



Introducción al diseño con FPGA

>> >> >>

PLACA DE DESARROLLO DEO

<< << <<

EJERCICIOS INTRODUCTORIOS

Ejercicio 1

Utilizando el kit de desarrollo encender los leds con el siguiente patrón.

PUERTO	ESTADO
LEDG[0]	ENCENDIDO
LEDG[1]	APAGADO
LEDG[2]	ENCENDIDO
LEDG[3]	APAGADO
LEDG[4]	APAGADO
LEDG[5]	ENCENDIDO
LEDG[6~9]	APAGADOS

Ejercicio 2

Utilizando algunos de los leds LEDG, los pulsadores BUTTON y los switch's SW implementar los siguientes algoritmos:

1.

```
Si SW(4) está en alto y BUTTON(2) está en bajo entonces
    LEDG(0) encendido
    LEDG(1) encendido
    LEDG(2~9) apagados
SiNo
    LEDG(0~9) apagados
Fin Si
```

2.

```
Si SW(0) está en alto entonces
    Si SW(1) está en bajo entonces
        LEDG(0) y LEDG(1) encendidos
        LEDG(2~9) apagados
    SiNo
        LEDG(2) y LEDG(3) encendidos
        LEDG(0~1) y LEDG(4~9) apagados
    Fin Si
SiNo
    Si BUTTON(2) está en bajo entonces
        LEDG(1), LEDG(3) y LEDG(5) encendidos
        Otros LEDG apagados
    SiNo
        LEDG(0~9) apagados
    Fin Si
Fin Si
```

Ejercicio 3

1. Utilizando uno de los display de 7 segmentos HEX y los switches SW(3~0) de la placa de desarrollo mostrar en el display el número codificado en binario a través de los switch's. Por ejemplo, si SW(3~0) = "0110" en el display deberá leerse el número 6; si SW(3~0) = "1100" en el display deberá leerse el número "C" (en hexadecimal), etc.
2. A la implementación del inciso anterior agregar una señal de control utilizando el SW(7) que permita apagar el display. Por ejemplo, cuando SW(7) = "0" el display se encenderá mostrando el número codificado en SW(3~0), caso contrario deberá permanecer apagado.
3. A la implementación del inciso 1, agregar una señal de control utilizando los SW(9~8) que permita seleccionar el display que se utilizará para mostrar el número codificado en SW(3~0). Por ejemplo, si SW(9~8) = "00" se utilizará el display HEX0, si SW(9~8) = "01" se utilizará el display HEX1, etc. Los displays que no estén activos deberán permanecer apagados.
4. Modificar la implementación del inciso 1 para que, en lugar de mostrar los números mayores que 9 en hexadecimal, mostrarlos en decimal en formato BCD utilizando dos displays de 7 segmentos (uno para las unidades y otro para las decenas).

Ejercicio 4

1. Utilizando la librería aritmética *ieee.std_logic_unsigned* implementar un divisor de frecuencia utilizando un contador síncronico que permita dividir la señal de reloj CLOCK_50 de 50MHz para obtener a la salida una frecuencia lo más cercana posible a 0,5Hz. Conectar las salidas de los 4 bits más significativos del contador a los LED's cuyos pines se muestran en la tabla al final del enunciado. Los demás LED's deberán permanecer apagados.
 - ¿Cuál es la frecuencia más cercana lograda? ¿Cuántos bits son necesarios para lograr dicha frecuencia? ¿Cuál es el error cometido?
 - ¿Con qué frecuencia se encenderá cada uno de los LED's?
 - Sin implementarlo, ¿cómo podría mejorarse la precisión para lograr una frecuencia de salida de 0,5Hz?
2. Agregar al divisor de frecuencia señales de reset y habilitación utilizando para ello los pines especificados. La señal de *reset* restablecerá el contador a cero en todos sus bits mientras que la señal de habilitación deberá congelar el estado del contador hasta tanto se habilite nuevamente.

PIN	DESCRIPCIÓN	ACTIVACIÓN
G5	Reset	Bajo
H7	Habilitación	Alto
C2	Led 3 (MSB)	-
C1	Led 2	-
E1	Led 1	-
F2	Led 0 (LSB)	-

Ejercicio 5

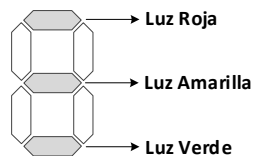
1. Aprovechando la implementación del Ejercicio 5 utilizar la señal de 0,5Hz como señal de habilitación de un contador tipo Anillo de 8 bits agregando las señales de control que sean convenientes. Conectar cada una de las salidas del contador a los LEDG(7~0).
2. Repetir la implementación del inciso anterior para un contador tipo Johnson.

EJERCICIOS OBLIGATORIOS

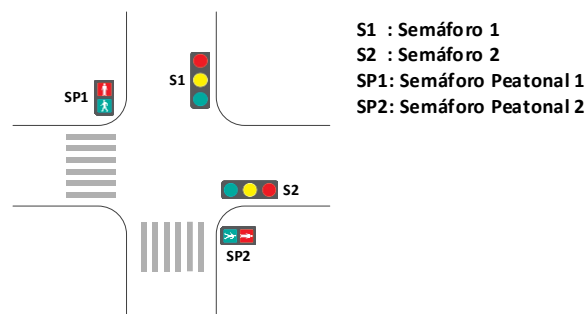
Ejercicio 6

Dado el ejemplo visto en clase de la implementación de un sistema de control de semáforos:

1. Implementar el sistema tal como se vio en clase en el kit de desarrollo y verificar su funcionamiento. De ser necesario modificar los pines utilizados para adaptarlos a los disponibles en el kit de desarrollo. Para modelar las luces de los semáforos utilizar dos de los displays 7-segmentos (uno para cada semáforo) de la siguiente manera:



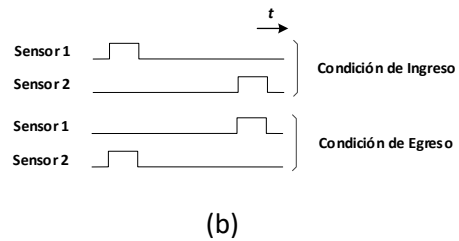
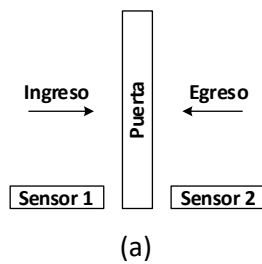
2. Modificar el diseño con el objetivo de incorporar la funcionalidad de controlar adicionalmente dos semáforos peatonales como se muestra en la siguiente figura y cuyo comportamiento deberá cumplir con las siguientes premisas. Utilizar los otros dos displays 7-segmentos para modelarlos.
 - Las luces verdes de SP1 y SP2 deberán encenderse 3 segundos después que las luces rojas de S2 y S1 respectivamente.
 - Las luces rojas de SP1 y SP2 deberán encenderse 3 segundos antes que las luces verdes de S2 y S1 respectivamente.
 - Durante los últimos 5 segundos de la fase de luz verde de SP1 y SP2 la luz deberá ser intermitente con una frecuencia de 2Hz.



3. Realizar el diagrama de estados del nuevo diseño y un diagrama de flujo que muestre claramente el algoritmo implementado.

Ejercicio 7

Se necesita diseñar un sistema que permita llevar la cuenta de cuántas personas permanecen en un local. Para ello se utilizan dos sensores instalados en la puerta que permiten detectar cuándo una persona sale o entra al mismo como se muestra en la Figura (a). Cada sensor posee una salida que provee un pulso cuando se produce una detección. Cuando una persona entra al local el contador del sistema se deberá incrementar en uno y cuando sale se deberá decrementar en la misma cantidad. Para detectar cuándo alguien entra o sale del local se debe analizar la secuencia de pulsos arrojada por los sensores como se muestra en la Figura (b):



1. Realizar un diagrama en bloques y un diagrama de flujo que describan claramente la solución propuesta para resolver el problema.
2. Profundizar el análisis anterior y calcular la cantidad de registros que se requerirán para la implementación.
3. Implementar el sistema utilizando el kit de desarrollo teniendo en cuenta los siguientes requerimientos y sugerencias:
 - Utilizar pulsadores o switch's para emular la salida de los sensores.
 - El rango de cuenta deberá ser de 0 a 99.
 - El estado de la cuenta deberá poder visualizarse en decimal en los displays de 7-segmentos.
 - El sistema deberá poseer una señal de *reset* que permita restablecer la cuenta a cero en cualquier momento.
4. Obtener de los reportes de compilación la cantidad de recursos utilizados (FF's, LUT's, MUX's, etc.) y el camino crítico de la implementación. Para el caso de los FF's comparar el número obtenido del reporte de compilación con el calculado en el inciso 2.

Ejercicio 8

Diseñar un cronómetro digital que permita hacer mediciones de tiempo y que cumpla con los siguientes requerimientos:

- Deberá ser capaz de contar centésimas de segundo, décimas de segundo y segundos.
 - La cuenta deberá iniciarse en cero y el rango deberá ser hasta 99 segundos.
 - Deberá incluir una señal de *reset* que permita restablecer la cuenta a cero en cualquier momento y una señal de habilitación para congelar el valor mostrado en los displays hasta tanto se vuelva a habilitar el sistema.
1. Realizar un diagrama de flujo que describa de forma simple el algoritmo a implementar.
 2. Realizar un diagrama en bloques detallado de la solución propuesta para implementar el algoritmo y que indique claramente qué recursos de los vistos durante la cursada utilizará para la implementación.
 3. Implementar el diseño del diagrama anterior en el kit de desarrollo teniendo en cuenta las siguientes cuestiones:
 - Utilizar los displays 7-segmentos para visualizar las centésimas, decimas, unidades y decenas de segundos.
 - Utilizar los switch's para las señales de reset y habilitación.
 - Puede utilizar la librería aritmética vista en clase para facilitar la implementación.
 4. Obtener de los reportes de compilación los recursos necesarios para la implementación y el camino crítico.