

# Práctica 1

## Periféricos I

### Controlador de pantalla VGA

#### 1 Introducción

La señal VGA contiene 5 señales activas. Dos señales de sincronismo horizontal y vertical compatibles con los niveles lógicos TTL y activas a nivel bajo. Respectivamente indican el cambio de línea y de cuadro. Las restantes tres líneas RGB son líneas analógicas cuyos valores oscilan 0,3V (ausencia de señal) y 1,0 V (máxima intensidad de rojo, verde y azul respectivamente). De esta manera, combinando diversas intensidades de las componentes RGB se construyen los colores.

En un monitor tradicional de computador el tubo de rayos catódicos es el elemento de mayor tamaño. La pantalla contiene puntos de fósforo rojo, verde y azul que emiten luz cuando son bombardeados por un haz de electrones. La parte trasera contiene un cátodo que se calienta a una temperatura de unos 800°, de forma que emiten tres haces de electrones (uno por cada color) que inciden sobre la pantalla de fósforo. Delante del cátodo se sitúa la rejilla de control que actúa sobre la intensidad de los haces. Éstos deben orientarse para que lleguen a cualquier punto de la pantalla. Esta orientación se logra mediante planos de deflexión horizontal y vertical. Adicionalmente una rejilla denominada máscara de sombras garantiza que cada rayo incida sobre los puntos de fósforo de su color. El rápido refresco de la pantalla consigue engañar al ojo humano siempre que supere los 30Hz, produciendo la impresión de una imagen fija. Si adicionalmente se pretenden representar imágenes en movimiento es recomendable aumentar esta frecuencia hasta los 60Hz. En los monitores de computadores esta frecuencia de refresco no es inferior a 60Hz y muchas veces es superior.

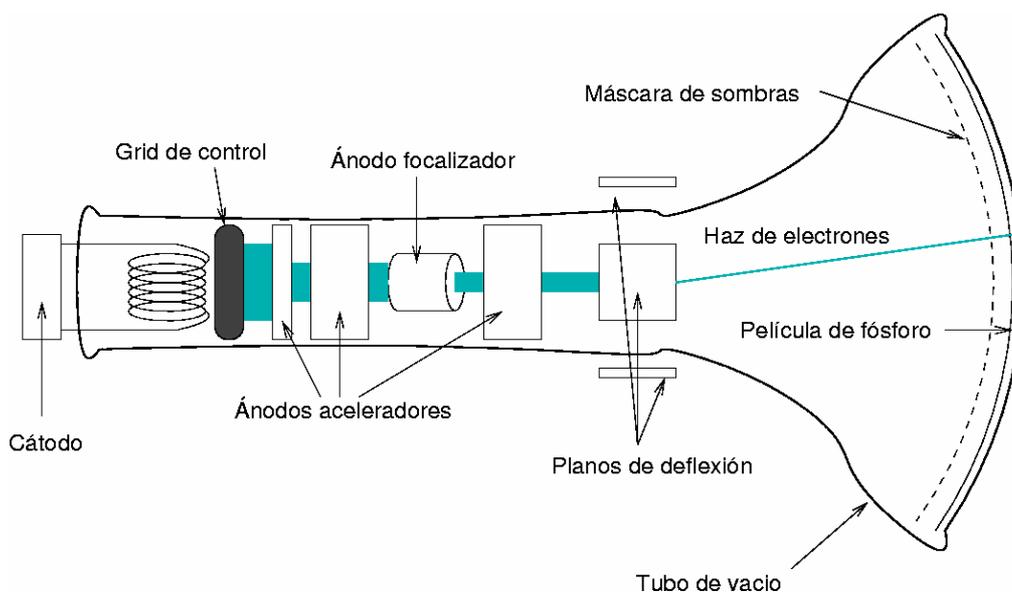


Figura 1: Corte transversal de un monitor de rayos catódicos

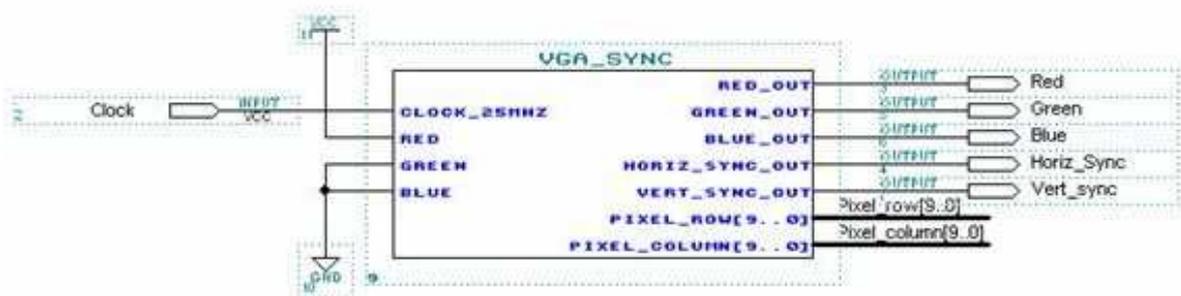
En una pantalla VGA la pantalla contiene 640x480 elementos pictóricos (*picture elements* = **píxel**). Los haces recorren la pantalla empezando por la esquina superior izquierda desde el punto de vista frontal de monitor. Para generar el cambio de línea y



El primer paso es bajar el fichero `sincronismo_vga.zip` de la página web de la asignatura, descomprimirlo en `c:\tmp`. A continuación se debe invocar la herramienta Max+plus II. Este programa se ejecuta bajo Windows (en el laboratorio Windows 2000). A continuación se debe abrir el diseño base ya preparado que incluye el bloque `VGA_SYNC`. Para ello en Max+plus II se escogerá

- *File > Open*

Aparecerá una ventana de diálogo donde se seleccionará *Graphic editor File*. Y se escogerá el fichero `sincronismo_vga.gdf`. Posteriormente se validará la ventana, con lo que aparecerá el editor gráfico con el diseño que se muestra en la figura 3.



**Figura 3:** Módulo generador del sincronismo VGA

A continuación se salvará el diseño con otro nombre individualizado. El nombre estará formado por un identificador de la práctica (PIAEC), un identificador del grupo de laboratorio (el número de grupo de laboratorio) y el número de la pareja en el laboratorio. Todo separado por guiones de subrayado. Por ejemplo: `PIAEC_G7_P10.gdf`

- *File > Save as...*

Una vez hecho esto se debe especificar que el diseño presente en la ventana activa va a ser el nivel más alto del proyecto. Para ello, y dejando la ventana del esquema activa se debe activar:

- *File>Project>Set Project to current file*

El nombre del fichero, con el camino completo, debe aparecer en la parte superior de la ventana junto al logotipo y nombre de la herramienta. Ahora ya se está en disposición de empezar a experimentar con el controlador VGA.

Una primera inspección del bloque ayudará a entender el funcionamiento del controlador. Para ello se debe pinchar dos veces en el bloque. Se abrirá una ventana de texto con el código VHDL del módulo. Es importante leer el código y entender el funcionamiento para poder implementar el resto de bloques que se piden en la sesión. El código tiene bastantes comentarios y sabiendo que el reloj es de 25MHz y observando la temporización de las señales VGA en la figura 2 se puede entender su significado.

Viendo la estructura del esquema es fácil adivinar cual será el comportamiento de la pantalla. En todos los puntos la señal roja estará activa y la verde y azul no, con lo que la pantalla será de un rojo uniforme. Para probar el diseño se debe compilar y programar

la FPGA. Por defecto el proyecto ya debe tener la asignación de dispositivo y de pines correcta. No está de más comprobarlo con:

- *Assign>Device*
- *Assign>Pin Location Chip*

A partir de aquí ya se puede abrir el compilador:

- *Max+plus II>Compiler*

Al ejecutarse START no deben aparecer errores, aunque si algún *warning* cuyo significado debe entenderse.

Una vez se ha compilado el diseño (se ha generado la lista de conexiones de la FPGA) debe programarse ésta utilizando el cable ByteBlaster(MV) que estará conectado al puerto paralelo del computador LPT1, y al conector especial de la tarjeta UP2. La FPGA es de tecnología SRAM con lo que no contendrá ninguna información de las sesiones anteriores si esta se ha desconectado. El programador se abrirá con:

- *Max+plus II>Programmer*

En el caso de que no detecte el conector con la ventana del programador activa se puede seleccionar:

- *Options>Hardware Setup*

Una vez la interfaz de programación sea correcta dentro de la ventana de programación bastará con pulsar *configure* para que se descargue la lista de conexiones.

Una vez programada la FPGA se podrá ver el funcionamiento del controlador VGA implementado sin más que conmutar la entrada al monitor con la salida VGA de la tarjeta. El resultado debe ser el descrito: toda la pantalla roja.

A continuación se va a introducir la utilización de una memoria ROM como generador de caracteres gráficos. El bloque ROM que se va a utilizar también está predefinido y se debe introducir en el esquema. Para ello se insertará el símbolo mediante dos pulsaciones del botón izquierdo y seleccionando el fichero CHAR\_ROM. Se propone montar un esquema como el que aparece en la figura 4. Con esta configuración se construye un pantalla en modo texto con 30 líneas x 40 columnas, con 64 caracteres definidos en la ROM en una matriz de 8x8 puntos. Sin embargo al no utilizar los bits menos significativos de las filas y las columnas en el direccionamiento se duplica el grosor de estos caracteres produciendo unos perfiles más grandes y claros.

Si se monta el esquema de la figura 4, se compila y programa la FPGA se mostrará un patrón de filas de caracteres. Es importante entender el funcionamiento del circuito para poder hacer las siguientes partes de la sesión. El fichero que describe el contenido de la ROM tiene extensión *.MIF* y un formato bastante intuitivo. Se pueden modificar los caracteres a voluntad sin más que editar este fichero.

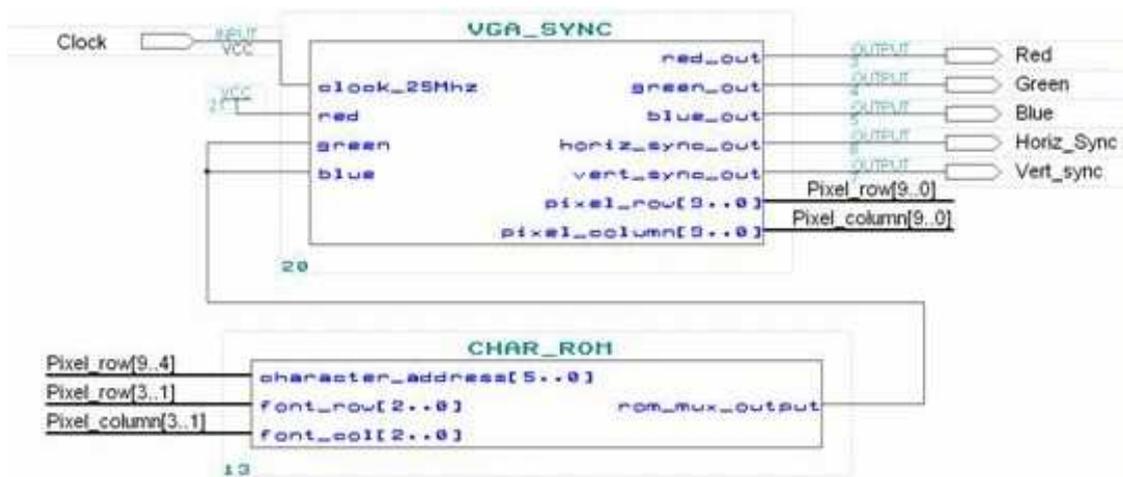


Figura 4: Esquema con la ROM de caracteres

A partir del diseño anterior es posible implementar un diseño algo más complejo que muestre caracteres predefinidos por la pantalla. De esta manera se propone:

- A) Implementar un diseño que muestre el nombre + el primer apellido de un miembro de la pareja en la zona central de la pantalla. Los caracteres se imprimirán en blanco sobre fondo negro y no debe aparecer ningún carácter adicional en la pantalla.
- B) En este caso el nombre se desplazará a lo largo de las filas de arriba abajo, a modo de salva-pantallas. Cambiando de fila una vez por segundo.

Una forma sencilla de realizar el apartado A (sin implementar una memoria de video) consiste en diseñar un bloque combinacional que en función de la fila y la columna genere o un blanco o un carácter de los que forman el nombre. El apartado B requiere un bloque que además de realizar la función del apartado A lleve la cuenta de los pulsos de reloj transcurridos para poder mover el nombre como se indica.

La funcionalidad de los bloques se realizará en VHDL. Para ello abrirá un editor de texto VHDL mediante

- *File>New*

Se seleccionará *Text Editor File* con la extensión `.vhd` (por defecto es `.t.d.f`). Una vez creado el bloque se generará un símbolo con:

- *File>Create default symbol*

Esta acción lanzará el compilador de la herramienta que verificará que la sintaxis es correcta y que no hay errores. Una vez generado el símbolo (que no aparecerá inicialmente en la pantalla) se incluirá de la misma manera que ya se ha descrito.