

GENERACION DE IMÁGENES ALMACENADAS EN MEMORIA SOBRE MONITORES VGA

Objetivos

- Implementar sobre una FPGA un código VHDL para visualizar imágenes almacenadas en memoria sobre un monitor de video VGA.
- Interpretar el código implementado.

Materiales

Software Xilinx ISE 12.1 instalado
Tarjeta de desarrollo Digilent Spartan3E
Monitor VGA

Introducción

El objetivo de esta práctica de laboratorio es establecer una conexión entre un monitor de video y la tarjeta de desarrollo Spartan-3E usando su puerto VGA, para mostrar una imagen almacenada la memoria de la FPGA. Para una mejor comprensión del código implementado es importante que el estudiante revise cuidadosamente toda la documentación anexa a esta guía.

Procedimiento

1. Cree un nuevo proyecto llamado VGA_image con iguales características a los proyectos creados en la práctica anterior. Copie los archivos *clk_div_25MHz.vhdl*, *VGA_SYNC.vhdl*, *reader.vhdl* y *memory_coe.coe* en el directorio o carpeta del proyecto creado. Estos archivos los encontrara en la carpeta **ARCHIVOS** anexa a esta guía.
2. Cree dentro del proyecto un esquemático llamado VGA_image_tets y defina como top module. Después genere y adicione los símbolos los archivos *VGA_SYNC.vhdl* y *reader.vhdl*.
3. El archivo o bloque de memoria contiene la información de color de los pixeles de la imagen que se desea visualizar. Este archivo será descargado en la memoria interna de la FPGA. Para ello, sobre el archivo VGA_image_tets seleccione la opción **New Source**. Aparecerá el cuadro de dialogo **New Source Wizard**. Seleccione la opción **IP (Coregen & Arquitectura Wizard)** y en el campo **File name** digite el nombre de archivo **mymemory** como se muestra en la figura 1. Presione el botón **Next**.

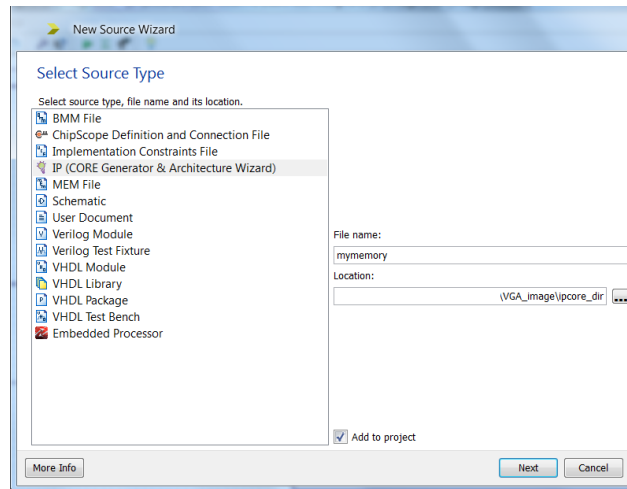


Figura 1. Ventana New Source Wizard

Seleccione la opción **Block Memory Generator 4.1** del submenú **RAMs & ROMs** dentro de la opción **Memories & Storage Elements**, tal como se muestra en la figura 2. Presione el botón **Next** y después el botón **Finish**.

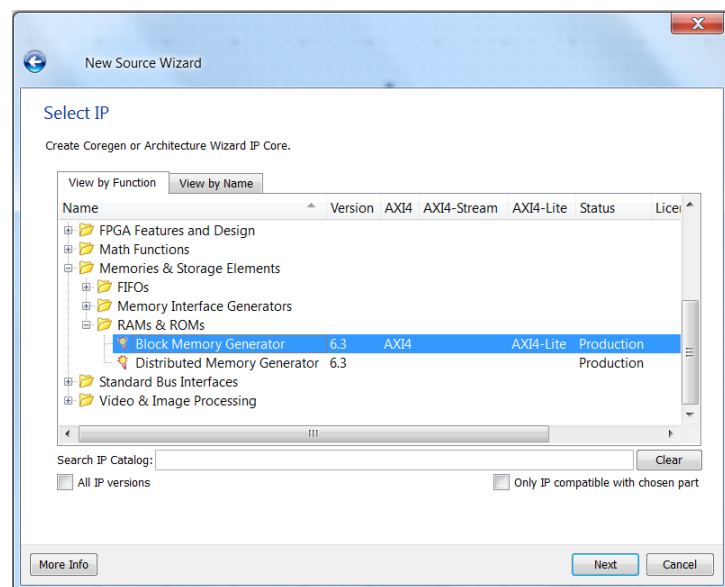


Figura 2. Ventana New Source Wizard – Select IP

En la ventana emergente (Block Memory Generator) de clic en NEXT y se podrán configurar las características de la memoria. Ya que no se escribirán datos en la memoria se selecciona la opción **Single Port ROM** en la sección **Memory Type** como se muestra en la figura 3, presione el botón **Next** para seguir con la configuración.

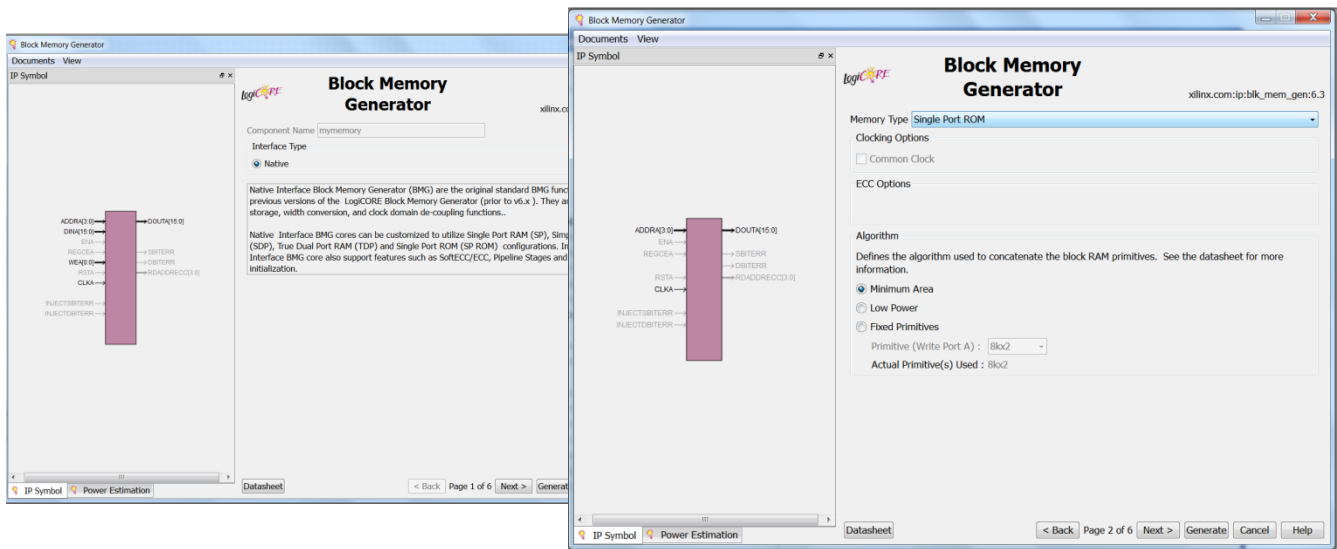


Figura 3. Ventana Block Memory Generator-Tipo de Memoria

La imagen que se desea almacenar tiene dimensiones de 224x224 pixeles y maneja solamente un bit de información por color. Por ello en el menú **Memory Size** la opción **Read Width** de la memoria se configura a **3** y la opción **Read depth** se configura a $224 \times 224 = 50176$ (que son las posiciones de memoria necesarias para almacenar la imagen) tal como se muestra en la figura 4.

Presione el botón **Next** para seguir con la configuración. Aparecerá una ventana como la mostrada en la figura 5. En esta ventana se cargará la memoria con la información de color de la imagen que se desea visualizar. Esta información se encuentra almacenada en el archivo **memory_coe.coe**, el cual se ha creado usando el script de Matlab **generar_imagen.m** anexo a esta guía. Este script permite generar un archivo, a partir de la imagen original (bart_simpson.bmp), en el formato que Xilinx soporta para sus memorias. Seleccione la opción **Load Init File** del menú **Memory Initialization** y presione el botón **Browse**. Seleccione el archivo **memory_coe.coe** y ábralo. Finalmente presione el botón **Generate** para crear el bloque de memoria. **Este proceso puede tomar algunos minutos.**

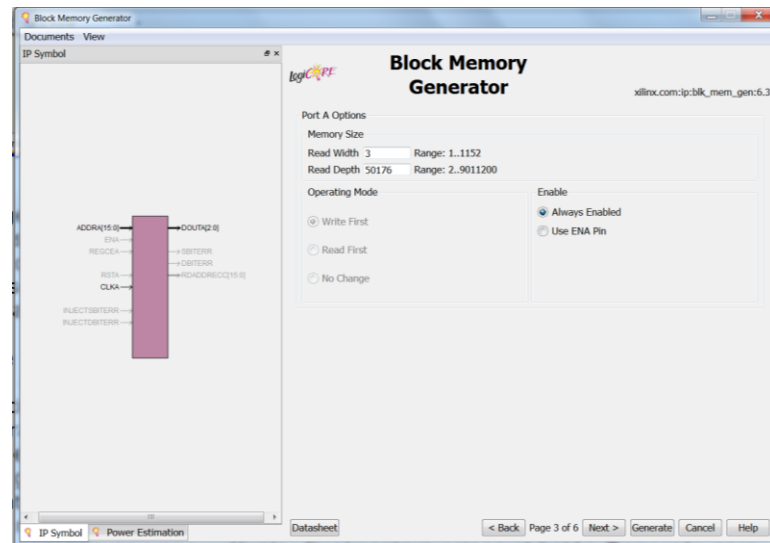


Figura 4. Configuración tamaño de Memoria

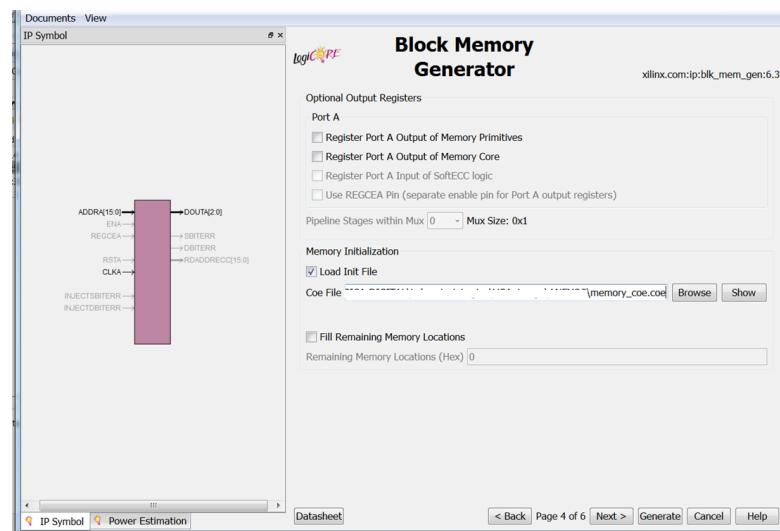


Figura 5. Inicialización de la Memoria

Una vez finalizado el proceso de generación se podrá observar el archivo de memoria adicionado al proyecto como se muestra en la figura 6.

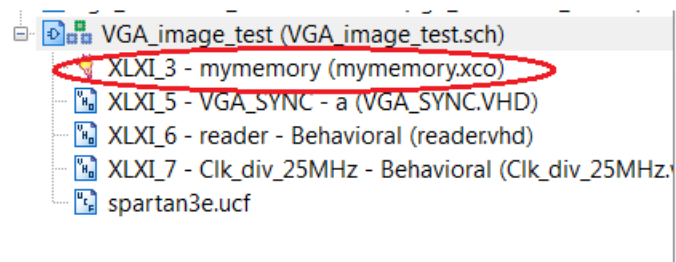
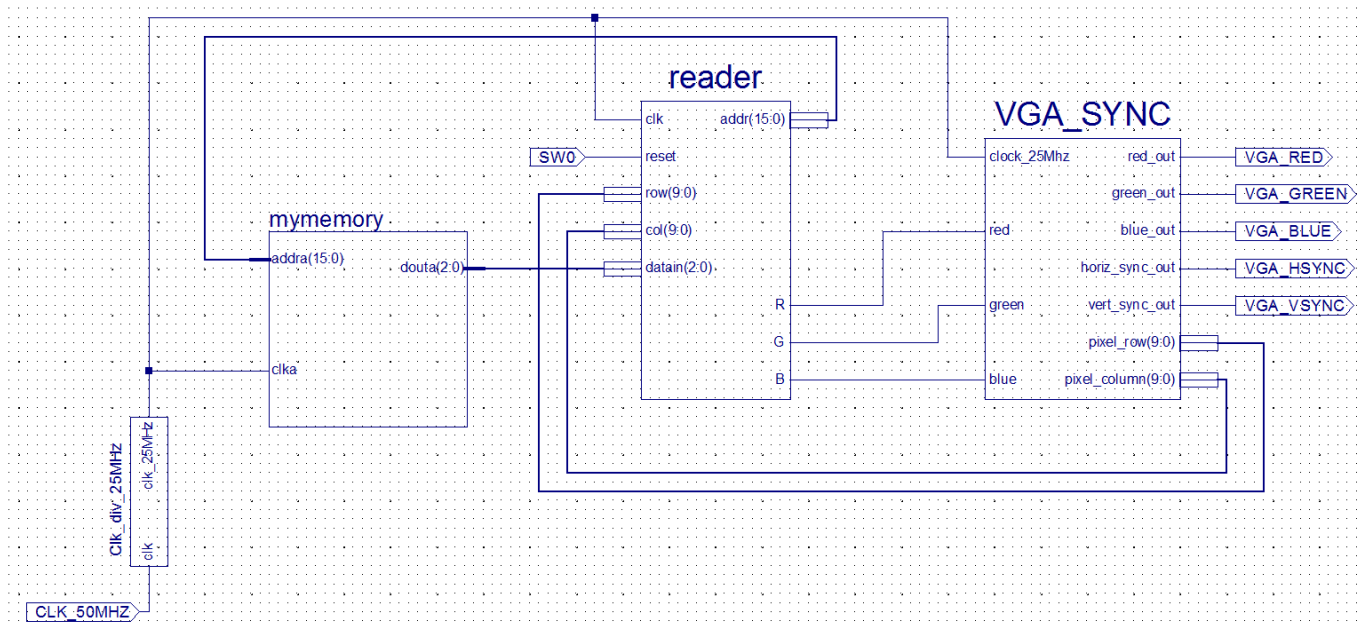


Figura 6. Bloque de memoria adicionado al proyecto

4. Sobre el esquemático, después de generar los símbolos, realice le conexionado como se muestra en la figura:



5. Sintetice el proyecto. Antes de poder programar el sistema de desarrollo recuerde modificar el archivo de restricciones como se muestra en la figura 7.

```

69 ## ==== Slide Switches (SW) ====
70 NET "SW0" LOC = "L13" | IOSTANDARD = LVTTTL | PULLUP;
71 #NET "SW1" LOC = "L14" | IOSTANDARD = LVTTTL | PULLUP;
72 #NET "SW2" LOC = "H18" | IOSTANDARD = LVTTTL | PULLUP;
73 #NET "SW3" LOC = "N17" | IOSTANDARD = LVTTTL | PULLUP;
74 #
75 ## ==== VGA Port (VGA) ====
76 NET "VGA_BLUE" LOC = "G15" | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST;
77 NET "VGA_GREEN" LOC = "H15" | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST;
78 NET "VGA_HSYNC" LOC = "F15" | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST;
79 NET "VGA_RED" LOC = "H14" | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST;
80 NET "VGA_VSYNC" LOC = "F14" | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST;
81

```

Figura 7. Archivo de restricciones

6. Descargué el programa en la FPGA del sistema de desarrollo y conecte el monitor, debe ver sobre la pantalla la imagen almacenada en memoria, que diferencias nota entre la imagen original y la decodificada y mostrada en el monitor?



Referencias

ISE In-Deph Tutorial en: www.xilinx.com

Manual usuario Tarjeta de desarrollo Spartan 3E en: www.digilentinc.com