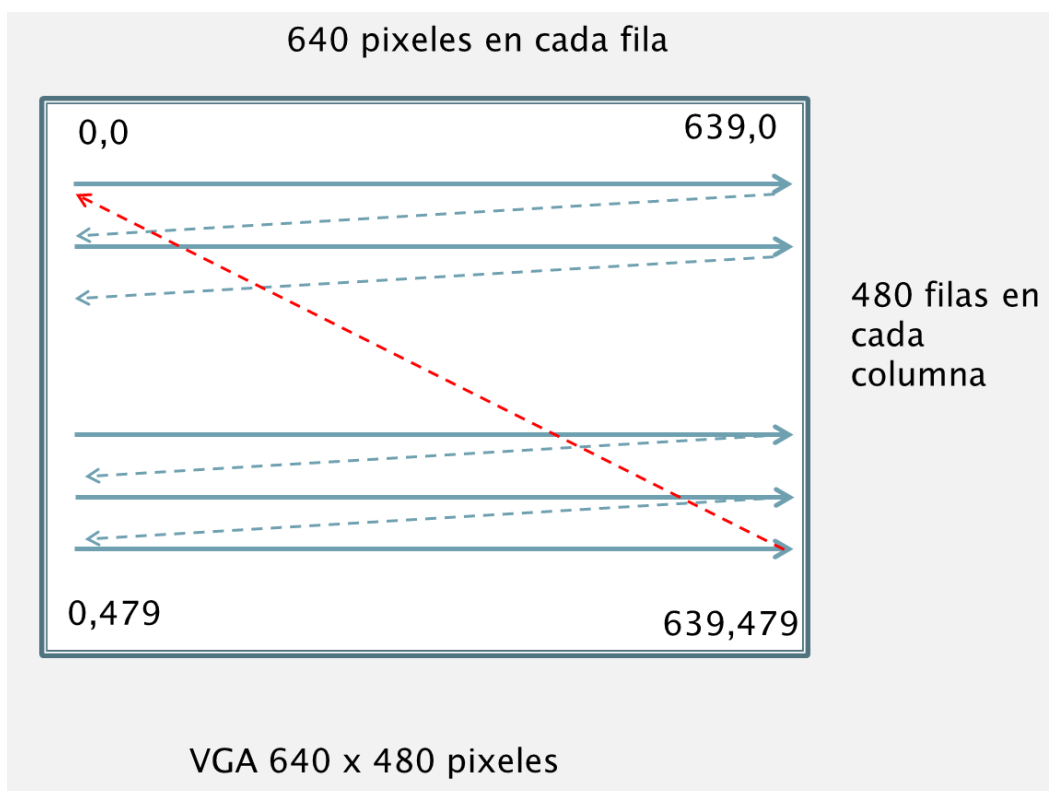


# Práctica 2

## Análisis, desarrollo y optimización de proyectos Implementación de un juego interactivo tipo Pong

*Basada en Capítulo 10 de Hamblen, J.O., Hall T.S., Furman, M.D.: Rapid Prototyping of Digital Systems : SOPC Edition, Springer 2008*

## Interfaz VGA



# Interfaz VGA

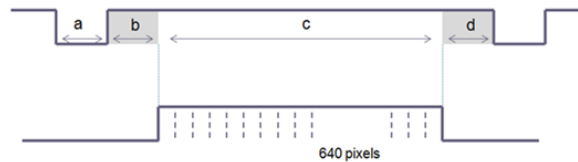
## Señales de sincronización

Formato 640 x 480 pixels (60 Hz)

Reloj de pixel: 25MHz

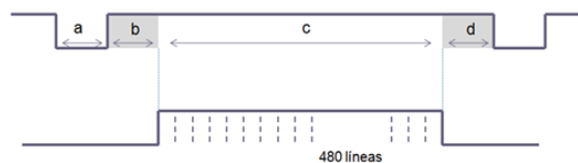
### Sincronización horizontal

Pulso (a): 3,77  $\mu$ s  
Back Porch (b): 1,89  $\mu$ s  
Front Porch (d): 0,94  $\mu$ s  
Vídeo on (c): 25,17  $\mu$ s



### Sincronización vertical

Pulso (a): 64  $\mu$ s  
Back Porch (b): 1,02 ms  
Front Porch (d): 0,35 ms  
Vídeo on (c): 15,24 ms



La frecuencia de reloj de pixel y los valores para los intervalos a, b, c y d determinan la resolución de pantalla y la frecuencia de refresco.

# Interfaz VGA

## Core “controlador\_640\_x\_480\_60” (I)

*Utilizado en la práctica 2. Temporización similar al FPGACore VGA\_SYNC de [1]*

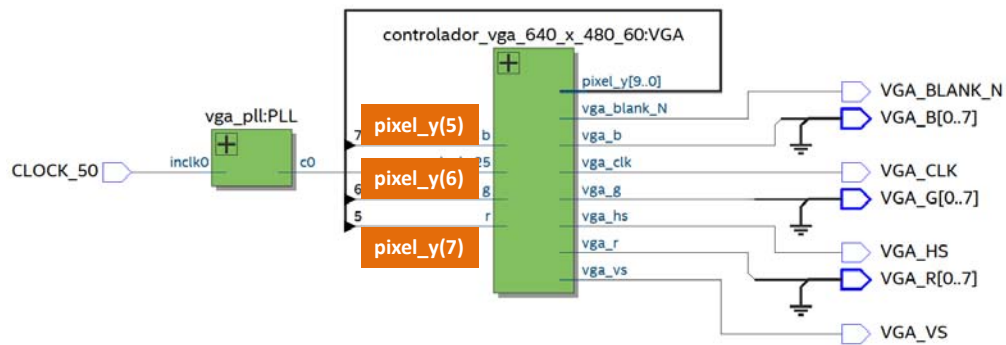
### Entradas

- `clock_25`: Entrada de reloj de 25 MHz
- `R`, `G` y `B`: Información de color para la señal de vídeo.

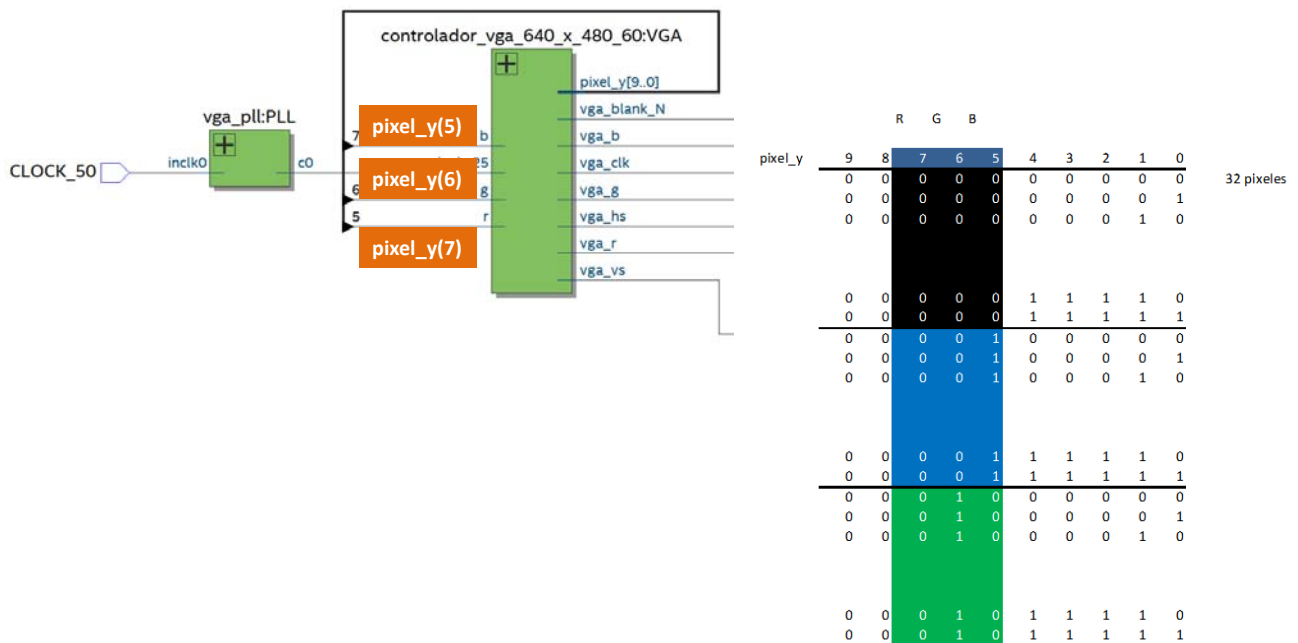
### Salidas

- `vga_hs` y `vga_vs` se conectan a las señales de sincronización horizontal y vertical del conector
- `vga_R`, `vga_G` y `vga_B` se conectan a un DAC externo que genera las señales analógicas RGB para el monitor. En la DE2-115 se dispone de 8 bits para cada componente de color. Con este controlador se emplean únicamente los bits más significativos de cada componente de color.
- `vga_clk` proporciona el reloj del pixel para el DAC.
- `vga_blank_N` indica si el pixel corresponde a área de visualización o no (tiempos de retroceso y sincronización).
- `Pixel_x` y `Pixel_y` proporcionan las coordenadas del pixel. Esta información es útil al usuario para generar las entradas de color.

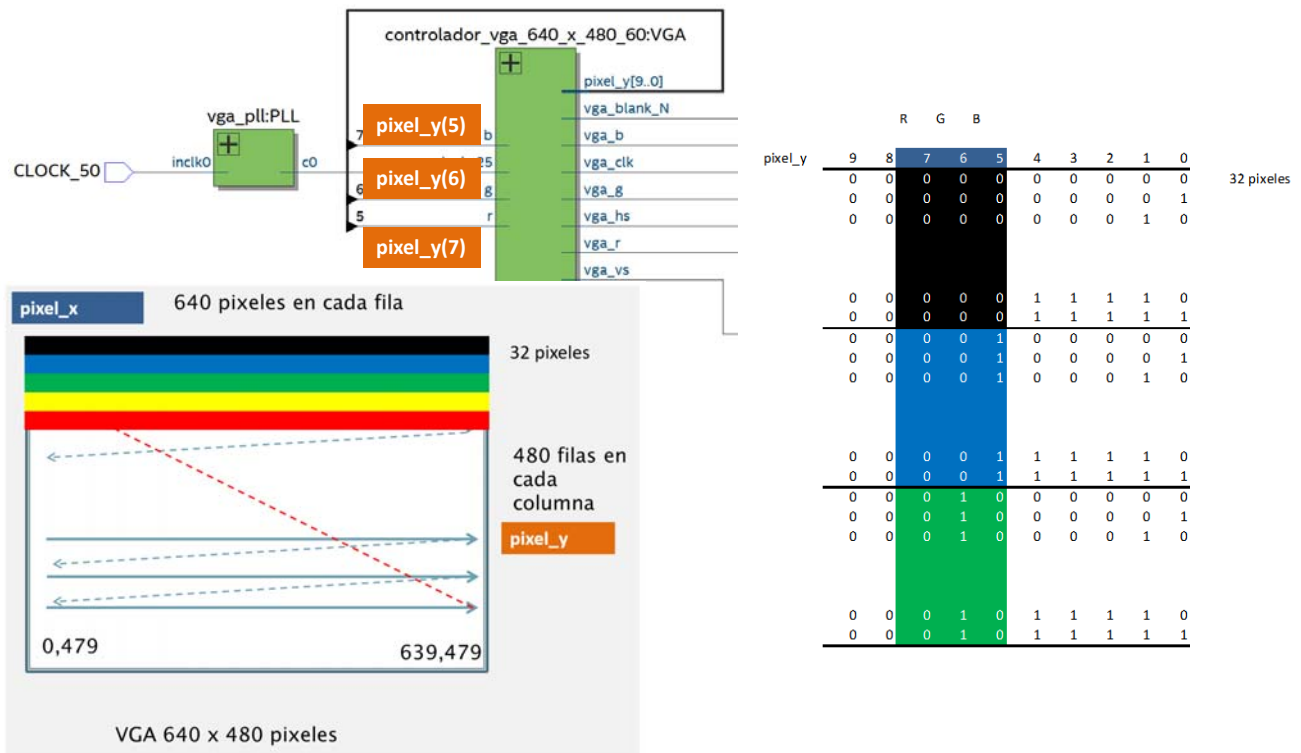
# ¿Qué se visualiza en pantalla?



# ¿Qué se visualiza en pantalla?

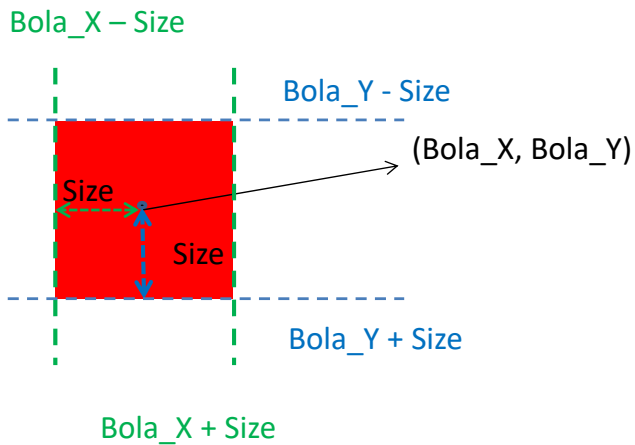


# ¿Qué se visualiza en pantalla?



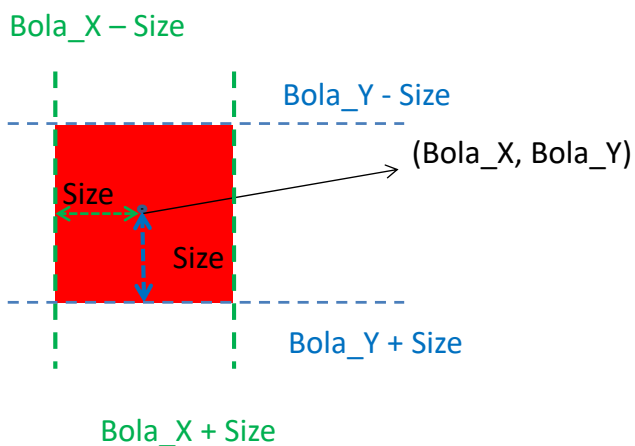
Analizar detenidamente la descripción `bola.vhd` y modificar la descripción `vga_top.vhd` para conectar el componente `bola.vhd` al controlador de VGA del siguiente modo: el componente `bola.vhd` recibe como entradas la señal de sincronización vertical (`vga_vs`) y las coordenadas del pixel a visualizar (`pixel_x`, `pixel_y`); y proporciona los valores de las componentes de color para dicho pixel (R, G, B).

# Dibujar\_bola



Si `Bola_Y - Size`  $\leq$  pixel\_Y  $\leq$  `Bola_Y + Size`  
y `Bola_X - Size`  $\leq$  pixel\_X  $\leq$  `Bola_X + Size`,  
entonces Bola\_on  $\leq$  '1' (bola)  
si no Bola\_on  $\leq$  '0' (fondo)

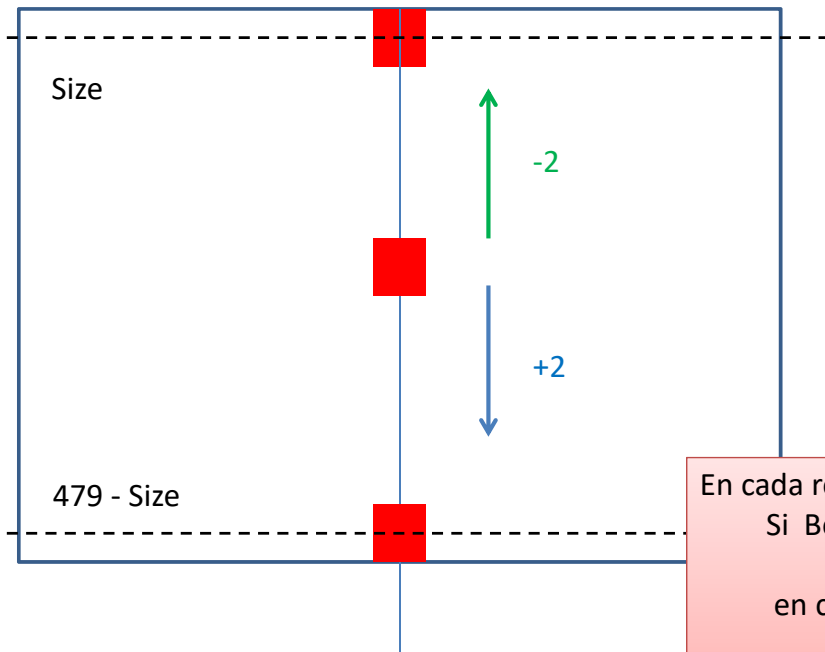
# Color de la bola



Si Bola\_on = '1', -- Blanco  
Red  $\leq$  '1'  
Green  $\leq$  '1'  
Blue  $\leq$  '1'

Si Bola\_on = '0' -- Fondo negro  
Red  $\leq$  '0'  
Green  $\leq$  '0'  
Blue  $\leq$  '0'

# Mover\_Bola



En cada refresco de pantalla (vs)  
 Si  $Bola\_Y \leq Size$   
     Desplaza  $Bola\_Y \leq +2$   
 en caso contrario, si  $Bola\_Y \geq 479 - Size$   
     Desplaza  $Bola\_Y \leq -2$   
 $Bola\_Y \leq Bola\_Y + Desplaza\_Bola\_Y$

Analizar detenidamente la descripción `bola.vhd` y modificar la descripción `vga_top.vhd` para conectar el componente `bola.vhd` al controlador de VGA del siguiente modo: el componente `bola.vhd` recibe como entradas la señal de sincronización vertical (`vga_vs`) y las coordenadas del pixel a visualizar (`pixel_x`, `pixel_y`); y proporciona los valores de las componentes de color para dicho pixel (`R`, `G`, `B`).

