

## Práctica 3. Diseño de un controlador para VGA.

Antonio María Franqués García (77791084R), grupo A2A.

### ***Tarea 2***

---

En esta tarea se implementa un código capaz de mostrar en la pantalla de la FPGA un color uniforme del color especificado.

#### Dificultades encontradas durante la realización:

- El hecho de que haya sido la primera toma de contacto con las señales de sincronización para la correcta visualización de la pantalla ha requerido más tiempo inicialmente.

Comentarios sobre la simulación: Se observa que la pantalla de la FPGA se ilumina del color indicado en el código Verilog y que al pulsar el primero de los cuatro pulsadores se *resetea* la pantalla.

#### Autoevaluación:

- Estimo que la dedicación a esta tarea ha sido de 4 horas, incluyendo las de clase práctica. Aunque en la primera clase de esta práctica todo me parecía nuevo y me costó arrancar y entender cómo se pretendía hacer el código, considerando que había leído previamente las explicaciones de la memoria y que en el aula comprendí los esquemas realizados por el profesor me parece un tiempo adecuado.
- La dedicación a la realización de la tarea ha sido total ya que tanto esta como las demás las he realizado de forma individual.

### ***Tarea 3***

---

En esta tarea se implementa un código capaz de mostrar en la pantalla de la FPGA una serie de columnas de distintos colores.

#### Dificultades encontradas durante la realización:

- Prácticamente ninguna ya que ha sido relativamente fácil darse cuenta de que tan sólo había que condicionar las señales RGB a la fila y columna que nos encontráramos en cada instante.

Comentarios sobre la simulación: Se observa que la pantalla de la FPGA se ilumina en franjas de colores según se hayan indicado en el código Verilog y que al pulsar el primero de los cuatro pulsadores se *resetea* la pantalla.

#### Autoevaluación:

- Estimo que la dedicación a esta tarea ha sido de 10 minutos. Considerando que es una modificación de la tarea anterior y que no se pide ningún diseño en concreto me parece un tiempo adecuado; si hubiese querido representar alguna bandera más espectacular tan sólo requería de dedicarle más tiempo y un cálculo más detallado de la zona de filas y columnas a colorear.

### ***Tarea 4***

---

En esta tarea se implementa un código capaz de mostrar en la pantalla de la FPGA una matriz de letras (todas la misma pero con posibilidad de seleccionar la deseada) cargadas de la memoria.

#### Dificultades encontradas durante la realización:

- Comprender cómo hacer correctamente los accesos a memoria y en cada acceso recoger correctamente los 8 bits que indican qué parte de la pantalla se iluminará de un color y cuál de otro.
- Entender que reduciendo el número de accesos a memoria adecuadamente se consigue un tamaño de letra mayor.

Comentarios sobre la simulación: Se observa que la pantalla de la FPGA se llena de la letra que hayamos indicado en el código y que al pulsar el primero de los cuatro pulsadores se *resetea* la pantalla.

#### Autoevaluación:

- Estimo que la dedicación a esta tarea ha sido de 2 horas, incluyendo las de clase práctica. Considerando que había leído y comprendido previamente las explicaciones de la memoria y que en el aula comprendí el esquema realizado por el profesor me parece un tiempo adecuado.

## ***Tarea 5***

En esta tarea se implementa un código capaz de mostrar en la pantalla de la FPGA una única letra centrada tanto horizontal como verticalmente, cargada de la memoria.

#### Dificultades encontradas durante la realización:

- Fue fácil darme cuenta de que para mostrar sólo una de las letras y que esta estuviera centrada debía de condicionar las señales RGB a la zona de filas y columnas deseada. La parte ligeramente complicada fue a lápiz y papel calcular entre qué fila y columna debía de situarse dicha letra. Para conseguir eso conté el número de filas y columnas que requería la reproducción de cada una de las letras y también de cuál era el número inicial y final de cada fila y columna. Así pues, por ejemplo, la primera letra ocuparía las filas de la 0 a la 63 y las columnas de la 0 a la 63, la segunda letra ocuparía las filas de la 64 a la 127 y las columnas de la 0 a la 63...
- Me tomó un tiempo comprender concienzudamente qué pasaría si la señal que se utiliza para seleccionar la fila del carácter a mostrar (nº de posición de memoria) fuese la señal *columna* y la que se usa para recoger sólo uno de los 8 bits que componen cada una de las columnas del carácter fuese la señal *fila*; así pues, al cabo de un rato comprendí que esa era la forma de girar 90º a la izquierda la letra a mostrar.

Comentarios sobre la simulación: Se observa que centrada en la pantalla de la FPGA se representa la letra que hayamos indicado en el código y que al pulsar el primero de los cuatro pulsadores se *resetea* la pantalla.

#### Autoevaluación:

- Estimo que la dedicación a esta tarea ha sido de 1 hora y media. Considerando (ahora que sé cómo se hace) que es una ligera modificación de la tarea anterior me parece que dediqué un poco más del tiempo necesario dándole vueltas sobre cómo simplificar al máximo la modificación.

## ***Tarea 6***

En esta tarea se implementa un código capaz de mostrar en la pantalla de la FPGA una frase cargada de la memoria.

#### Dificultades encontradas durante la realización:

- Condicionar nuevamente las señales RGB a la nueva zona de filas y columnas donde queremos que aparezca la frase a mostrar.

- Utilizar correctamente las señales *fila* y *columna* para que la frase aparezca en la zona deseada y ordenar correctamente el archivo que será cargado a memoria con los caracteres a mostrar.

Comentarios sobre la simulación: Se observa que centrado tanto horizontal como verticalmente en la pantalla de la FPGA aparece mi nombre (ANTONIO M.) y que al pulsar el primero de los cuatro pulsadores se *resetea* la pantalla.

Autoevaluación:

- Estimo que la dedicación a esta tarea ha sido de 1 hora. Considerando que ya tenía el esquema hecho a mano sobre lo que ocupaba cada carácter y las zonas de aprovechamiento de la pantalla bien situadas creo que el tiempo fue adecuado.

## ***Tarea 7***

En esta tarea se implementa un código capaz de mostrar en la pantalla de la FPGA una única letra centrada tanto horizontal como verticalmente, seleccionada con el teclado y cargada de la memoria.

Dificultades encontradas durante la realización:

- Comprender cómo organizar las dos memorias ROM y sus archivos .mif para que a cada código hexadecimal que se asocia la pulsación de una tecla le corresponda la selección del carácter adecuado en el acceso a memoria que mostrará el carácter en pantalla.

Comentarios sobre la simulación: Se observa que centrada en la pantalla de la FPGA se representa la letra que hayamos seleccionado con el teclado y que al pulsar el primero de los cuatro pulsadores se *resetea* la pantalla.

Autoevaluación:

- Estimo que la dedicación a esta tarea ha sido de 1 hora y media. El tiempo empleado me ha parecido correcto, me hice una idea inicial rápida de cómo iba a configurar las dos memorias ROM así que desde el primer momento empecé directo a teclear el poco código que había que modificar de la tarea 5 y preparar el segundo archivo que se iba a cargar en la segunda memoria ROM.