

# UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL CÓRDOBA



## CARRERA DE GRADO EN INGENIERÍA ELECTRÓNICA

### **“TÉCNICAS DIGITALES IV”**

Sistemas embebidos con SOC programable

**Materia Optativa de 6º nivel – 1º Cuatrimestre**

#### **Guía de Prácticos de Entrenamiento:**

Prof. Ing. Sergio D. Olmedo  
Prof. Ing. Pablo Cayuela

**Año 2020**

---

1. Guía didáctica de prácticos de entrenamiento.

1.1 Objetivo del aprendizaje

Realizar un repaso para la utilización de herramientas que nos permite utilizar lenguajes de descripciones de hardware para realizar la síntesis de diferentes descripciones de procesos para ser aplicados en FPGAs o CPLDs.

Desarrollar capacidades mínimas de diseño de circuitos y sistemas con técnicas de alto nivel (lenguajes de descripción de hardware).

Conocer el instrumental electrónico, y a la utilización de herramientas de software de simulación de circuitos y sistemas electrónicos.

1.2 Bibliografía

1.2.1 Principal

1. Ubicación: 621.392 PER d

Autor/es: Pérez, Serafín Alfonso; Soto, Enrique; Fernández, Santiago

Título: Diseño de Sistemas Digitales con VHDL

Publicación: Madrid, Thomson - Paraninfo

2. Ubicación: 629.89 GAR d

Autor/es: García Iglesias, José M.; Pérez Iglesias, Emilio J.

Título: Dispositivos Lógicos Programables (PLDs): diseño práctico de aplicaciones

Publicación: México, Madrid, Alfaomega - Ra-Ma

3. Ubicación: CDR/17 (CD) y 621.395 BRO f (libro)

Autor/es: Brown, Stephen; Vranesic, Zvonko

Título: Fundamentals of Digital Logicwith VHDL design (libro con CD)

Publicación: México, McGraw-Hill

4. Ubicación: CDR/19 (CD) y 621.382'2 MEY d (libro)

Autor/es: Meyer-Baese, Uwe

Título: Digital signal processing with field programmable gate arrays (DSP with FPGAs)

Publicación: Florida (USA), Springer

5. Ubicación: 621.395 UYE i

Autor/es: Uyemura, John P.

Título: Introducción al diseño de sistemas digitales: un enfoque integrado

Publicación: México, Thomson

6. Ubicación: 621.395 LLO s

Autor/es: Lloris Ruiz, Antonio; Prieto Espinosa, Alberto; Parrilla Roure, Luis

Título: Sistemas digitales

Publicación: Madrid, McGraw-Hill

7. Ubicación: 621.392 ASH d

Autor/es: Ashenden, Peter J.

Título: The designer's guide to VHDL

Publicación: USA, Elsevier

8. Ubicación: CDR/16 (CD) y 621.392 PAR v (libro)

Autor/es: Pardo Carpio Fernando; Boluda Grau, José

Título: VHDL - Lenguaje para síntesis y modelado de circuitos (libro con CD)

Publicación: México, Alfaomega - Ra-Ma

### 1.2.2 Auxiliar

Documentos subidos en la autogestión.

## 2. [Actividades](#)

- Estudiar la bibliografía recomendada.
- Realizar los ejercicios propuestos en la materia.

### 2.1) Tareas a resolver

## 3. [Guía a resolver](#)

### 3.1 Instrucciones

La tarea consiste en realizar los prácticos de la guía de entrenamiento.

### 3.2 Consejos

- Encabezar las descripciones con la mayor información posible.
- Documentar inconvenientes encontrados y realizar auto consejo
- Crear carpetas con nombre acorde al practico para una posterior instanciación del mismo

```
-----  
-- UNIVERSIDAD TECNOLÓGICA NACIONAL - FACULTAD REGIONAL CORDOBA  
-- Carrera: INGENIERIA ELECTRÓNICA.  
-- Asignatura: TÉCNICAS DIGITALES IV (ELECTIVA).  
-- Año: 20__.  
-- Grupo n°: XX.  
-- Integrantes:  
--  
--     CASIMIRO, Nahuel lito           Leg. n°:01520  
--     MANODEACHA, Duran Anastasio.   Leg. n°:02524  
--     POCAONDA, Clementina           Leg. n°:03177  
--  
-- Fecha de Entrega: xx/xx/20xx.  
-- hardware utilizado  
-- VHDL auxiliares utilizados  
-- UCF utilizado  
  
-- Práctico n°XX: FlipFlop Tipo D - Activo por Flanco Ascendente.  
-- Descripción: Descripción de cómo se realizó.  
-- Colocar dificultades  
-- Colocar algún consejo para recordar en el futuro
```

### 3.3 [La Forma de presentación](#)

La presentación será informal a fines de que el docente pueda evaluar si el estudiante ha logrado dominio suficiente para pasar a resolver problemas mayores en VHDL.

---

### 3.4 Guía de Trabajos prácticos de entrenamiento.

#### **Practico\_01) Sumador de un bit (usando "when / else")**

Objetivo: Conocer la construcción "when / else"  
Interpretar el concepto de "proceso concurrente".  
Implementar en un CPLD un sumador de un bit.

#### **Especificaciones de entradas / salidas**

El sumador tendrá dos entrada (bit) y una salida (bus de 2 bits)

```
a : in std_logic;  
b : in std_logic;  
Enable : in std_logic;  
Resultado : out std_logic_vector(1 downto 0)
```

#### **Pasos a seguir**

1. Diseñar primero la entidad y la arquitectura (pero sin llenar el cuerpo). En este punto se debería pasar la verificación de sintaxis sin problema.
  2. Dentro del cuerpo de la arquitectura, se debe describir el comportamiento del sumador solicitado.
  3. Probar la correcta descripción mediante la simulación.
- 

#### **Practico\_02) Sumador de un bit (usando " case / when ").**

Objetivo: Conocer la construcción " case / when ".  
Interpretar el concepto de "proceso".  
Implementar en un CPLD un Sumador de un bit

#### **Especificaciones de entradas / salidas**

El sumador tendrá dos entrada (bus de 2 bits) y una salida (bus de 2 bits)

```
A : in std_logic_vector(1 downto 0)  
Enable : in std_logic  
D : out std_logic_vector(1 downto 0)
```

#### **Pasos a seguir**

1. Diseñar primero la entidad y la arquitectura (pero sin llenar el cuerpo). En este punto de debería pasar la verificación de sintaxis sin problema.
2. Dentro del cuerpo de la arquitectura, se debe describir el comportamiento del sumador solicitado, pero utilizando la construcción " case / when ".
3. Probar la correcta descripción mediante la simulación.
4. Sacar conclusiones sobre ambas descripciones

#### **Practico \_03) Decodificador BCD a Decimal (símil cd4028)**

Objetivo: Reconocer un proceso secuencial.  
Conocer la construcción "if / else"

Interpretar el concepto de “proceso concurrente”.

Utilizar la herramienta de plantilla del software.

Destacar el sincronismo con el CLK.

Implementar en un CPLD un Decodificador BCD a Decimal con el comportamiento de circuito integrado ( MC14028B).

### Especificaciones de entradas / salidas

clock : in std\_logic

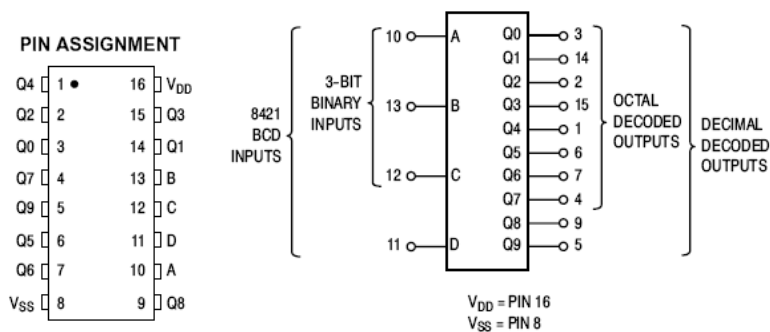
reset : in std\_logic

input : in std\_logic\_vector(3 downto 0)

salida : outstd\_logic\_vector(9 downto 0)

### Referencia

#### MC14028B



**TRUTH TABLE**

D	C	B	A	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0
0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0

### Pasos a seguir

1. Diseñar la entidad y la arquitectura.
2. Dentro del cuerpo de la arquitectura, se debe describir el comportamiento similar al del integrado MC14028B .
3. Utilizar una arquitectura secuencia. Utilizar dentro de una proceso la construcción “case / when ”.
4. Probar la correcta descripción mediante la simulación.
5. Utilizar el CPLD.
6. Leer el manual del kit de CPLD y generar el UCF correspondiente a su entidad.
7. Generar la síntesis de la descripción.
8. Realizar la simulación postfit.
9. Descargar en el CPLD

### Practico \_04) Decodificador de 4 to 2

Objetivo:

Reconocer un proceso secuencial.

Conocer la construcción "if / else"

Interpretar el concepto de “proceso concurrente”.  
 Utilizar Selección de dispositivo (CE).  
 Utilizar la herramienta de plantilla del software.  
 Utilizar las salidas del tipo “1”; ”0” y ”Z”.  
 Destacar la importancia de “Reset” por hardware.  
 Implementar en un CPLD un decodificador de 4 to 2.

### Especificaciones de entradas / salidas

pReset : in std\_logic.  
 pInput : in std\_logic\_vector(3 downto 0)  
 pOutput : out std\_logic\_vector(1 downto 0)  
 Clk : in std\_logic

### Referencia

Reset	pInput				pOutput	
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	1	0	0	0	1	1
1	x	X	x	x	0	0

### Pasos a seguir

1. Diseñar la entidad y la arquitectura.
2. Dentro del cuerpo de la arquitectura, se debe describir el comportamiento como se indica en la tabla.
3. Utilizar una arquitectura secuencial. Utilizar dentro de un proceso la construcción “when/ case”.
4. Probar la correcta descripción mediante la simulación.
5. Utilizar las salidas del tipo “1”; ”0” y ”Z”.

### Practico \_05) Multiplexor de 8 a 1

#### Objetivo:

Interpretar el concepto de “proceso concurrente”.  
 Utilizar Selección de dispositivo (CE)  
 Implementar en un CPLD al integrado MC74151 .solo como multiplexor y suponiendo solo señales digitales.  
 Utilizar salida “1” “0” y “Z” lógicos.

### Especificaciones de entradas / salidas

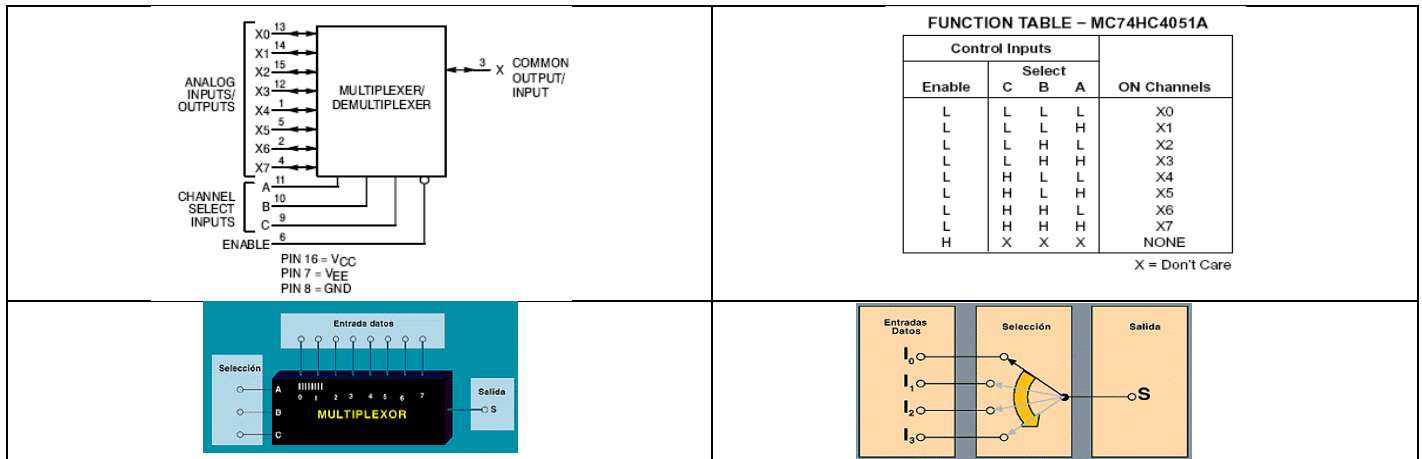
pE : in std\_logic;  
 input1,input2,input3,input4 input5,input6,input7,input8,input9,input10,input11: in std\_logic;  
 pSalida : out std\_logic);

#### Nota:

Generar señales utilizando la concatenación “&” para asignar las entradas a las señales  
 sSelector : in std\_logic\_vector(2 downto 0);  
 sEntrada : in std\_logic\_vector(7 downto 0);

**Referencia:**

MC74151

**Pasos a seguir**

1. Diseñar la entidad y la arquitectura.
2. Dentro del cuerpo de la arquitectura, se debe describir el comportamiento como se indica en la tabla (describiendo solamente como multiplexor de 8 a 1).
3. Utilizar una arquitectura concurrente. Utilizar dentro de un proceso la construcción "when/ case".
4. Probar la correcta descripción mediante la simulación.
5. Utilizar entrada de habilitación (enable).
6. utilizar la construcción when others => NULL;

**Practico \_06) Flipflop tipo JK(con reset síncrono // con reset asíncrono)****Objetivo:**

Interpretar la diferencia de comportamiento con una descripción con reset asíncrono respecto a una descripción con reset síncrono.  
Diseñar test bench para la simulación

**Especificaciones de entradas / salidas**

Clk : in std\_logic;  
pJ,pK,pS,pR : in std\_logic;  
Psalida : out std\_logic;  
pEntrada : in std\_logic);

## Referencia:

J	K	Q	Q <sub>siguiente</sub>
0	0	0	0
0	0	1	1
0	1	X	0
1	0	X	1
1	1	0	1
1	1	1	0
X=no importa			

J	K	Q
0	0	q
0	1	0
1	0	1
1	1	$\bar{q}$

$$Q_{siguiente} = J\bar{Q} + \bar{K}Q$$

Integrado MC74HCT112

## Pasos a seguir

1. Diseñar la entidad y la arquitectura.
2. Dentro del cuerpo de la arquitectura, se debe describir el comportamiento del flipflop con resetasincronico y el dato se sincroniza por flanco ascendente de Clk. Nota: Tener en cuenta que se quiere que al haber con el reset la salida sea 0, y no X como figura en la tabla.
3. Realizar la simulación de todas las posibles combinaciones.

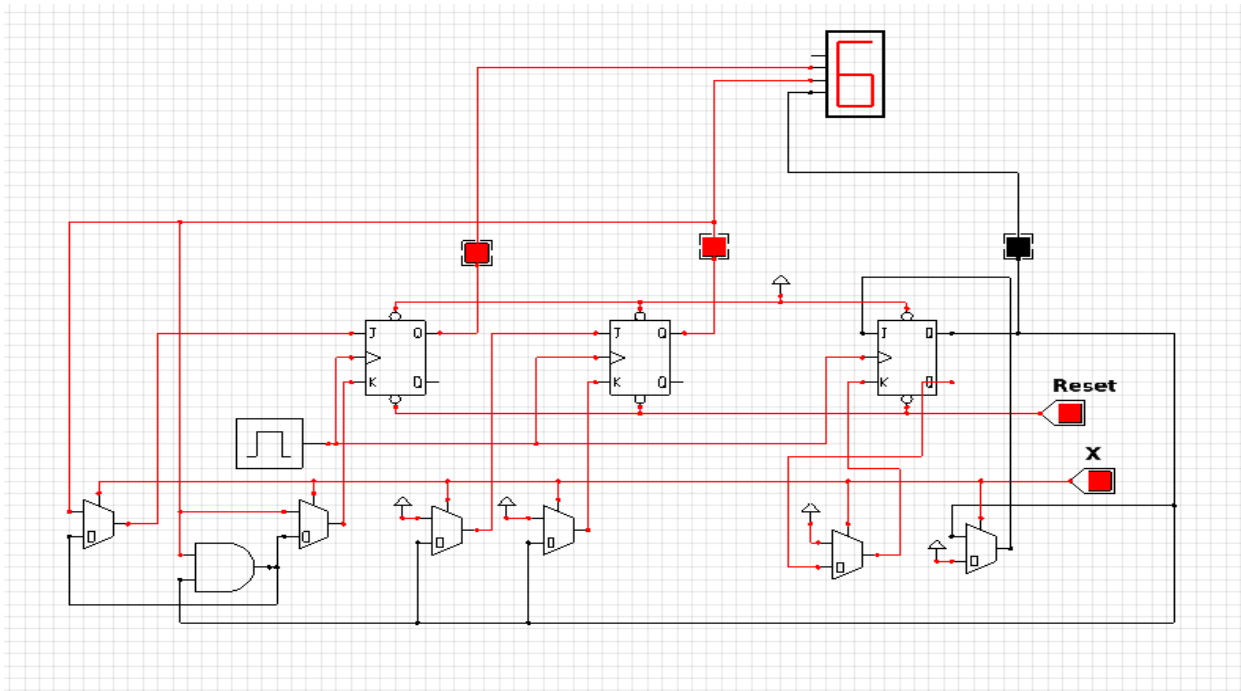
## Practico \_07) Contador Par / Impar utilizando FF JK , compuertas lógicas y multiplexores utilizando instanciación de componentes

1. **Dentro** del cuerpo de la arquitectura, se debe describir el comportamiento de un sumador par/impar mediante la selección de una entrada.
2. Los valores de la cuneta debe estar comprendido desde 0 al 7 ( 3 flipflopJK)con resetasincrónico y el dato se sincroniza por flanco ascendente de Clk,
3. Se debe instanciar el FlipFlop del entrenamiento anterior.
4. Los elementos lógicos (and orxor, etc) utilizados, también deberán ser instanciado.
5. Deberá recordad prácticos realizados en TD1

## Referencia

<https://youtu.be/BV8F7MSKMSM>



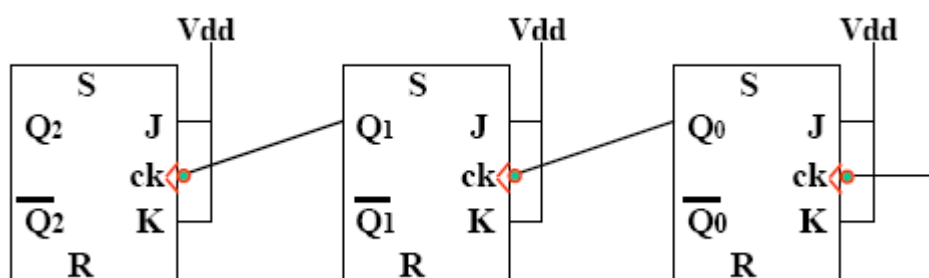


### Practico \_08) Contador asíncrono

Descripción en VHDL de varios FFJK en cascadas para realizar un contador asíncrono módulo 7.  
Utilizar un reset general que inicia la cuenta en 3

Objetivo:

1. Esto realiza un contador asíncrono que nos permite visualizar fácilmente los cambios producidos a la salida por un circuito combi nacional cuando las señales va llegando con retraso
2. Verificar la diferencia en la simulación comportamental y simulación Postfit.
3. Verificar los cambios de estados por los retardos internos en el CPLD (simular Postfit)



### Referencia

<https://www.youtube.com/watch?v=MkjXRQB60Lk>

### Practico \_9) Convertor serie paralelo parametrizable

Objetivo:

Incorporar el concepto descripción parametrizable

## Especificaciones de entradas / salidas

```
generic (  
    L :INTEGER := 8);
```

```
Port ( Clk : in std_logic;
```

```
pEntrada: in std_logic;  
pReset,pCarga : in std_logic;  
pD : in std_logic_vector(L-1 downto 0);  
pQ : out std_logic_vector(L-1 downto 0));
```

### Pasos a seguir

1. Diseñar la entidad y la arquitectura.
  2. Dentro del cuerpo de la arquitectura, se debe describir el comportamiento de un coruito donde los datos entran en serie y se muestran en paralelo. La cantidad de bit paralelo debe ser parametrizable.
  3. Simular la descripción para la salida en paralelo de 8 bit y 16 bit.
- 

## **Practico \_10) Realizar un cronómetro Up/down de 00 a 59**

Objetivo:

Realizar la descripción de divisores de frecuencia para el barrido de los displays a una frec.de 1KHz utilizando un CLK de 15 Khz.

Describir un contador decimal de 00 a 59. Incorporar el concepto de descripción jerárquica.

Utilizar el kit de CPLD para demostrar el funcionamiento.

## Especificaciones de entradas / salidas

A convenir;

### Pasos a seguir

1. Diseñar la entidad y la arquitectura.
  2. Simular el comportamiento.
  3. Realizar el post fit.
  4. Simular post fit.
  5. Descargar en el “Dispositivo de Entrenamiento con CPLD”
  6. Verificar en el hardware el comportamiento logrado.
-