	<p>Universidad de Costa Rica Escuela de ingeniería Eléctrica Laboratorio 4</p>	<p><b>EIE</b> Escuela de Ingeniería Eléctrica</p>
<p>IE0424: Laboratorio de Circuitos Digitales II-2020</p>		

## Objetivo General

Introducir el manejo de las entradas y salidas en una FPGA.

## Objetivos específicos

- Investigar el funcionamiento de la tarjeta de desarrollo FPGA Nexys 4.
- Utilizar las herramientas del Xilinx Vivado.
- Conocer y aplicar el flujo de diseño para sistemas basados en FPGA.

## Anteproyecto

- Investigue cómo funcionan los LEDs de siete segmentos de la tarjeta Nexys 4 DDR y describa los pasos y tiempos que se necesitan para desplegar números.
- Investigue qué son bancos en las FPGAs y para qué sirven.
- Localice el esquemático de la tarjeta Nexys 4 DDR y ubique:
  - El número de parte de los LEDs de 7 segmentos.
  - Los números de bancos a los que están conectados cada uno de los ánodos y cátodos de los LEDs de 7 segmentos.

## Propuesta del problema.

Proceda ahora a obtener el código inicial del proyecto. Para ello, siga el siguiente enlace:

<https://classroom.github.com/g/tjiPS9Ek>

De la misma forma que en el Laboratorio 1, 2 y 3, autorice la aplicación, busque su nombre (si su nombre no aparece, contacte al profesor) y seleccione un equipo (si su equipo no aparece listado, proceda a crear uno). Una vez finalizado un repositorio debió haber sido creado. Este repositorio va a ser el repositorio con el que va a trabajar y va a ser uno de los entregables de este laboratorio.

Proceda a clonar el repositorio que ha sido creado:

```
$ git clone https://github.com/ucr-ie-0424/...
```

La dirección completa de su repositorio debe aparecer seleccionando el botón verde que dice *Code*.

El código que se provee es el mismo código que se usó como punto de inicio del Laboratorio 1, 2 y 3.

En este laboratorio se trabajará de nuevo con los dispositivos de entrada y salida de la FPGA.

### **Ejercicio 1:**

Lance una simulación y mida cuánto tiempo transcurre entre dos cambios consecutivos en la salida *out\_byte* y analice cuánto debería ser el tiempo mínimo para que el ojo humano pueda detectar con facilidad cambios de este tipo si la salida se conectara a LEDs.

**Sintetice el código y cargue el circuito en la tarjeta de desarrollo. ¿Qué nota en los LEDs?**

Seguidamente, realice cambios en el firmware de forma tal que se pueda ver con facilidad las transiciones de los LEDs y se pueda apreciar una cuenta ascendente en el circuito implementado en la tarjeta de desarrollo. Llame a este firmware *firmware\_lab4\_part1.c*.

Para este ejercicio, suba un video a Youtube con el resultado. Coloque el video como 'Unlisted' y para efectos de la Bitácora, coloque un enlace al video.

### **Ejercicio 2:**

Realice los cambios necesarios para desplegar el contador ascendente utilizando el display de 7 segmentos de la tarjeta de desarrollo. Para esto deberá de cambiar el archivo de constraints. La cuenta debe de aparecer en decimal en los LEDs de 7 segmentos.

Para ello, instancie en *system.v* un módulo con el nombre **7\_segment\_dec**. Este módulo debe de tener 3 entradas:

- Un número a desplegar en los LEDs de 7 segmentos de 32 bits.
- La señal de clock.
- La señal de reset.

Adicionalmente debe de tener dos salidas de 8 bits:

- El valor de los cátodos de los LEDs de 7 segmentos.
- El valor de los ánodos de los LEDs de 7 segmentos.

Note que no todos los números de 32 bits se podrán desplegar. Si existiese un número que no alcance a desplegarse en decimal en los LEDs de 7 segmentos, despliegue el número máximo a representar (99 999 999).

De igual forma que el ejercicio anterior, suba un video a Youtube con el resultado. Coloque el video como 'Unlisted' y para efectos de la Bitácora, coloque un enlace al video.

### Ejercicio 3:

Realice los cambios necesarios para desplegar el contador utilizando el display de 7 segmentos de la tarjeta de desarrollo pero esta vez la cuenta deberá ser en hexadecimal.

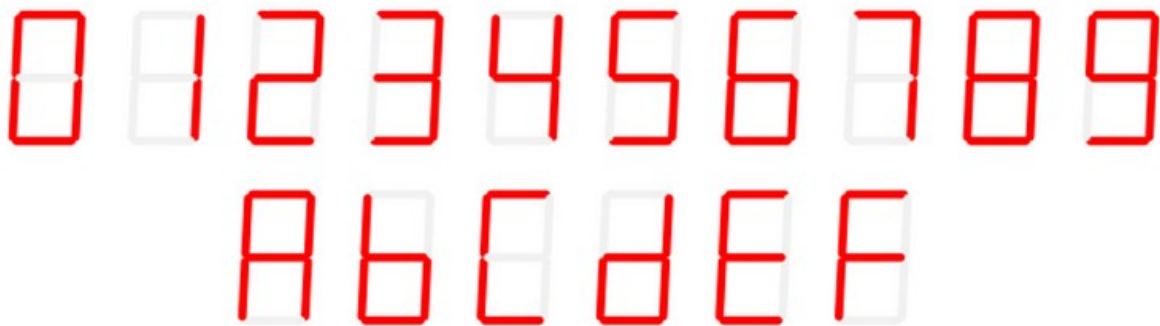
Para ello, instancie en system.v un módulo con el nombre `7_segment_hex`. Este módulo debe de tener 3 entradas:

- Un número a desplegar en los LEDs de 7 segmentos de 32 bits.
- La señal de clock.
- La señal de reset.

Adicionalmente debe de tener dos salidas de 8 bits:

- El valor de los cátodos de los LEDs de 7 segmentos.
- El valor de los ánodos de los LEDs de 7 segmentos.

Para la representación en hexadecimal de los números utilice:



Fuente: <https://d2vlcm61l7u1fs.cloudfront.net/media%2Fb4f%2Fb4f6d2a0-318f-4d2d-966e-4dee1e9d92ca%2FphpalKdNL.png>

De igual forma que los ejercicios anteriores, suba un video a Youtube con el resultado. Coloque el video como 'Unlisted' y para efectos de la Bitácora, coloque un enlace al video.

### Ejercicio 4:

Finalmente, agregue al archivo de constraints una entrada más para poder leer el valor del switch J15. Seguidamente, instancie en system.v un módulo con el nombre `7_segment_switch` de forma tal que:

- Si el switch J15 está en 1, la cuenta en los LEDs de 7 segmentos debe de desplegarse en decimal.
- Si el switch J15 está en 0, la cuenta en los LEDs de 7 segmentos debe de desplegarse en hexadecimal.

Este módulo debe de tener 4 entradas:

- Un número a desplegar en los LEDs de 7 segmentos de 32 bits.
- La señal de clock.
- La señal de reset.
- Un bit que indique si el número a desplegar debe de mostrarse en decimal o en hexadecimal.

Adicionalmente debe de tener dos salidas de 8 bits:

- El valor de los cátodos de los LEDs de 7 segmentos.
- El valor de los ánodos de los LEDs de 7 segmentos.

Note, que el cambio en el switch no debe provocar un reinicio en la cuenta, sino solamente del formato del número que se despliega en los LEDs de 7 segmentos.

De igual forma que los ejercicios anteriores, suba un video a Youtube con el resultado. Coloque el video como 'Unlisted' y para efectos de la Bitácora, coloque un enlace al video.