

Objetivo General

Utilizar los periféricos de la tarjeta de desarrollo.

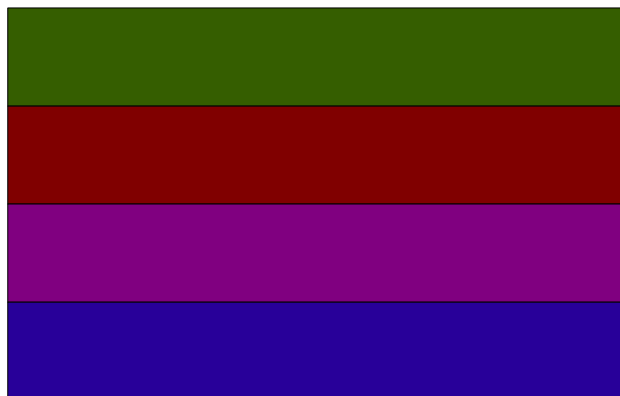
Objetivos Específicos

- Investigar cómo utilizar periféricos en la tarjeta de desarrollo.
- Realizar interfaz entre el usuario y Spartan 3E Starter Kit

El objetivo de este experimento es practicar el uso de dispositivos de entrada y salida. Se va a utilizar el monitor VGA en modo 640x480 con 3 bits de color. También se va a utilizar un teclado.

Ejercicio 1 (Obligatorio):

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.

Basicamente deberá generar un reloj de 25Mhz. 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 5 del **Spartan-3 FPGA Starter Kit Board User Guide** tiene una explicación de estos conceptos con mayor detalle.

Puede ver la asignación de pines de la tabla 5.1 de la guía de usuario del Spartan 3E

Signal	FPGA Pin
Red (R)	R12
Green (G)	T12
Blue (B)	R11
Horizontal Sync (HS)	R9
Vertical Sync (VS)	T10

Las señales R,G y B son de 1 bit. La siguiente tabla muestra los posibles colores:

R	G	B	Color
0	0	0	Negro
0	0	1	Azul
0	1	0	Verde
0	1	1	Cian
1	0	0	Rojo
1	0	1	Magenta
1	1	0	Amarillo
1	1	1	Blanco

Para eso será necesario que lleve a cabo las siguientes defneciones en el archivo de Definitions.v

```
`define COLOR_BLACK 3'b000
`define COLOR_BLUE 3'b001
`define COLOR_GREEN 3'b010
`define COLOR_CYAN 3'b011
`define COLOR_RED 3'b100
`define COLOR_MAGENTA 3'b101
`define COLOR_YELLOW 3'b110
`define COLOR_WHITE 3'b111
```

Además deberá crear una memoria RAM para guardar la memoria de video. Note que el 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 3 bits, y que tenga suficientes posiciones para una resolución de 640x480. Puede instanciar la memoria de la siguiente manera:

```
RAM_SINGLE_READ_PORT # (3,24,640*480) VideoMemory
(
    .Clock(      Clock      ),
    .iWriteEnable( rVGAWriteEnable ),
    .iReadAddress( 24'b0 ),
    .iWriteAddress( {wSourceData1[7:0],wSourceData0} ),
    .iDataIn(      wInstruction[23:21]      )
    .oDataOut( {oVGA_R,oVGA_G,oVGA_B} )
);
```

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

Deberá agregar una instrucción que permita escribir un pixel a una posición específica de la memoria de video. El formato de la instrucción puede ser como el siguiente:

```
`VGA, `COLOR_BLUE, `R4, `R1
```

En este caso la instrucción VGA escribe el valor AZUL, a la posición de memoria resultante de la concatenación del contenido de los registros R4 y R1. Puede entonces pensar en R4 y R1 como punteros.

Recuerde que debe iniciar la memoria VGA con un valor conocido, ya que no queremos que lleguen valores “x” a los pines de monitor. Por ejemplo COLOR_BLACK es un buen valor para iniciar.

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

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.



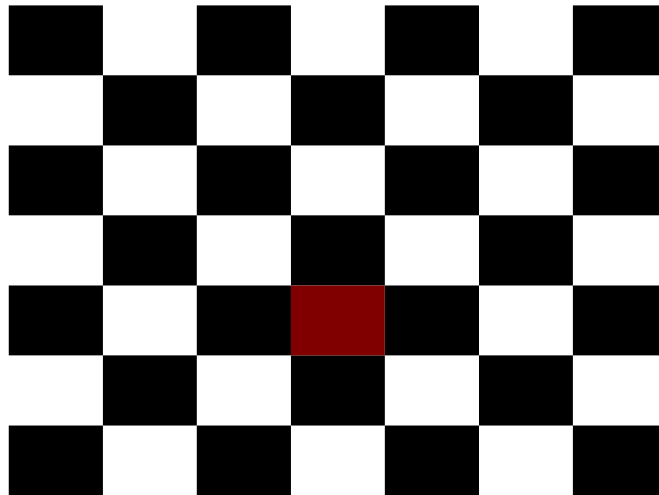
Para esto, puede declarar 2 contadores que sean pines de salida de su controlador de video, oCurrentRow y oCurrentCol, Estos contadores le indicarán la fila y columna correspondientes al pixel que se esta desplegando en un momento determinado del tiempo. Usando estás dos señales, usted podrá definir un “marco” de un solo color (puede ser negro) el cual se despliga alrededor su imagen principal. En siguiente seudo código ilustra esta idea para un marco de 100 píxeles:

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

Ejercicio 2 (Obligatorio):

Para este ejercicio deberá conectar un teclado PS2 al la Tarjeta de desarrollo del spartan 3E.

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



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	Valor
A	Izquierda
S	Abajo
D	Derecha
W	Arriba

Encontrará información de como usar el teclado PS2 en el capítulo 6 de la guía de usuario del spartan 3E.

Ejercicio 4 (Opcional):

Como notó del ejercicio 1, guardar una memoria de video para una resolución de 640x480 consume gran cantidad de los recursos del FPGA. Sin embargo algunos modelos de la tarjeta de desarrollo de Spartan 3E cuentan con una memoria SRAM.

Si su tarjeta siene SRAM, puede intentar conectarla para usarla como su memoria de video.