DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

FPGA: Mode7

Fecha de entrega: 10 de julio de 2015 FPGA

Integrante	LU	Correo electrónico
Ignacio Gleria	387/10	igleria@dc.uba.ar
Federico Beuter	827/13	federicobeuter@gmail.com
Juan Rinaudo	864/13	jangamesdev@hotmail.com

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

http://www.fcen.uba.ar

Índice

1. Introducción

El presente trabajo tiene como objetivo implementar en una fpga un modo gráfico llamado **Mode 7**. Este modo gráfico tiene sus orígenes en la consola **SNES** (Super Nintendo Entertainment System), cuya aparición fue en el año 1990. Este modo gráfico permite realizar transformaciones de rotación (sobre un punto-origen), traslación y escalado a una imagen, logrando distintos efectos. Para ello se valía de un circuito en hardware especializado, que permitía lograr los cálculos en la velocidad en tiempo real para renderizar imágenes de una manera que no se había conseguido hasta el momento en consolas hogareñas. Algunos videojuegos que implementan Mode 7 son Super Castlevania 4 y Super Mario World.

2. Desarrollo

2.1. Teoría

Para lograr los efectos necesarios, las coordenas de la imagen original deben ser modificadas mediante una transformación lineal seguida de una traslación (lo que comúnmente se conoce como *transformación afín*).

La transformación utilizada puede verse en la figura **??**. Los valores a, b, c, d son los coeficiententes de transformación, (x, y) son las coordenadas originales, x_0 e y_0 son el offset y x', y' son las coordenadas resultantes.

Figura 1: Transformación utilizada

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & x_0 \\ a & b & y_0 \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \\ 1 \end{bmatrix}$$

En nuestro caso, los coeficientes de transformación deben ser configurados para que la ecuación quede como en la figura 2

Figura 2: Transformación con los coeficientes

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & x_0 \\ -\sin\alpha & \cos\alpha & y_0 \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \\ 1 \end{bmatrix}$$

2.2. Detalles de la implementación

2.2.1. Utilización de memoria

Uno de los primeros problemas a solucionar fue cómo calcular los valores de las funciones seno y coseno, ya que la fpga no posee una fpu. La solución consistió en guardar en dos tablas 360 valores

correspondientes a parámetros angulares para dichas funciones (de 0 a 359 grados). Cada resultado se guardó en 16 bits en forma signo más magnitud, con 7 bits para la parte entera y 8 para la decimal, lo que en total ocupó 720 bytes de memoria. Es necesario agregar que las operaciones matemáticas se realizaron con 24 bits signo más magnitud (15 bits parte entera, 8 parte decimal) por la resolución que utiliza la salida VGA.

La imagen a cargar, por otro lado, es de 64 por 64 pixeles, y cada pixel tiene tres bits para el canal rojo, tres bits para el verde y dos para el azul. Esto resulta en un bitmap que tiene 4096 valores de 8 bits y ocupa en memoria 4096 bytes. En la figura 3 se puede observar el bitmap utilizado.

Figura 3: bitmap utilizado

2.2.2. Paralelismo

En la implementación de las operaciones matemáticas necesarias para resolver el problema consideramos oportuno utilizar lógica combinacional para mayor velocidad. De haber usado máquina de estados, la velocidad de ejecución habría sido bastante menor.

2.2.3. Salida a video

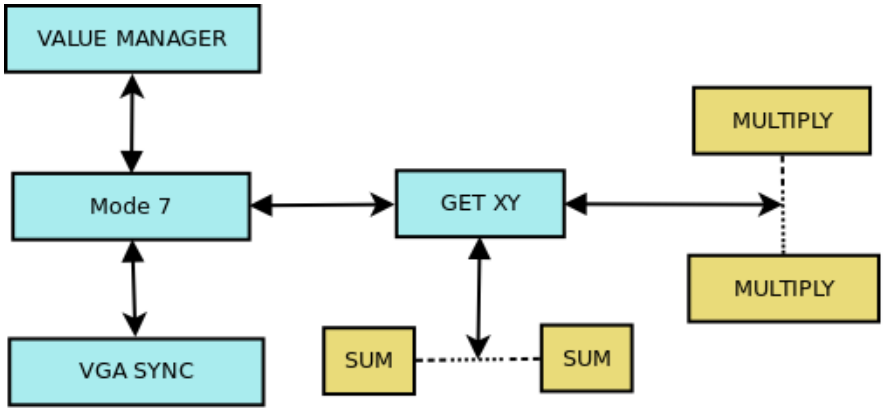
La implementación de la salida a video está basada en la explicación brindada en el libro **Verilog by Example: A Concise Introduction for FPGA Design**

2.3. Arquitectura

Podemos ver un esquema general en la figura 4

- **Mode 7** es el módulo central de la arquitectura, es decir, el encargado de hacer pedidos a los otros módulos y trasladar los resultados entre ellos.
- El módulo **Value Manager** es el encargado de interpretar las entradas de los switches y botones, que sirven para cambiar entre las distintas transformaciones de la imagen.
- El módulo **Get XY** es el encargado de hacer operaciones sobre los pixeles, utilizando varias encarnaciones de los módulos **SUM** y **MULTIPLY**. Allí también se realiza el cargado de las tablas de seno, coseno y la imagen.

Figura 4: Arquitectura general



- Finalmente, el módulo **VGA SYNC** es el encargado de la salida al monitor.

2.4. Utilización

La fpga, una vez programada y enchufada a un monitor, puede ya utilizarse. El mappeo de los controles es el siguiente:

- Switches 3 a 0: Son los que configuran los distintos modos de manipulación del bitmap.
- Switches 7 a 6: Son los que configuran la parte entera de la velocidad de cambio de cada modo. Por ejemplo, al mover horizontalmente el bitmap, este número regula que tan rápido se mueve.
- Switches 5 a 4: controlan la parte decimal del punto anterior.
- Botones arriba/abajo: Modifican los valores de cada modo. En sentido práctico, si estamos en el modo vertical, estos harán que se mueva el bitmap hacia arriba o abajo.
- Botón central: Resetea todos los valores y el circuito de la fpga.

Los distintos modos configurables por los switches 3 a 0 son:

- **0000**: Modo en el cual el bitmap se moverá horizontalmente.
- **0001**: Modo en el cual el bitmap se moverá verticalmente.
- **0010**: Modo en el cual el punto de origen del mundo se moverá horizontalmente.
- **0011**: Modo en el cual el punto de origen del mundo se moverá verticalmente.
- **0100**: Modo en el cual la textura del bitmap se multiplicará horizontalmente.
- **0101**: Modo en el cual la textura del bitmap se multiplicará verticalmente.
- **0110**: Modo en el cual el bitmap escalará horizontalmente.
- **0111**: Modo en el cual el bitmap escalará verticalmente.
- **1000**: Modo en el cual el bitmap rotará en relación al punto relativo de giro.

3. Conclusiones

Se logró implementar satisfactoriamente Mode 7 en una fpga, teniendo en cuenta las limitaciones de recursos del modelo disponible (spartan 6). Un posible trabajo a futuro implicaría lograr la aplicación de Mode 7 a una imagen animada, o utilizar la implementación existente en un contexto interactivo como un videojuego.