1
LEDG[6]	PIN_C1
LEDG[7]	PIN_C2
LEDG[8]	PIN_B2
LEDG[9]	PIN_B1

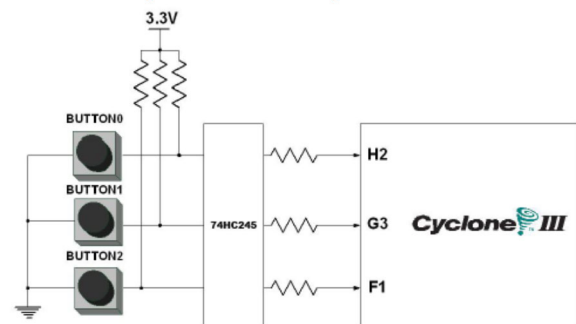
### Esquema LEDs



### Esquema de los interruptores



### Esquema de los pulsadores



### Display

Signal Name	FPGA Pin No.	Description
HEX0_D[0]	PIN_E11	Seven Segment Digit 0[0]
HEX0_D[1]	PIN_F11	Seven Segment Digit 0[1]
HEX0_D[2]	PIN_H12	Seven Segment Digit 0[2]
HEX0_D[3]	PIN_H13	Seven Segment Digit 0[3]
HEX0_D[4]	PIN_G12	Seven Segment Digit 0[4]
HEX0_D[5]	PIN_F12	Seven Segment Digit 0[5]
HEX0_D[6]	PIN_F13	Seven Segment Digit 0[6]
HEX0_DP	PIN_D13	Seven Segment Decimal Point 0

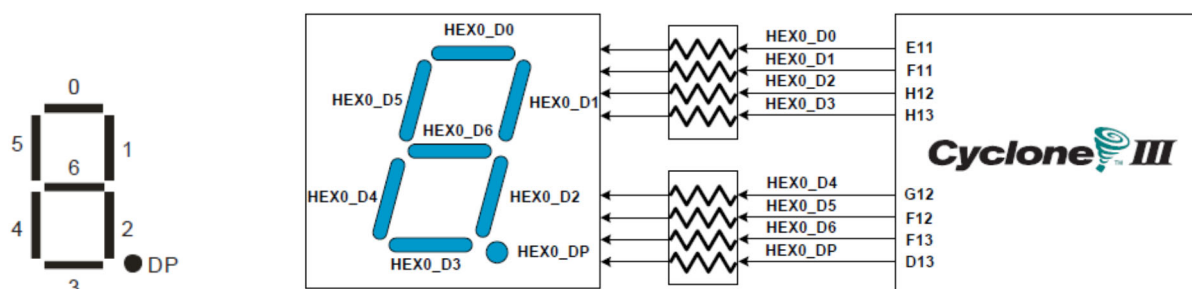
### Pulsadores

Signal Name	FPGA Pin No.
BUTTON [0]	PIN_ H2
BUTTON [1]	PIN_ G3
BUTTON [2]	PIN_ F1

### Relojos

Signal Name	FPGA Pin No.
CLOCK_50	PIN_G21
CLOCK_50_2	PIN_B12

### HEX0

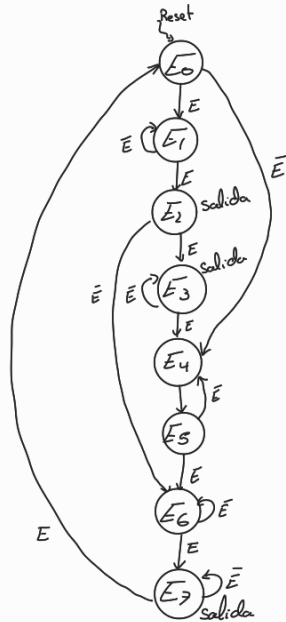


Para más información consultar el manual de la placa DE0 (archivo [DE0\\_User\\_manual.pdf](#)).

PÁGINA DEL PROFESOR (para entregar al terminar la práctica)

# Diagrama de Estados

estado actual		Grupo 6	
ESTADO	entrada	estado sigui.	salida
0	0	4	0
	1	1	
1	0	1	0
	1	2	
2	0	6	1
	1	3	
3	0	3	1
	1	4	
4	0	5	0
	1	5	
5	0	4	0
	1	6	
6	0	6	0
	1	7	
7	0	7	1
	1	0	



Estado actual	Estado sig.		salida
	$Q_2^*, Q_1^*, Q_0^*$		
$Q_2 \ Q_1 \ Q_0$	Entrada		
	0	1	
(0) 0 0 0	100	001	0
(1) 0 0 1	001	010	0
(2) 0 1 0	110	011	1
(3) 0 1 1	011	100	1
(4) 1 0 0	101	101	0
(5) 1 0 1	100	110	0
(6) 1 1 0	110	111	0
(7) 1 1 1	111	000	1

	Estado actual					Estado sig			salida
	$Q_2$	$Q_1$	$Q_0$	Entrada		$Q_2^*$	$Q_1^*$	$Q_0^*$	
E0	0	0	0	0		1	0	0	0
	0	0	0	1		0	0	1	0
E1	0	0	1	0		0	0	1	0
	0	0	1	1		0	1	0	0
E2	0	1	0	0		1	1	0	1
	0	1	0	1		0	1	1	1
E3	0	1	1	0		0	1	1	1
	0	1	1	1		1	0	0	1
E4	1	0	0	0		1	0	1	0
	1	0	0	1		1	0	1	0
E5	1	0	1	0		1	0	0	0
	1	0	1	1		1	1	0	0
E6	1	1	0	0		1	1	0	0
	1	1	0	1		1	1	1	0
E7	1	1	1	0		1	1	1	1
	1	1	1	1		0	0	0	1

Señal	DEB	PIN
Reset	SW0	PIN-J6
Entrada	SW1	PIN-H5
salida	LEDG9	PIN-B1
Reloj	Button 2	PIN-F1
ESTADO [0]	LEDG0	PIN-J1
ESTADO [1]	LEDG1	PIN-J2
ESTADO [2]	LEDG2	PIN-J3