

ESTRUCTURA DE COMPUTADORES (EdC-ISW)

Práctica 3: Diseño e implementación de un contador con visualización en display 7-segmentos

Objetivos de la práctica

Uno de los objetivos generales de la asignatura Estructura de Computadores es llegar a conocer la metodología de diseño de sistemas digitales a nivel de transferencia entre registros. Para ello en las clases de teoría y prácticas, que se desarrollan en clases de aula, se introduce dicha metodología y se aplica a un amplio número de casos de diseño de sistemas digitales. Para completar la formación en este tema es importante no quedarnos exclusivamente en la parte teórica sino que es necesario completarla con conocimientos sobre la implementación real de dichos sistemas digitales.

Los objetivos generales de esta práctica son:

- Familiarizarse con el lenguaje de descripción de hardware Verilog.
- Conocer el entorno de diseño sobre FPGA (herramienta ISE de Xilinx).
- Conocer las herramientas de verificación del diseño desarrollado.
- Desarrollar el proceso de diseño y simulación.

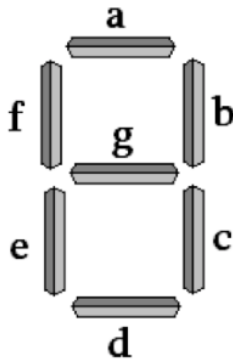
El objetivo específico de esta práctica es el desarrollo de un pequeño sistema secuencial que permita visualizar el estado de cuenta de un contador de 4 bits.

Material:

- Ordenador con XilinX ISE Design Suite 14.3 instalado y placa de entrenamiento de FPGA Basys2

1. ESTUDIO TEÓRICO (Debe presentarse antes de empezar la práctica)

- a) Obtenga la tabla de funcionamiento de un **convertidor binario/7 segmentos**, esto es, obtenga la tabla de un circuito que devuelva una representación en siete segmentos del carácter hexadecimal que representa en binario la entrada. En la representación, **los segmentos activos deben valer 0 y los inactivos 1.**



i3	i2	i1	i0	g	f	e	d	c	b	a

2. TRABAJO EXPERIMENTAL EN EL LABORATORIO

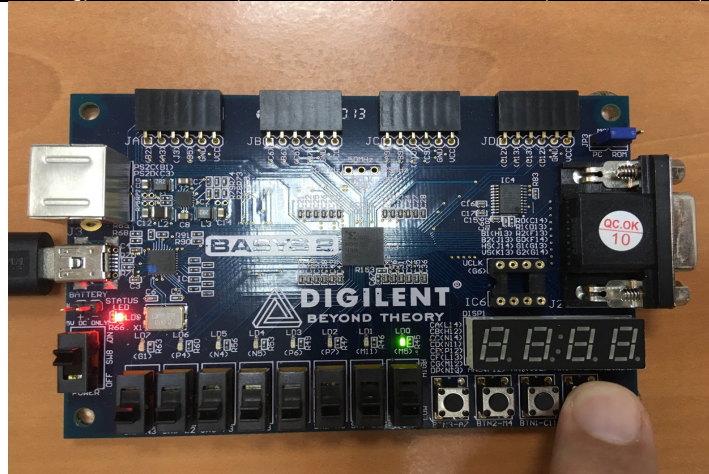
Cree un nuevo proyecto en la herramienta ISE. Añada un nuevo archivo principal en formato Verilog y no olvide agregar también el archivo UCF correspondiente a la BASYS 2. A continuación, y de manera progresiva, vaya realizando los siguientes ejercicios hasta llegar al diseño final (en todos los casos, realice primero una simulación adecuada mediante un banco de pruebas y luego programe el circuito en la placa de entrenamiento BASYS 2 para comprobar su correcto funcionamiento). Consulte el documento “Tutorial de XILINX ISE”, donde se detallan los pasos del proceso completo desde el diseño a la implementación.

AVISE AL PROFESOR PARA QUE VERIFIQUE EL CORRECTO FUNCIONAMIENTO DE CADA EJERCICIO

EJERCICIO 1. Gestión de la botonera

Diseñe, verifique e implemente un sistema que encienda el LED 0 (LD0) de la placa mientras se encuentre pulsado el botón 0 (BTN0).

Name	Value	0 ns	200 ns	400 ns	600 ns	800 ns	1,000 ns
led[0:0]	1	0	1	0	1	0	1
btn[0:0]	1	0	1	0	1	0	1

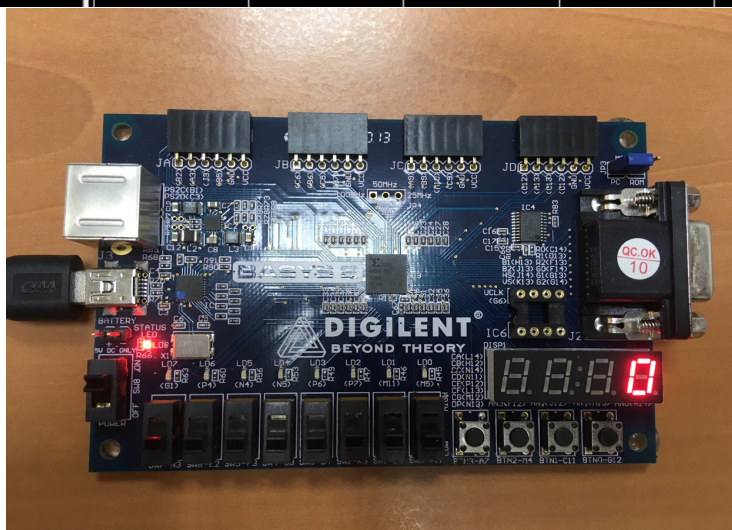


Ejercicio 1. Resultados de la simulación (arriba) e implementación (abajo)

EJERCICIO 2. Salida de datos a display 7-segmentos

Modifique el sistema anterior para que, además, muestre el dígito 0 en la posición de más a la derecha del display 7-segmentos.

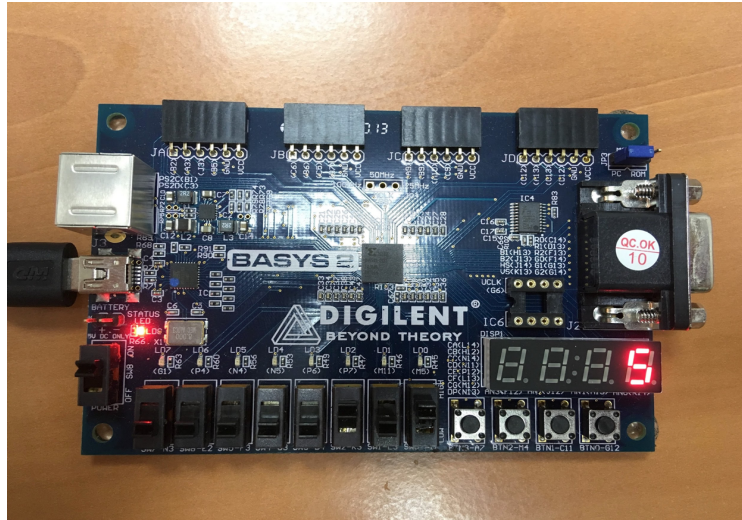
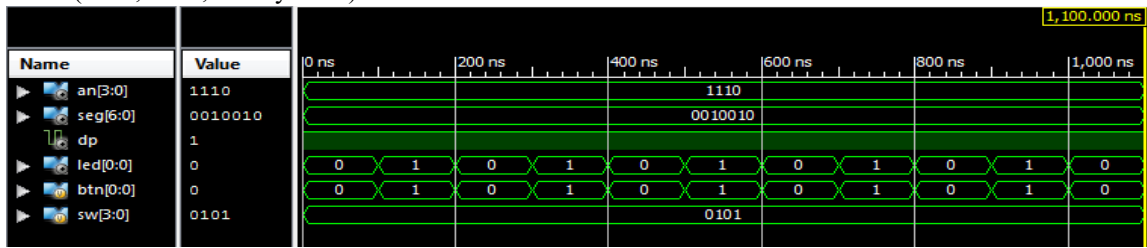
Name	Value	0 ns	200 ns	400 ns	600 ns	800 ns	1,000 ns
an[3:0]	1110			1110			
seg[6:0]	1000000			1000000			
dp	1						
led[0:0]	0	0	1	0	1	0	1
btn[0:0]	0	0	1	0	1	0	1



Ejercicio 2. Resultados de la simulación (arriba) e implementación (abajo).

EJERCICIO 3. Conversión de código binario a 7 segmentos

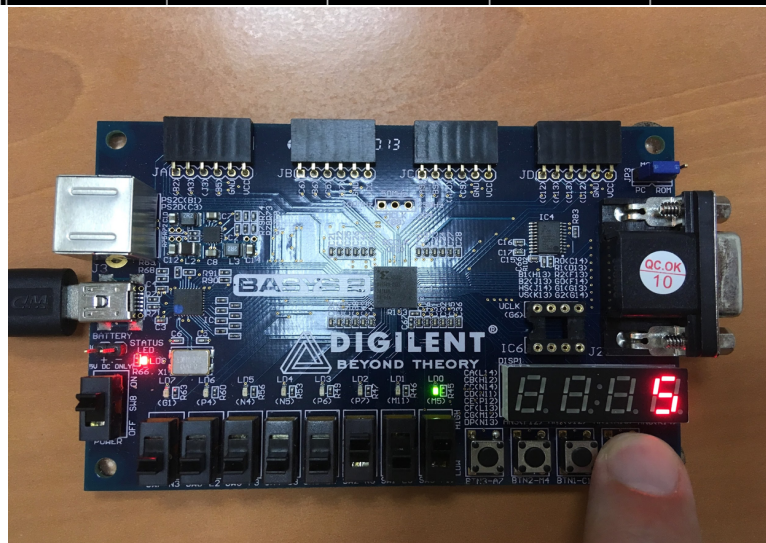
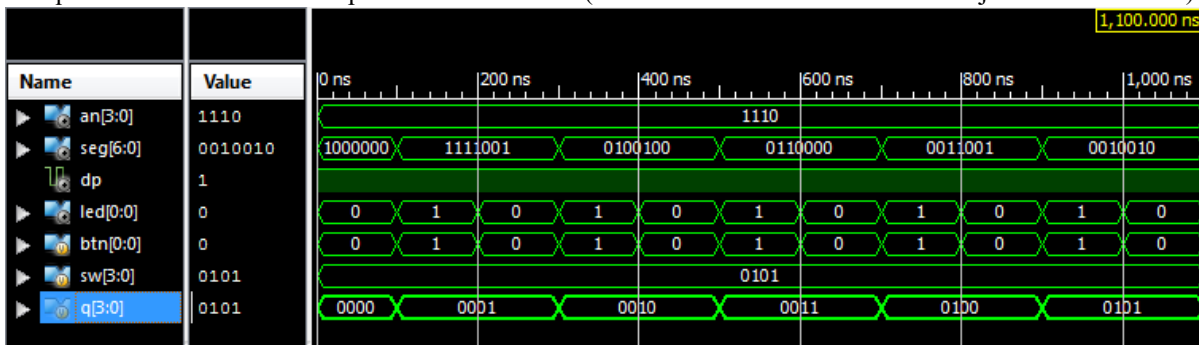
Modifique el sistema anterior de forma que, en lugar del dígito 0, muestre en el display el número binario seleccionado mediante los switches 3 a 0 (SW3, SW2, SW1 y SW0).



Ejercicio 3. Resultados de la simulación (arriba) e implementación (abajo).

EJERCICIO 4. Contador con visualización del estado de cuenta en display de 7-segmentos

Modifique el sistema anterior de forma que, en lugar del número indicado en los switches, muestre la salida de un contador de 4 bits que se incrementa con cada pulsación del botón 0 (utilice el botón 0 como señal de reloj de dicho contador).



Ejercicio 4. Resultados de la simulación (arriba) e implementación (abajo).

OTROS POSIBLES DISEÑOS

EJERCICIO 5. Contador ascendente, con visualización simultánea en binario (elija 4 leds) y en el display de 7-segmentos.

EJERCICIO 6. Contador ascendente/descendente con visualización del estado de cuenta en display de 7-segmentos.

EJERCICIO 7. Contador con cuenta ascendente/descendente, inhibición y carga en paralelo

APÉNDICE: ESQUEMA DE LA PLACA BASYS2 Y FICHERO DE ASOCIACIONES UCF

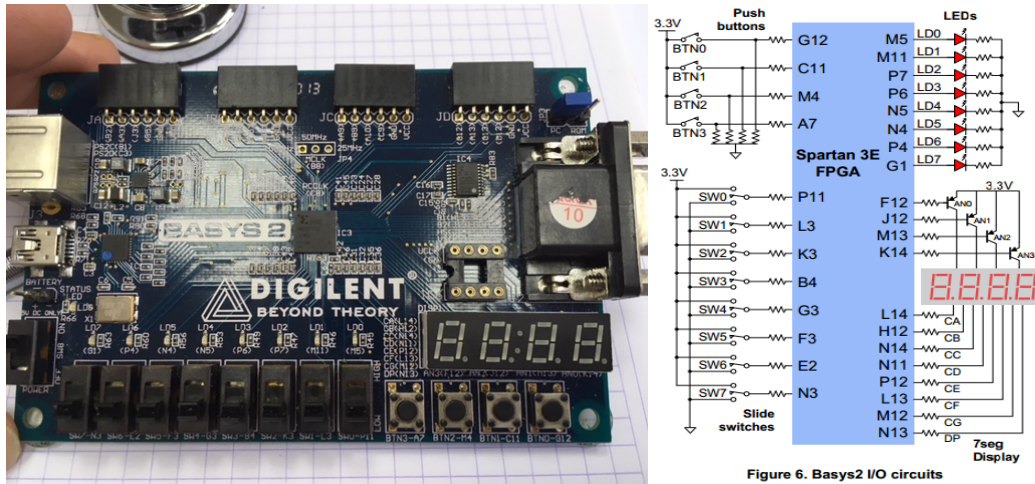


Figure 6. Basys2 I/O circuits

```
# This file is a general .ucf for Basys2 rev C board
# To use it in a project:
# - remove or comment the lines corresponding to unused pins

# Pin assignment for DispCtl
# Connected to Basys2 onBoard 7seg display
# NET "seg<0>" LOC = "L14"; # Bank = 1, Signal name = CA
# NET "seg<1>" LOC = "H12"; # Bank = 1, Signal name = CB
# NET "seg<2>" LOC = "N14"; # Bank = 1, Signal name = CC
# NET "seg<3>" LOC = "N11"; # Bank = 2, Signal name = CD
# NET "seg<4>" LOC = "P12"; # Bank = 2, Signal name = CE
# NET "seg<5>" LOC = "L13"; # Bank = 1, Signal name = CF
# NET "seg<6>" LOC = "M12"; # Bank = 1, Signal name = CG
# NET "dp" LOC = "N13"; # Bank = 1, Signal name = DP

#NET "an<3>" LOC = "K14"; # Bank = 1, Signal name = AN3
#NET "an<2>" LOC = "M13"; # Bank = 1, Signal name = AN2
#NET "an<1>" LOC = "J12"; # Bank = 1, Signal name = AN1
#NET "an<0>" LOC = "F12"; # Bank = 1, Signal name = AN0

# Pin assignment for LEDs
#NET "Led<7>" LOC = "G1"; # Bank = 3, Signal name = LD7
#NET "Led<6>" LOC = "P4"; # Bank = 2, Signal name = LD6
#NET "Led<5>" LOC = "N4"; # Bank = 2, Signal name = LD5
#NET "Led<4>" LOC = "N5"; # Bank = 2, Signal name = LD4
#NET "Led<3>" LOC = "P6"; # Bank = 2, Signal name = LD3
#NET "Led<2>" LOC = "P7"; # Bank = 3, Signal name = LD2
#NET "Led<1>" LOC = "M11"; # Bank = 2, Signal name = LD1
#NET "Led<0>" LOC = "M5"; # Bank = 2, Signal name = LD0

# Pin assignment for SWs
#NET "sw<7>" LOC = "N3"; # Bank = 2, Signal name = SW7
#NET "sw<6>" LOC = "E2"; # Bank = 3, Signal name = SW6
#NET "sw<5>" LOC = "F3"; # Bank = 3, Signal name = SW5
#NET "sw<4>" LOC = "G3"; # Bank = 3, Signal name = SW4
#NET "sw<3>" LOC = "B4"; # Bank = 3, Signal name = SW3
#NET "sw<2>" LOC = "K3"; # Bank = 3, Signal name = SW2
#NET "sw<1>" LOC = "L3"; # Bank = 3, Signal name = SW1
#NET "sw<0>" LOC = "P11"; # Bank = 2, Signal name = SW0

#NET "btn<3>" LOC = "A7"; # Bank = 1, Signal name = BTN3
#NET "btn<2>" LOC = "M4"; # Bank = 0, Signal name = BTN2
#NET "btn<1>" LOC = "C11"; # Bank = 2, Signal name = BTN1
#NET "btn<0>" LOC = "G12"; # Bank = 0, Signal name = BTN0
#NET "btn<0>" CLOCK_DEDICATED_ROUTE = FALSE;

# ETC
```