	<p>Universidad de Costa Rica Escuela de ingeniería Eléctrica Experimento 3</p>	<p><b>EIE</b> Escuela de Ingeniería Eléctrica</p>
<p>IE0424: Laboratorio de Circuitos Digitales II-2019</p>		

## Objetivo General

Utilizar los periféricos de la tarjeta de desarrollo.

## Objetivos específicos

- Investigar cómo utilizar periféricos en la tarjeta de desarrollo.
- Realizar interfaz entre el usuario y la tarjeta Nexys 4 DDR.

## Propuesta del problema.

El objetivo de este experimento es practicar el uso de dispositivos de entrada y salida. Se va utilizar el monitor VGA en modo 640x480 con 16 bits de color. También se va a utilizar un teclado.

## Preguntas a responder en el anteproyecto.

De acuerdo a la cantidad de señales utilizadas para el color, ¿cuántos colores se pueden mostrar? ¿Cómo se definen los colores? Dé ejemplos de secuencias de bits y el color correspondiente, indique al menos 5.

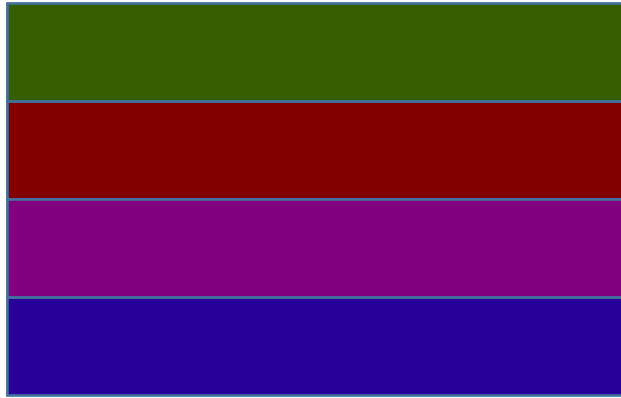
¿Cuáles son los cambios necesarios al archivo de *constraints* para correctamente manejar los pines correspondientes al puerto VGA?

Explique cómo podría conectarse un teclado a la tarjeta de desarrollo. ¿Qué protocolo se utilizaría para leer datos provenientes de un teclado? ¿Explique cómo funcionaría a alto nivel el paso de datos desde un teclado hacia la FPGA?

¿Cuáles son los cambios necesarios al archivo de *constraints* para correctamente manejar los pines correspondientes al puerto que emula el protocolo PS2?

## Ejercicio 1 (Obligatorio) (20%):

En este ejercicio se va a escribir un patrón de varias filas de colores en el monitor, como se muestra a continuación.



Deberá estudiar el modo VGA 640x480. Este es en realidad muy sencillo.

Básicamente deberá generar un reloj de 25MHz. En cada flanco del reloj de 25MHz se deberá leer el valor de una memoria de video correspondiente al color en una fila y columna determinada.

Cada vez que se llega al final de una fila, la señal HS (*Horizontal Synch*), la cual es activa en cero, deberá cambiar su valor a uno para indicarle al monitor que debemos cambiar de fila.

Cada vez que se llega a la última fila, la señal VS (*Vertical Synch*), la cual es activa en cero, deberá cambiar su valor a uno para indicarle al monitor que debemos regresar a la primera fila.

El capítulo 9 del *Manual de Referencia de la tarjeta Nexys 4 DDR* tiene una explicación de estos conceptos con mayor detalle.

Revise los pines que se necesita conectar a agréguelos al archivo de *constraints*.

Para este laboratorio, deberá crear una memoria RAM para guardar la memoria de video. Note que la FPGA cuenta con recursos limitados, así que deberá guardar solo lo necesario.

Puede definir una memoria RAM de un canal de la siguiente manera:

```
module RAM_SINGLE_READ_PORT # ( parameter DATA_WIDTH= 16,
parameter ADDR_WIDTH=8, parameter MEM_SIZE=8 )
(
    input wire                Clock,
    input wire                iWriteEnable,
    input wire[ADDR_WIDTH-1:0] iReadAddress,
    input wire[ADDR_WIDTH-1:0] iWriteAddress,
    input wire[DATA_WIDTH-1:0] iDataIn,
    output reg [DATA_WIDTH-1:0] oDataOut
)
```

```

);

reg [DATA_WIDTH-1:0] Ram [MEM_SIZE:0];
always @(posedge Clock)
begin
    if (iWriteEnable)
        Ram[iWriteAddress] <= iDataIn;

    oDataOut <= Ram[iReadAddress];
end
endmodule

```

Recuerde instanciarla de manera que cada posición de memoria solo guarde la cantidad de bits necesarias que corresponden al color, y que tenga suficientes posiciones para una resolución de 640x480.

Puede instanciar la memoria de la siguiente manera:

```

RAM_SINGLE_READ_PORT # (12,24,640*480) VideoMemory
(
    .Clock(          Clock          ),
    .iWriteEnable(   rVGAwritEnable ),
    .iReadAddress(   24'b0 ),
    .iWriteAddress( {wSourceData1[7:0],wSourceData0} ),
    .iDataIn(        wInstruction[23:21]        )
    .oDataOut( {oVGA_R,oVGA_G,oVGA_B} )
);

```

El ejemplo anterior muestra cómo la salida de esta memoria está conectada a los pines de salida del VGA para los 3 colores: Rojo, Verde y Azul.

Del lado del firmware, se usarán 3 direcciones de memoria para interactuar con la memoria de video:

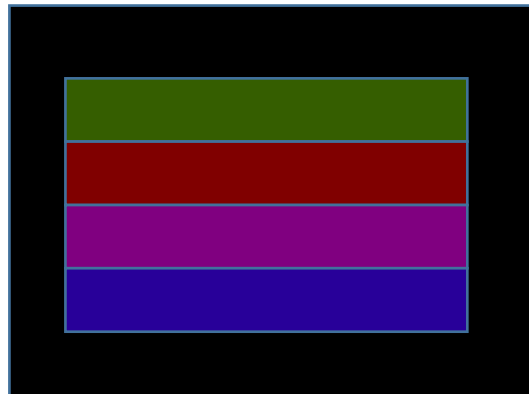
- 0x10000000: Posición en la memoria de video a escribir. Los 16 bits menos significativos indicarán la fila del pixel, mientras que los 16 bits más significativos la columna del pixel.
- 0x10000004: Color a pintar en el pixel cuya posición corresponda al indicado en la dirección anterior. El color es de 12 bits. Use los bits menos significativos.
- 0x10000008: Registro de control. Si al registro se le escribe un 1, el pixel indicado en la posición que indica 0x10000000 se pintará con el color indicado en 0x10000004.

Con las direcciones anteriores, escriba un firmware que le permita pintar la pantalla con los colores requeridos en este ejercicio.

Recuerde que debe iniciar la memoria VGA con un valor conocido, ya que se quiere que lleguen valores X a los pines de monitor. Por ejemplo el color negro es un buen valor de inicio.

¿Es suficiente la memoria de la FPGA para almacenar la memoria de video para una resolución de 640x480? Comente.

En caso de que su conclusión sea que la memoria es insuficiente, puede usar una memoria más pequeña y encerrar la imagen resultante en un marco de un solo color como se muestra a continuación.



Para esto, puede declarar 2 contadores que sean pines de salida de su controlador de video: oCurrentRow y oCurrentCol.

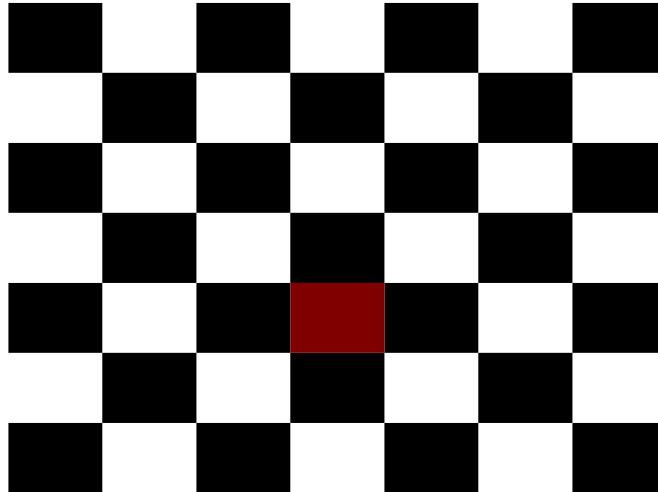
Estos contadores le indicarán la fila y columna correspondientes al pixel que se esta desplegando en un momento determinado del tiempo.

Usando estas dos señales, usted podrá definir un “marco” de un solo color (puede ser negro) el cual se despliega alrededor su imagen principal. En siguiente pseudo código ilustra esta idea para un marco de 100 pixeles:

```
assign {oRed,oGreen,oBlue} = (   wCurrentRow < 100 ||   wCu-
rrentRow > 540 || CurrentCol < 100 || CurrentCol > 380 ) :
{0,0,0} : wColorFromVideoMemory;
```

### Ejercicio 2 (Obligatorio) (30%):

El objetivo es generar un patrón de tablero de ajedrez en el monitor. Uno de los cuadrados del tablero será de color rojo como se muestra en el siguiente diagrama:



Conecte y use los botones de la tarjeta de desarrollo para mover el cuadro rojo hacia arriba, abajo, derecha o izquierda. La Nexys 4 DDR cuenta con 5 botones: el botón central no se usará.

Para este ejercicio, no es necesario usar *firmware* para pintar los píxeles de la VGA.

La descripción de los botones la puede encontrar en el capítulo 10 del *Manual de Referencia de la tarjeta Nexys 4 DDR*.

### Ejercicio 3 (Obligatorio) (50%):

En el ejercicio anterior se usó los botones de la FPGA. En este ejercicio, deberá conectar un teclado a la tarjeta de desarrollo. Asimismo, deberá usar las teclas A,S,D,W para mover el cuadrado rojo por el tablero. La asignación de las teclas es la siguiente:

Tecla	Dirección
A	Izquierda
S	Abajo
D	Derecha
W	Arriba

Encontrará información de cómo usar el teclado PS2 en el capítulo 8 del *Manual de Referencia de la tarjeta Nexys 4 DDR*.

**Ejercicio 4 (Opcional):**

Como notó del Ejercicio 1, guardar una memoria de video para una resolución de 640x480 consume gran cantidad de los recursos de la FPGA.

Sin embargo, la tarjeta de desarrollo cuenta con una memoria DDR2. Puede intentar conectarla para usarla como su memoria de video.