

Práctica 6. Circuito Secuencial Contador DNI.

Documentación.

La **documentación** debe ser impresa y llevada el día que se asista al laboratorio. Esta documentación debe conservarse como material de la asignatura para el alumno y en ella se deben anotar las soluciones obtenidas (además de anotarlas en **las hojas de entrega previas**) para que el alumno tenga una copia de los resultados obtenidos.

La **primera parte de la práctica** consiste en el diseño y simulación de los circuitos propuestos mediante la herramienta *Xilinx Foundation 3.1i*. **Los diseños se recogerán en las hojas de resultados previos**, que serán entregadas en el momento de acceder al laboratorio, conservando el alumno el enunciado de la práctica como material docente. Asimismo, antes de la fecha indicada al efecto, deben subirse al Campus Virtual los ficheros del proyecto en un único archivo .zip.

La **segunda parte de la práctica**, que consiste en el **montaje experimental** del circuito, **se revisará en el laboratorio** el último día de prácticas junto al montaje de la práctica 5. En esta ocasión no será necesaria la entrega de hojas de resultados posteriores puesto que serán evaluados en el propio laboratorio.

Página intencionalmente en blanco

Parte 1. Creación de los esquemáticos.

Material Necesario

- Ordenador Personal
- Xilinx Foundation 3.1i

Objetivos

- Conocer el manejo del entorno de diseño digital Xilinx Foundation 3.1i para realizar el diseño y la simulación de circuitos secuenciales basados en elementos de memoria (flip-flops).
- Introducir técnicas de diseño modular para la implementación de metodologías de diseño "top-down" y "bottom-up".
- Poner en práctica los conocimientos teóricos para el diseño e implementación de un sistema secuencial síncrono.

1.1. Diseño y simulación de un sistema secuencial síncrono con 2 secuencias diferentes del DNI.

El **objetivo** de esta práctica es diseñar un circuito secuencial síncrono Moore que posea 2 comportamientos distintos dependiendo del valor de una señal de entrada X:

- Si $X = 0 \rightarrow$ Generará una secuencia binaria de 4 bits con las 8 cifras del DNI del alumno de manera consecutiva tal y como aparecen en su documento, y cuando termine comenzará otra vez por el principio.
- Si $X = 1 \rightarrow$ Generará una secuencia binaria de 4 bits con las 8 cifras del DNI del alumno de manera consecutiva, pero en este caso ordenadas de menor a mayor, y cuando termine comenzará otra vez por el principio.

Si el DNI del alumno no posee 8 cifras se rellenará con ceros por la izquierda en ambas secuencias. Si en mitad de la cuenta en una de las dos secuencias se cambia el valor de la entrada X, la nueva secuencia continuará a partir del último valor generado.

Ejemplos:

DNI = 99.321.125

Secuencia (en binario) con $X = 0 \rightarrow 9, 9, 3, 2, 1, 1, 2, 5$ y vuelta a empezar.

Secuencia (en binario) con $X = 1 \rightarrow 1, 1, 2, 2, 3, 5, 9, 9$ y vuelta a empezar.

DNI = 9.745.428 (DNI de 7 cifras, rellenamos por la izquierda con un cero)

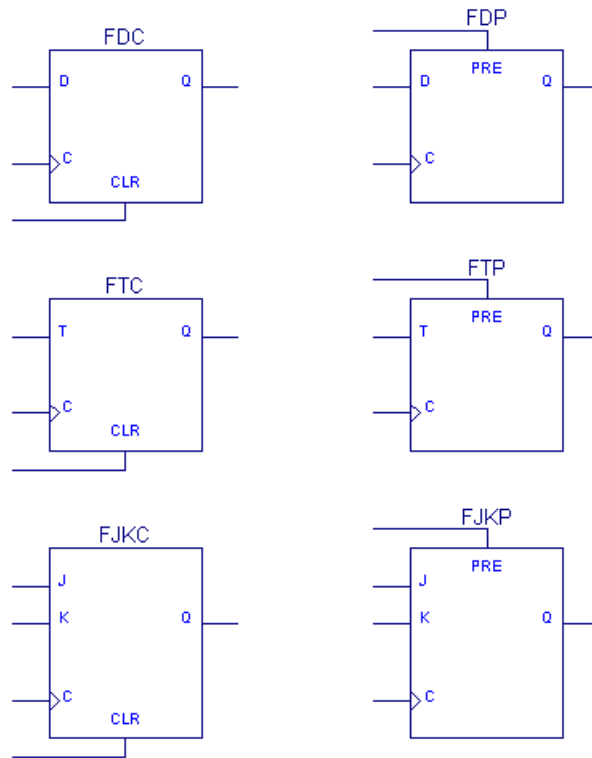
Secuencia (en binario) con $X = 0 \rightarrow 0, 9, 7, 4, 5, 4, 2, 8$ y vuelta a empezar.

Secuencia (en binario) con $X = 1 \rightarrow 0, 2, 4, 4, 5, 7, 8, 9$ y vuelta a empezar.

Restricciones del diseño:

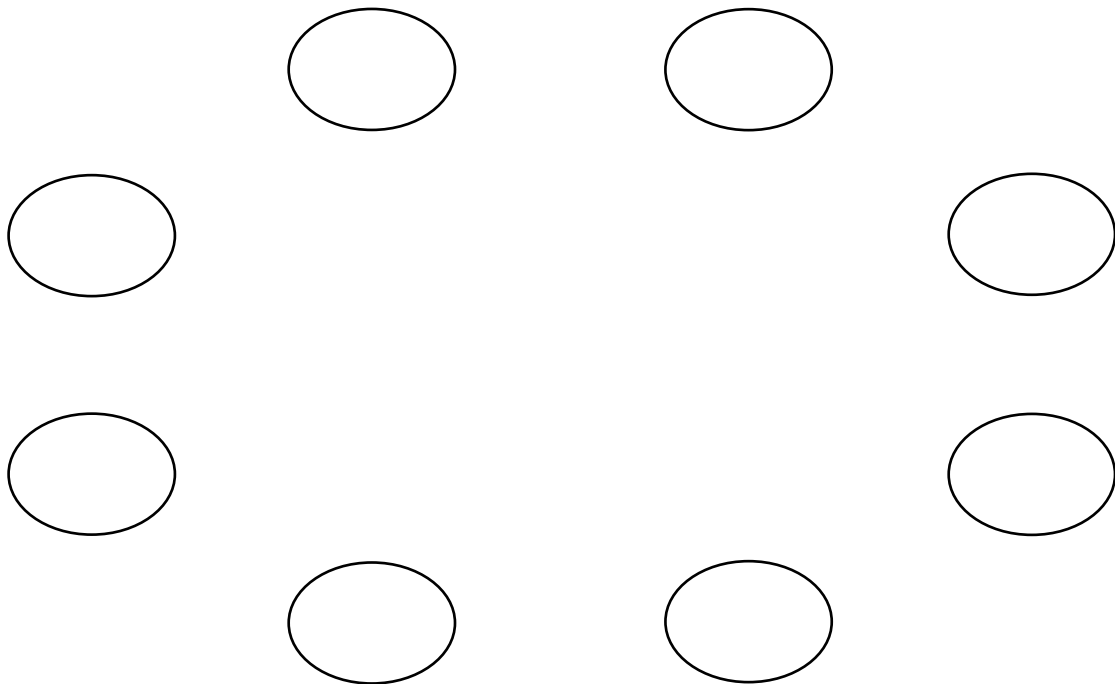
1.- Para el diseño del circuito secuencial síncrono se pueden usar flip-flops de cualquier tipo (D, T ó JK), con señal de "CLEAR" (fuerza el valor de salida a 0) o "PRESET" (fuerza el valor de salida a 1) según convenga (ver próxima figura).

2.- También se debe incluir una señal de entrada RESET para, una vez conectada a las entradas "PRESET" o "CLEAR" de los biestables, llevar al sistema a su estado inicial. Dicho estado inicial corresponderá a la primera cifra del DNI en el orden como aparece en el documento del alumno, es decir, la primera cifra de la secuencia cuando $X = 0$.



3.- La simulación del bloque debe realizarse utilizando los estímulos adecuados, para posteriormente almacenarlo como macro en la librería del proyecto. Dicha macro se usará en el proyecto final, que se unirá a la ya hecha en la práctica 5 para convertir la secuencia de binario a 7 segmentos, y así poder mostrarla por el display.

1.1.1. Diagrama de Estados:



1.1.2. Tabla de Estados-Salida:

q → Estado Presente

Q → Estado Futuro

X → Señal de entrada que decide el tipo de secuencia a generar:

- X = 0 → Secuencia de salida ordenada como en el DNI
- X = 1 → Secuencia de salida ordenada de menor a mayor

q (Estado Presente)	Q (Estado Futuro) Con X = 0	Q (Estado Futuro) Con X = 1	Secuencia de Salida (en decimal)

1.1.3. Tabla de Transición:

$q_2 q_1 q_0 / Q_2 Q_1 Q_0$ → 3 bits que codifican en binario el estado Presente / Futuro

$Y_3 Y_2 Y_1 Y_0$ → 4 bits que codifican en binario el número de secuencia generado en un instante concreto

q (Estado Presente) $q_2 \quad q_1 \quad q_0$	Q (Estado Futuro) Con X = 0 $Q_2 \quad Q_1 \quad Q_0$	Q (Estado Futuro) Con X = 1 $Q_2 \quad Q_1 \quad Q_0$	Salida (en binario) $Y_3 \quad Y_2 \quad Y_1 \quad Y_0$

1.1.4. Tabla de Excitación:

$B_2 B_1 B_0 \rightarrow$ 3 bits de entrada a los biestables que codifican el estado (usando flip-flops tipo D ó T). En caso de usar JK, serán 2 bits a la entrada de cada biestable, 6 bits en total

q (Estado Presente)			Entrada a los Biestables, con X = 0			Entrada a los Biestables, con X = 1			Salida (en binario)			
q_2	q_1	q_0	B_2	B_1	B_0	B_2	B_1	B_0	Y_3	Y_2	Y_1	Y_0

1.1.5. Funciones de Excitación:

Se usarán 3 mapas de Karnaugh en el caso de biestables D ó T, y 6 mapas en el caso de utilizar JK:

X, q_2	q_1, q_0	00	01	11	10
00					
01					
11					
10					

Función Lógica Mínima:

.....

.....

.....

.....

.....

X, q_2	q_1, q_0	00	01	11	10
00					
01					
11					
10					

Función Lógica Mínima:

.....

.....

.....

.....

.....

X, q_2 q_1, q_0				
	00	01	11	10
00				
01				
11				
10				

Función Lógica Mínima:

.....

.....

.....

.....

.....

X, q_2 q_1, q_0				
	00	01	11	10
00				
01				
11				
10				

Función Lógica Mínima:

.....

.....

.....

.....

.....

X, q_2 q_1, q_0				
	00	01	11	10
00				
01				
11				
10				

Función Lógica Mínima:

.....

.....

.....

.....

.....

X, q_2 q_1, q_0				
	00	01	11	10
00				
01				
11				
10				

Función Lógica Mínima:

.....

.....

.....

.....

.....

1.1.6. Funciones de Salida:

q_2, q_1		00	01	11	10
q_0	0				
	1				

Función Lógica Mínima:

.....

.....

.....

q_2, q_1		00	01	11	10
q_0	0				
	1				

Función Lógica Mínima:

.....

.....

.....

q_2, q_1		00	01	11	10
q_0	0				
	1				

Función Lógica Mínima:

.....

.....

.....

q_2, q_1		00	01	11	10
q_0	0				
	1				

Función Lógica Mínima:

.....

.....

.....

1.1.7. Diagrama del Circuito:

Página intencionalmente en blanco

Parte 2. Montaje Experimental en Laboratorio.

Material Necesario

- Fuente de Alimentación
- Generador de funciones
- Osciloscopio
- Material de la caja del puesto: Cables, sondas, etc.
- Placa de CPLD y de Periféricos
- PC
- Cable Paralelo ↔ JTAG

Objetivos

- Testear en las placas de la CPLD y de Periféricos el funcionamiento de las macros creadas en las prácticas 5 y 6 a través del uso del software Xilinx Foundation 3.1i

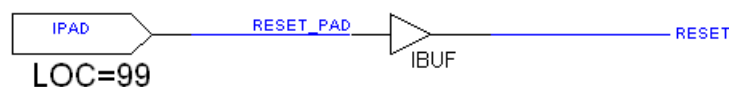
2.1. Implementación del sistema secuencial síncrono con 2 secuencias diferentes del DNI.

Utilizar las macros previamente diseñadas en las prácticas 5 y 6 para construir un circuito que, implementado en la placa de la CPLD, utilice la placa de periféricos para mostrar en su display 1 la secuencia del DNI escogida a través de la señal de entrada X.

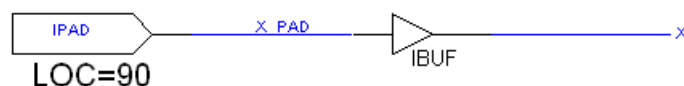
Para poder hacer uso de la placa de Periféricos desde la placa CPLD se han realizado unos módulos que deben ser copiados a la librería local del proyecto para posteriormente ser usados, al igual que se hizo en la práctica 5.

Restricciones del diseño:

- 1.- El pulsador de RESET necesario para inicializar el sistema secuencial síncrono está localizado en el pin 99:

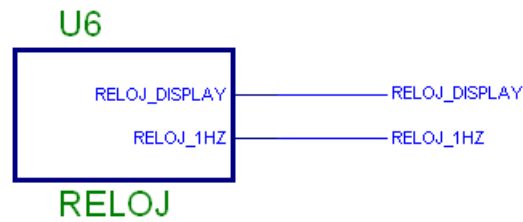


- 2.- La señal de entrada X del sistema secuencial se introducirá por el microinterruptor 1 de la placa de la CPLD (pin 90).

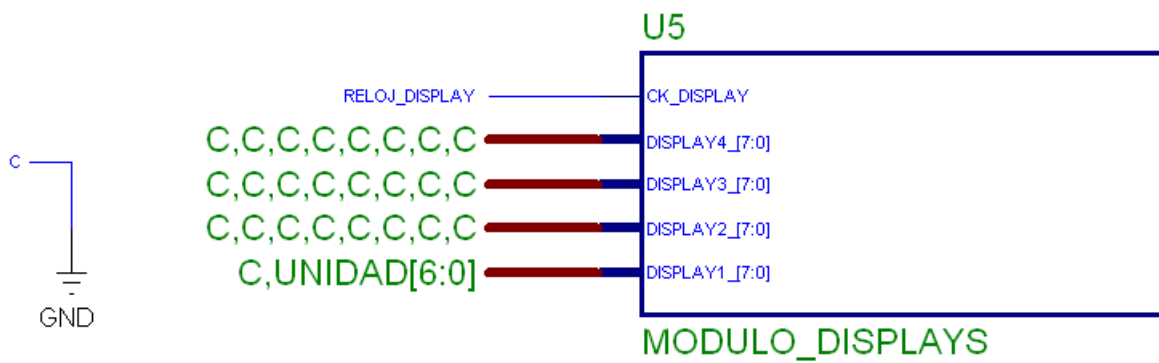


- 3.- Como reloj para los biestables del sistema secuencial síncrono será necesario usar la señal "RELOJ_1HZ" que sale de la macro "RELOJ" existente en el proyecto "MODULOS.ZIP". Será necesario por tanto, al igual que en la práctica 5, copiar todos los bloques de dicho proyecto al de la práctica 6 para disponer de ellos.
Nota: Usamos este reloj de 1 Hz de frecuencia para poder apreciar visualmente los cambios en la secuencia de salida, a razón de una cifra por segundo. Si utilizáramos un reloj mucho más rápido nuestros ojos no serían capaces de detectar estos cambios, ésa es la razón de usar un reloj tan lento para el secuencial.

Además, este mismo módulo se encargará de hacer parpadear unos leds de la placa para marcar el ritmo de los cambios del reloj de 1 Hz.



4.- La salida del sistema secuencial (4 bits) se conectará a la entrada del decodificador BCD – 7 segmentos diseñado en la práctica 5, y éste a su vez al "MODULO_DISPLAYS" como ya vimos en dicha práctica:



Alumno: Grupo: Puesto de trabajo:

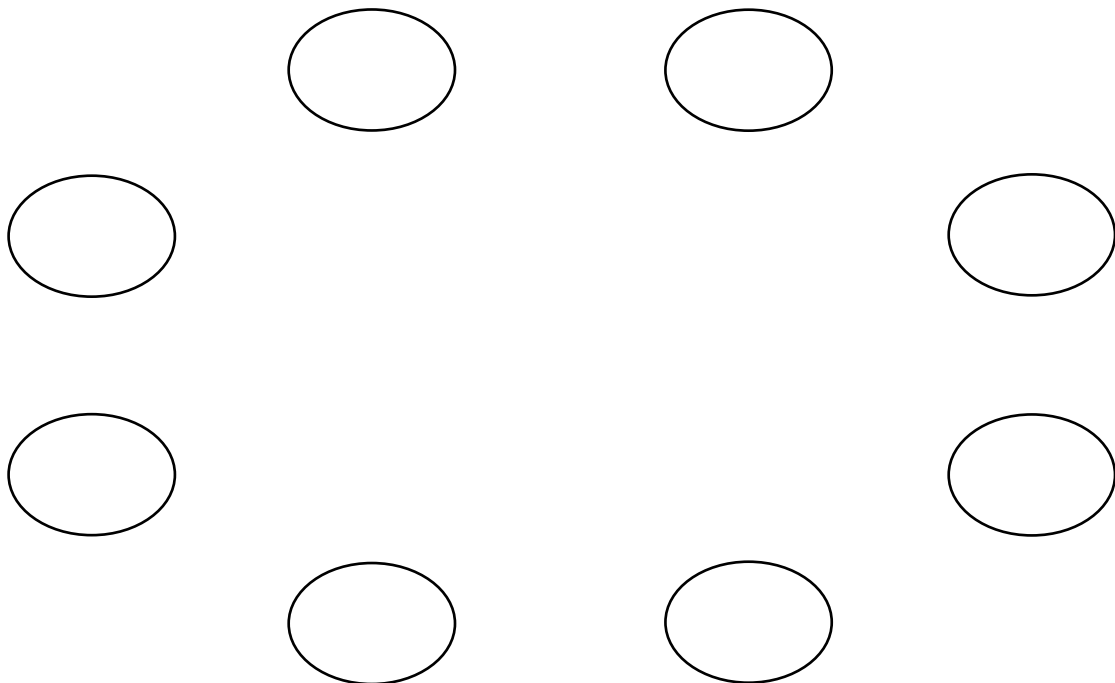
Hojas de entrega previa.

Las **hojas de entrega previa** deben ser impresas, rellenas con sus soluciones y entregadas a la entrada de la sesión de prácticas.

Parte 1. Creación de los esquemáticos.

1.1. Diseño y simulación de un sistema secuencial síncrono con 2 secuencias diferentes del DNI.

1.1.1. Diagrama de Estados:



Alumno: Grupo: Puesto de trabajo:

1.1.2. Tabla de Estados-Salida:

q → Estado Presente

Q → Estado Futuro

X → Señal de entrada que decide el tipo de secuencia a generar:

- X = 0 → Secuencia de salida ordenada como en el DNI
- X = 1 → Secuencia de salida ordenada de menor a mayor

q (Estado Presente)	Q (Estado Futuro) Con X = 0	Q (Estado Futuro) Con X = 1	Secuencia de Salida (en decimal)

1.1.3. Tabla de Transición: $q_2 q_1 q_0 / Q_2 Q_1 Q_0 \rightarrow$ 3 bits que codifican en binario el estado Presente / Futuro $Y_3 Y_2 Y_1 Y_0 \rightarrow$ 4 bits que codifican en binario el número de secuencia generado en un instante concreto

q (Estado Presente)	Q (Estado Futuro) Con X = 0	Q (Estado Futuro) Con X = 1	Salida (en binario)
$q_2 \quad q_1 \quad q_0$	$Q_2 \quad Q_1 \quad Q_0$	$Q_2 \quad Q_1 \quad Q_0$	$Y_3 \quad Y_2 \quad Y_1 \quad Y_0$

Alumno: Grupo: Puesto de trabajo:

1.1.4. Tabla de Excitación:

$B_2 B_1 B_0 \rightarrow$ 3 bits de entrada a los biestables que codifican el estado (usando flip-flops tipo D ó T). En caso de usar JK, serán 2 bits a la entrada de cada biestable, 6 bits en total

q (Estado Presente)			Entrada a los Biestables, con $X = 0$			Entrada a los Biestables, con $X = 1$			Salida (en binario)			
q_2	q_1	q_0	B_2	B_1	B_0	B_2	B_1	B_0	Y_3	Y_2	Y_1	Y_0

1.1.5. Funciones de Excitación:

Se usarán 3 mapas de Karnaugh en el caso de biestables D ó T, y 6 mapas en el caso de utilizar JK:

X, q_2	00	01	11	10
q_1, q_0				
00				
01				
11				
10				

Función Lógica Mínima:

.....

.....

.....

.....

.....

X, q_2	00	01	11	10
q_1, q_0				
00				
01				
11				
10				

Función Lógica Mínima:

.....

.....

.....

.....

.....

Alumno: Grupo: Puesto de trabajo:

X, q ₂	00	01	11	10
q ₁ , q ₀				
00				
01				
11				
10				

Función Lógica Mínima:

.....

.....

.....

.....

.....

X, q ₂	00	01	11	10
q ₁ , q ₀				
00				
01				
11				
10				

Función Lógica Mínima:

.....

.....

.....

.....

.....

X, q ₂	00	01	11	10
q ₁ , q ₀				
00				
01				
11				
10				

Función Lógica Mínima:

.....

.....

.....

.....

.....

X, q ₂	00	01	11	10
q ₁ , q ₀				
00				
01				
11				
10				

Función Lógica Mínima:

.....

.....

.....

.....

.....

1.1.6. Funciones de Salida:

q_2, q_1		00	01	11	10
q_0	0				
	1				

Función Lógica Mínima:

.....

.....

.....

q_2, q_1		00	01	11	10
q_0	0				
	1				

Función Lógica Mínima:

.....

.....

.....

q_2, q_1		00	01	11	10
q_0	0				
	1				

Función Lógica Mínima:

.....

.....

.....

q_2, q_1		00	01	11	10
q_0	0				
	1				

Función Lógica Mínima:

.....

.....

.....

Alumno: Grupo: Puesto de trabajo:

1.1.7. Diagrama del Circuito: