# Contenido

[Contenido i](#_Toc477258828)

[Objetivos 1](#_Toc477258829)

[Proyecto propuesto 1](#_Toc477258830)

[Desarrollo 1](#_Toc477258831)

[Componentes 2](#_Toc477258832)

[Driver VGA 2](#_Toc477258833)

[DesplazaXYAlCentro 3](#_Toc477258834)

[Rotar 3](#_Toc477258835)

[Control Giro 3](#_Toc477258836)

[Jugador 3](#_Toc477258837)

[Generador Aleatorio 3](#_Toc477258838)

[Dificultad 3](#_Toc477258839)

[Circulo 3](#_Toc477258840)

[Juego\_FSM 4](#_Toc477258841)

[Dibujo 4](#_Toc477258842)

[Conv\_hexagonal 4](#_Toc477258843)

[GeneradorParedes y Paredes 5](#_Toc477258844)

[Bluetooth 6](#_Toc477258845)

[Problemas encontrados 7](#_Toc477258846)

[Problema 1 7](#_Toc477258847)

[Problema 2 7](#_Toc477258848)

[Problema 3 8](#_Toc477258849)

[Problema 4 8](#_Toc477258850)

[Mejoras propuestas 8](#_Toc477258851)

[Apéndice A: Diagrama del sistema iv](#_Toc477258852)

[Apéndice B vi](#_Toc477258853)

[Apéndice C viii](#_Toc477258854)

[Apéndice D viii](#_Toc477258855)

[Apéndice E x](#_Toc477258856)

[Apéndice F x](#_Toc477258857)

[Apéndice G xv](#_Toc477258858)

[Apéndice H xxxiii](#_Toc477258859)

[Apéndice I xxxiv](#_Toc477258860)

[Apéndice J xxxix](#_Toc477258861)

[Apéndice K xli](#_Toc477258862)

[Archivo L xlii](#_Toc477258863)

[Apéndice M xliv](#_Toc477258864)

[Apéndice N xlv](#_Toc477258865)

[Apéndice Ñ li](#_Toc477258866)

[Apéndice O li](#_Toc477258867)

[Apéndice P lii](#_Toc477258868)

[Apéndice Q liii](#_Toc477258869)

[Apéndice R lv](#_Toc477258870)

[Apéndice S lv](#_Toc477258871)

[Apéndice T lvi](#_Toc477258872)

[Apéndice U lviii](#_Toc477258873)

[Apéndice V lx](#_Toc477258874)

[Apéndice W lxii](#_Toc477258875)

[Apéndice X lxiii](#_Toc477258876)

[Apéndice Y lxiii](#_Toc477258877)

[Apéndice Z lxiv](#_Toc477258878)

[Presupuesto lxvi](#_Toc477258879)

[Figura 1 2](#_Toc477258471)

[Figura 2 2](#_Toc477258472)

[Figura 3 4](#_Toc477258473)

[Figura 4 5](#_Toc477258474)

[Figura 5 5](#_Toc477258475)

[Figura 6 6](#_Toc477258476)

[Figura 7 7](#_Toc477258477)

[Figura 8 i](#_Toc477258478)

[Figura 9 ii](#_Toc477258479)

# Objetivos

* Familiarizarse con los conceptos del protocolo VGA.
* Estudio e implementación del procesamiento de datos necesario para la generación de una imagen.
* Familiarizarse con un entorno de desarrollo para producir una aplicación para Android que haga uso de datos provenientes de sensores de un Smartphone.
* Establecimiento de una enlace de comunicación con el uso de un módulo Bluetooth.
* Realizar un acercamiento al campo de desarrollo de videojuegos.
* Utilización de las instrucciones secuenciales, concurrentes y paquetes aprendidos en clase.
* Interpretación de información especificada en hojas de datos o especificaciones de diseño.
* Familiarizarse con el DE2-115 Cyclone IV development kit board.
* Uso de un constraint file para asignación de pines de I/O, pulsadores, switches y LEDs.
* Uso de la herramienta MegaWizard para la implementación de un DLL o PLL para generar un reloj.
* Uso de la herramienta SignalTrap para depuración y análisis del sistema implementado.

# Proyecto propuesto

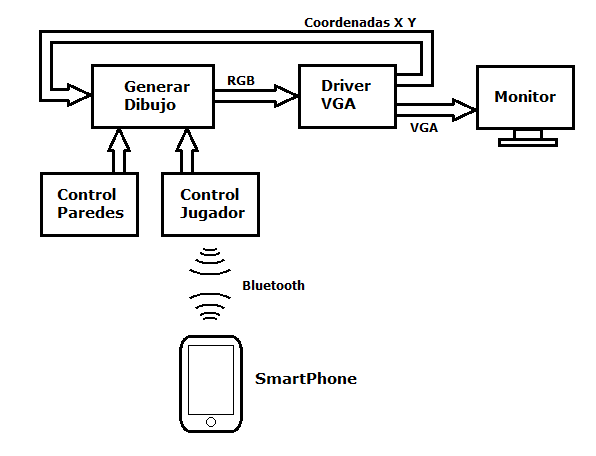
Se pretende implementar un juego que deberá ser visualizado en un monitor, conectado directamente a la placa de desarrollo DE2-115, y deberá ser controlado inalámbricamente por sensores presentes en un Smartphone comercial.

El objetivo del juego consiste en que el jugador guie un puntero, el cual sólo puede desplazarse en una trayectoria circular, alrededor del hexágono central. A este convergen diversos obstáculos que intentarán aplastar al jugador y que éste deberá esquivar. Para ello, deberá cambiar la posición de un Smartphone el cual deberá ejecutar una aplicación que se comunicará con un módulo Bluetooth conectado a la placa de desarrollo.

A su vez, para generar un aumento en la dificultad, el entorno del juego presenta distracciones al jugador; como por ejemplo, rotación de la pantalla y cambios de color en los elementos visualizados.

# Desarrollo

En la Figura 1 se observa un diagrama de bloques conceptuales del funcionamiento del sistema.



Figura

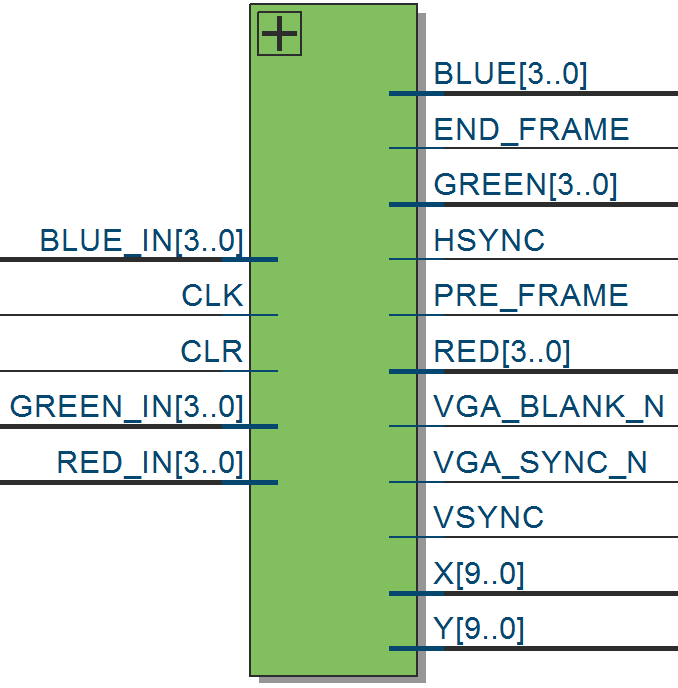
El Driver VGA realiza un barrido de las coordenadas de la pantalla y para cada una de ellas, el bloque Generar Dibujo busca coincidencias con las posiciones de las paredes y el jugador, que son controlados por el bloque Control de Paredes y Control de Jugador respectivamente. Este último a su vez, se comunica con el Smartphone vía Bluetooth para cambiar la posición del puntero.

Cuando en la coordenada existe un objeto, el bloque Generar Dibujo envía, a través del Driver VGA, el Color del pixel que debe ser dibujado en el monitor.

## Componentes

En Apéndice A: Diagrama del sistema se muestra el diagrama completo de la conexión de los componentes desarrollados. A continuación se detalla la función de cada uno.

### Driver VGA



Figura

En la Figura 2 se observan las entradas y salidas del Driver VGA. Este módulo se encarga de enviar los bits que definen el color a mostrar en cada pixel y las señales de control necesarias al chip ADV7123, presente en el kit DE2-115, encargándose de que se respeten los requerimientos de tiempo; para ello requiere un reloj de entrada de 25MHz. A su vez, envía señales de señalización a otros módulos dentro del sistema como las coordenadas del pixel que está siendo procesado y señalización de comienzo y fin de frame.

El driver fue configurado para el standard 480x640 @ 60Hz. Aunque la resolución es la dicha, los contadores internos horizontales y verticales verdaderamente cuentan hasta 800 por línea y 525 por frame respectivamente. Esto se debe a que en ellos, se incluyen no solo el área visible sino el pulso de sincronización, el front-porch y el back-porch.

### DesplazaXYAlCentro

Este bloque realiza una traslación del sistema de coordenadas generado por el driver VGA al centro de lo pantalla. Para ello es necesario realizar una conversión de dato de entero sin signo a entero con signo.

### Rotar

Este componente se encarga de cambiar el sistema de coordenadas por una versión rotada del mismo. Para esto, tiene por entradas las coordenadas X e Y, y el ángulo α; y por salidas las coordenadas X e Y en el nuevo sistema.

### Control Giro

Este bloque tiene 3 funciones; La primera es entregar el ángulo para que el bloque Rotar rote la imagen; La segunda es aumentar la velocidad del giro al recibir la señal de overflow de un contador; mientras que la tercera es cambiar el sentido de giro cada cierto tiempo. Estas últimas cumplen el objetivo de aumentar la dificultad del juego.

### Jugador

Este bloque se encarga del procesamiento de la posición del jugador y sus desplazamientos en base a lo recibido por bluetooth, o en base a la presión de los pulsadores de la placa. Teniendo en cuenta la dificultad configurada, se cambia la velocidad de desplazamiento.

### Generador Aleatorio

Es un contador LFSR de 64 bits. Para evitar correspondencias entre los diferentes datos aleatorios necesitados en el sistema, se utilizan distintos bits del contador para distintos módulos.

### Dificultad

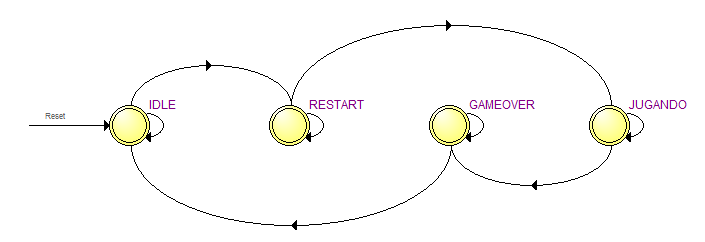
Este bloque obtiene y envía la información de la dificultad configurada a través de los switches de la placa. También da retroalimentación visual para el jugador mostrando el nivel de dificultad en los LEDs presentes en el kit DE2-115.

### Circulo

Este componente se encarga de la visualización del puntero del jugador. Su funcionamiento se basa en comparar cada uno de los pixeles de la pantalla con la posición del jugador y si la distancia entre estos es menor a un valor determinado, se activa una señal de salida que es enviada al bloque dibujo.

### Juego\_FSM

Este componente implementa la máquina de estados finita mostrada en la Figura 3. Esta controla las distintas etapas del juego.



Figura

La MEF cuenta con 4 estados:

* Idle: estado presente cuando el sistema se encuentra a la espera de que se inicie un nuevo juego
* Restart: ocurre cuando se inicia una nueva partida y envía una señal de reset a los demás componentes
* Jugando: sucede cuando el usuario se encuentra jugando
* Gameover: activo cuando se produce una colisión.

Las señales de salida de este componente define el comportamiento de los demás bloques

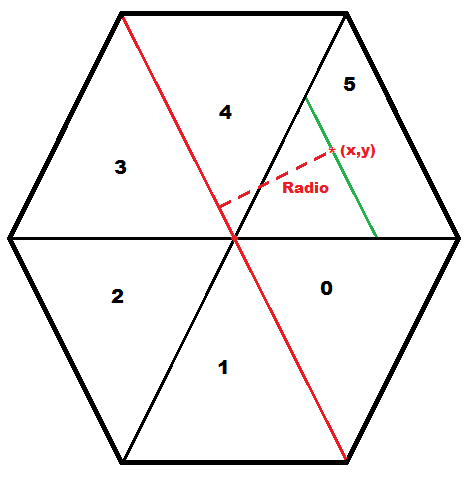
### Dibujo

Este componente se encarga de indicarle al driver VGA el color del pixel analizando si éste corresponde a una pared, jugador, isla o vacío. A su vez, también se ocupa de alternar los colores de fondo y la luminosidad de cada cuadrante.

### Conv\_hexagonal

El espacio de la pantalla se divide en 6 áreas denominadas cuadrantes. Estas pueden observarse en la Figura 4. Como las paredes y los colores de fondo son idénticos para el mismo cuadrante, se utilizaran coordenadas hexagonales, compuestas por el cuadrante y el radio.

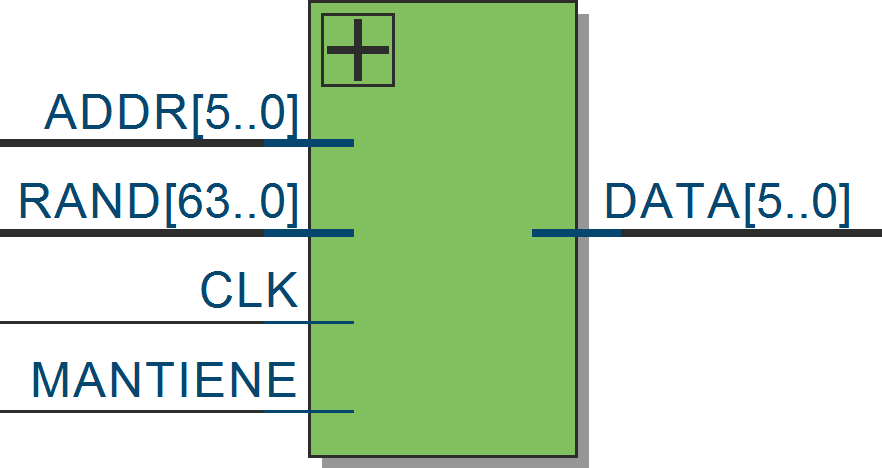
El componente conv\_hexagonal transforma las coordenadas cartesianas en hexagonales. El cuadrante lo define según en qué área esta la coordenada analizada. Mientras que el radio es definido como la distancia del punto a una línea perpendicular al cuadrante. De esta forma, todos los puntos pertenecientes a la línea verde, son procesados como si fueran el mismo.



Figura

### GeneradorParedes y Paredes

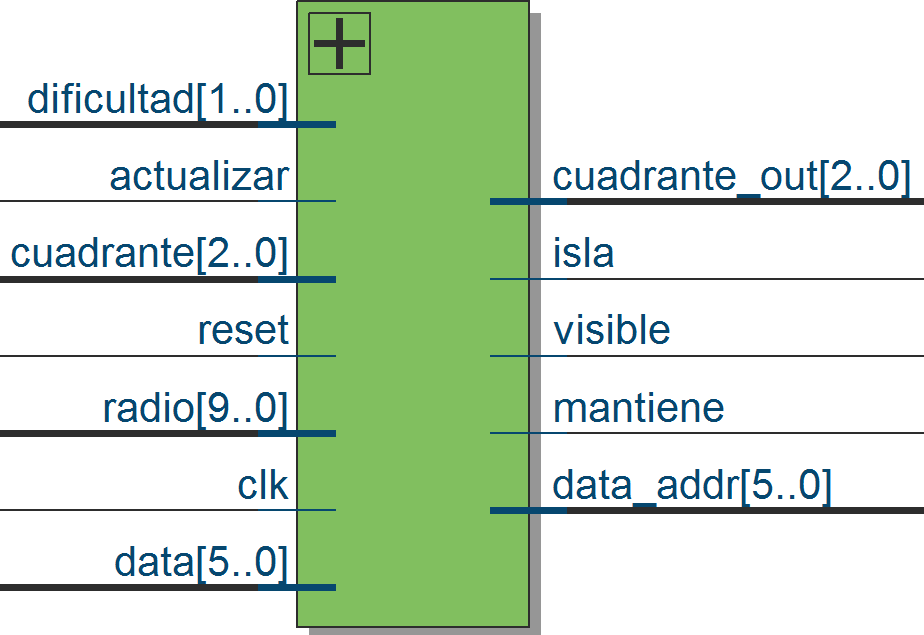
Las paredes están definidas en memorias de solo lectura en el componente GENERADOR\_PAREDES mostrado en la Figura 5. Este bloque funciona como una memoria en la cual los 2 bits más significativos de la dirección definen el tipo de patrón al que se va a acceder, los siguientes 3 bits son definidos aleatoriamente y cambian cuando la señal mantiene tiene un ‘0’ lógico, y los 4 bits menos significativos seleccionan a que dato de cada patrón se va a acceder.



Figura

El componente paredes, mostrado en la Figura 6, es el encargado de decidir cuándo, para la coordenada analizada, existe o no una pared. Para esto accede a la memoria definida en GENERADOR\_PAREDES y carga en un arreglo, denominado paredes\_a, los patrones a dibujar.

La señal centro es un puntero que barre al arreglo paredes\_a a la velocidad indicada por la señal dificultad. La dirección del arreglo a la que se acede para saber si debe dibujarse o no una pared, es indicada por la señal offset, la cual se calcula como la suma de centro y el radio de la coordenada.



Figura

Además, este componente dibuja un hexágono en el centro de la pantalla. Para esto, activa la señal paredes y la señal isla siempre que el radio es menor a cierto valor.

## Bluetooth

Para la transmisión de la información sobre la posición del Smartphone se creó una aplicación para Android utilizando la herramienta de MIT, App Inventor. En esta se envía una trama de 1 byte en la cual solo se utilizan los 3 bits menos significativos, 1 para cada posición de giro y el tercero para iniciar un nuevo juego.

Además se optó por incluir una lógica aparte para encender el bluetooth del Smartphone cuando este lo tiene apagado ya que la aplicación presentaba error cuando esto sucedía. Una vez arreglado esto, se agregaron los componentes necesarios para que cualquier Smartphone se pueda conectar desde la aplicación y no sea necesario cambiar la aplicación cada vez que se cambie de Smartphone.

Como última medida, se decidió incluir un slider que modifica el umbral en el cual la posición del celular es considerada inclinada hacia un lado o el otro. Esto se hizo con el fin de que cada usuario lo modifique a su gusto.

En la Figura 7 se observa la interfaz resultante de la aplicación de Android.



Figura

# Problemas encontrados

## Problema 1

En los primeros prototipos del sistema, se observó que la velocidad de rotación aumentaba drásticamente pasado el primer intervalo estipulado para el cambio.

En principio se pensó que el origen del problema provenía del bloque Control\_giro pero no se encontró ningún error.

Haciendo un testeo de la función de cambio de luminosidad del bloque Dibujo se comprobó que tenía un error de naturaleza similar al nombrado, y se dedujo que el factor común entre ambos problemas era el bloque contador que producía la señal que generaba el evento.

Revisándose el código de dicho componente, se encontró el error en su arquitectura que no le permitía auto resetearse una vez llegado alcanzado el overflow. Cuando el error fue enmendado, los cambios de velocidad, las variaciones de luminosidad entre cuadrantes y los cambios de color periódicos funcionaron correctamente.

## Problema 2

Se tuvo problemas en el procesamiento de los gráficos lo que generaba, en principio, diferentes grosores de paredes entre los diferentes cuadrantes; y luego, puntos en el mapa que eran procesados como pared cuando debían ser un vacío, y viceversa.

Se encontró que el problema era generado por el cálculo incorrecto del radio para cada cuadrante implementado en el componente conv\_hexagonal. Se cambió el código para resolver los errores, exitosamente.

## Problema 3

Al implementar el comando del jugador mediante bluetooth se detectó un delay que sin ser exageradamente alto, aumentaba la complejidad para realizar un control preciso del puntero. Como primera posible solución se optimizó la decodificación de la trama, utilizando los bits enviados directamente, con lo que se eliminó el posible retardo generado, sin embargo, esto no solucionó el problema.

Luego se decidió por aumentar todo lo posible la velocidad de trasmisión serie entre el módulo receptor y el FPGA, esto mejoró ligeramente la respuesta del sistema pero no se consideró aún aceptable.

Por último, se dedujo que el problema podía ser producido por la “app” generada para la comunicación desde la aplicación móvil. Dado el desconocimiento del grupo frente al desarrollo de aplicaciones para Android, se propuso como solución la generación del software sin el uso del entorno AppInventor para dar una trasmisión de datos optimizada, pero no se pudo llevar a cabo planteándose como mejora para trabajos futuros.

## Problema 4

Dado que las salidas del sistema en su totalidad son señales de video resultó difícil hacer la depuración del sistema debido a que las mismas son difícilmente interpretables en simulación. Esto generó que todas las pruebas debieran realizarse en línea.

Esto generó que se tuviese menos información a la hora de corregir los errores y mayores retardos de testeo debido a los mayores tiempos de compilación.

# Mejoras propuestas

Como se mencionó, se propone el desarrollo de la app utilizada por el dispositivo móvil sin el uso de AppInventor con el fin de optimizar el envío de datos hacia el módulo y conseguir disminuir el retardo de comunicación a niveles aceptables.

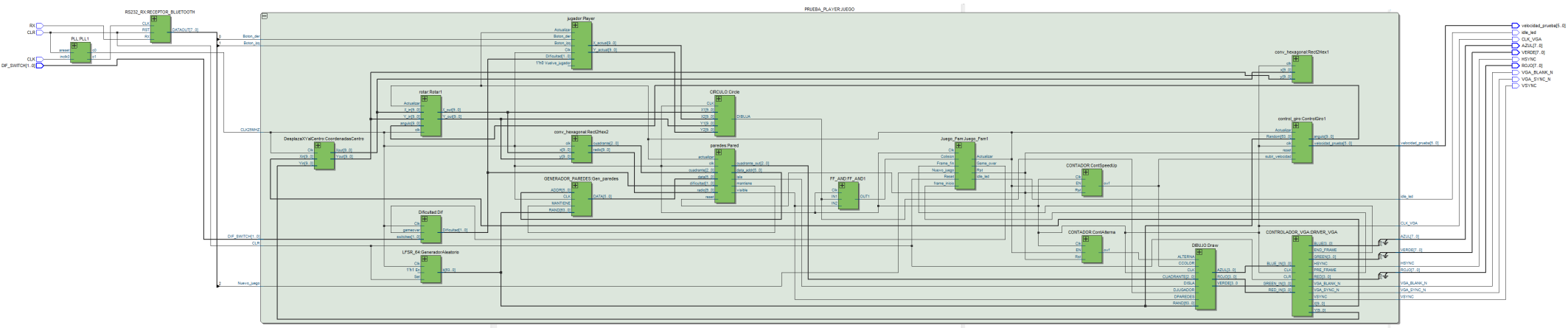
La implementación de una interfaz HDMI para el uso en pantallas digitales y en alta resolución. Esto sugiere el cambio de la placa utilizada ya que la DE2-115 no posee soporte nativo para dicha interfaz.

Música ambiental para el juego. Esto sugiere el uso del códec Wolfson WM8731 provisto por la placa.

Permitir que el puntero tope contra el costado de los obstáculos sin perder. Esto permitiría mejorar la jugabilidad ya que en estos casos, el puntero no es “aplastado” por el obstáculo.

Modo multijugador cooperativo/competitivo. La implementación de este sistema aumentaría la complejidad del juego.

# Apéndice A: Diagrama del sistema



Figura

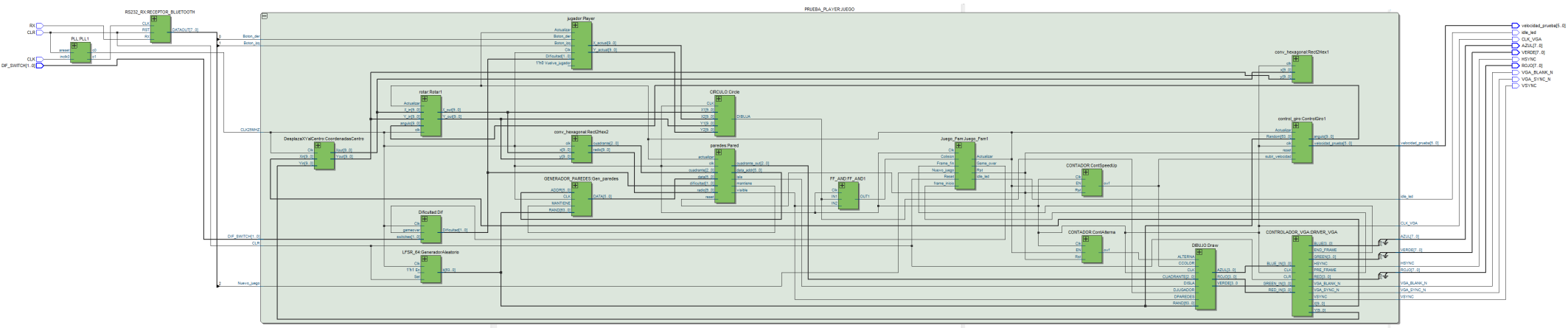


Figura 9

# 

# Apéndice B

Archivo SUPERHEXAGON2016.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use work.HexaPackage.all;

ENTITY SUPERHEXAGON2016 IS

PORT(

CLK : IN STD\_LOGIC;

CLR : IN STD\_LOGIC;

DIF\_SWITCH : IN STD\_LOGIC\_VECTOR(1 DOWNTO 0);

RX : IN STD\_LOGIC;

CLK\_VGA : OUT STD\_LOGIC;

HSYNC : OUT STD\_LOGIC;

VSYNC : OUT STD\_LOGIC;

VGA\_SYNC\_N : OUT STD\_LOGIC;

VGA\_BLANK\_N : OUT STD\_LOGIC;

ROJO,VERDE,AZUL : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0);

velocidad\_prueba : out unsigned(5 DOWNTO 0);

idle\_led : out std\_logic

);

END ENTITY;

----------------------------------------------------------------------------------------------------------------

ARCHITECTURE BEH OF SUPERHEXAGON2016 IS

COMPONENT PLL IS

PORT

(

areset : IN STD\_LOGIC := '0';

inclk0 : IN STD\_LOGIC := '0';

c0 : OUT STD\_LOGIC ;

c1 : OUT STD\_LOGIC ;

c2 : OUT STD\_LOGIC

);

END COMPONENT;

COMPONENT PRUEBA\_PLAYER IS

PORT(

CLK25MHZ : IN STD\_LOGIC;

CLR : IN STD\_LOGIC;

DIF\_SWITCH : IN STD\_LOGIC\_VECTOR(1 DOWNTO 0);

Nuevo\_juego : in std\_logic;

Boton\_der : in std\_logic;

Boton\_izq : in std\_logic;

CLK\_VGA : OUT STD\_LOGIC;

HSYNC : OUT STD\_LOGIC;

VSYNC : OUT STD\_LOGIC;

VGA\_SYNC\_N : OUT STD\_LOGIC;

VGA\_BLANK\_N : OUT STD\_LOGIC;

ROJO,VERDE,AZUL : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0);

velocidad\_prueba : out unsigned(5 DOWNTO 0);

idle\_led : out std\_logic

);

END COMPONENT;

COMPONENT RS232\_RX IS

PORT(

CLK: IN STD\_LOGIC;

RST: IN STD\_LOGIC;

RX: IN STD\_LOGIC;

DATA\_OK: OUT STD\_LOGIC;

DATAOUT: OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END COMPONENT;

SIGNAL CLK25MHZ,CLKBAUD : STD\_LOGIC;

SIGNAL DATA : STD\_LOGIC\_VECTOR(7 DOWNTO 0);

BEGIN

PLL1: PLL PORT MAP(

areset => CLR,

inclk0 => CLK,

c0 => CLK25MHZ,

c1 => CLKBAUD,

c2 => OPEN

);

JUEGO: PRUEBA\_PLAYER PORT MAP(

CLK25MHZ =>CLK25MHZ,

CLR =>CLR,

DIF\_SWITCH =>DIF\_SWITCH,

Nuevo\_juego =>DATA(2),

Boton\_der =>DATA(0),

Boton\_izq =>DATA(1),

CLK\_VGA =>CLK\_VGA,

HSYNC =>HSYNC,

VSYNC =>VSYNC,

VGA\_SYNC\_N =>VGA\_SYNC\_N,

VGA\_BLANK\_N =>VGA\_BLANK\_N,

ROJO =>ROJO,

VERDE =>VERDE,

AZUL =>AZUL,

velocidad\_prueba =>velocidad\_prueba,

idle\_led =>idle\_led

);

RECEPTOR\_BLUETOOTH: RS232\_RX PORT MAP(

CLK =>CLKBAUD,

RST =>CLR,

RX =>RX,

DATA\_OK =>OPEN,

DATAOUT =>DATA

);

END ARCHITECTURE;

# Apéndice C

Archivo SHIFT\_REGISTER\_RX.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

ENTITY SHIFT\_REGISTER\_RX IS

GENERIC(

data\_width: integer:=8

);

PORT(

CLK: IN STD\_LOGIC;

RST: IN STD\_LOGIC;

DATAIN: IN STD\_LOGIC;

SHIFT: IN STD\_LOGIC;

DATAOUT: OUT STD\_LOGIC\_VECTOR(data\_width-1 DOWNTO 0)

);

END SHIFT\_REGISTER\_RX;

ARCHITECTURE BEH OF SHIFT\_REGISTER\_RX IS

SIGNAL DATA: STD\_LOGIC\_VECTOR(data\_width-1 DOWNTO 0);

BEGIN

DATAOUT<=DATA;

SR: PROCESS (CLK, RST, SHIFT)

BEGIN

IF(RST='1') THEN

DATA<=(OTHERS=>'0');

ELSIF(RISING\_EDGE(CLK)) THEN

IF (SHIFT='1') THEN

DATA <= STD\_LOGIC\_VECTOR(UNSIGNED(DATA) SRL 1);

DATA(data\_width-1) <= DATAIN;

END IF;

END IF;

END PROCESS;

END BEH;

# Apéndice D

Archivo RS232\_RX\_FSM.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

ENTITY RS232\_RX\_FSM IS

PORT(

CLK: IN STD\_LOGIC;

RST: IN STD\_LOGIC;

TX: IN STD\_LOGIC;

SHIFT: OUT STD\_LOGIC;

DATA\_OK: OUT STD\_LOGIC

);

END RS232\_RX\_FSM;

ARCHITECTURE BEH OF RS232\_RX\_FSM IS

SIGNAL CONT: UNSIGNED(2 DOWNTO 0);

SIGNAL CONTCTRL: STD\_LOGIC\_VECTOR(1 DOWNTO 0); --00: NADA, 01: SUMA, 1X: RESETEA

TYPE STATE IS (IDLE, START, DATA, PARITY, STOP);

SIGNAL CS, NS: STATE;

BEGIN

NEXT\_STATE: PROCESS (CS, TX, CONT)

BEGIN

CASE CS IS

WHEN IDLE =>

IF (TX='0') THEN

NS <= DATA;

ELSE

NS <= IDLE;

END IF;

WHEN START => NS <= DATA;

WHEN DATA =>

IF (CONT = "111") THEN

NS <= STOP;

ELSE

NS <= DATA;

END IF;

WHEN PARITY => NS <= STOP;

WHEN STOP =>

IF(TX='1') THEN

NS <= IDLE;

ELSE

NS <= STOP;

END IF;

WHEN OTHERS => NS <= IDLE;

END CASE;

END PROCESS;

CURRENT\_STATE: PROCESS (CLK, RST)

BEGIN

IF(RST='1') THEN

CS <= IDLE;

ELSIF(RISING\_EDGE(CLK)) THEN

CS <= NS;

END IF;

END PROCESS;

OUTPUTS: PROCESS (CS,RST,TX)

BEGIN

IF (RST='1') THEN

SHIFT <='0';

DATA\_OK <='0';

CONTCTRL<="11";

ELSE

SHIFT <='0';

DATA\_OK <='0';

CONTCTRL<="00";

CASE CS IS

WHEN IDLE => CONTCTRL<="11";

WHEN START => NULL;

WHEN DATA =>

CONTCTRL<="01";

SHIFT <= '1';

--WHEN PARITY =>

--SHIFT <= '1';

WHEN STOP =>

IF(TX='1') THEN

DATA\_OK<='1';

END IF;

WHEN OTHERS => CONTCTRL<="11";

END CASE;

END IF;

END PROCESS;

COUNT\_UP: PROCESS (CLK,CONTCTRL)

BEGIN

IF(RISING\_EDGE(CLK)) THEN

IF(CONTCTRL="01") THEN

CONT<=CONT+1;

ELSIF(CONTCTRL(1)='1') THEN

CONT<="000";

END IF;

END IF;

END PROCESS;

END BEH;

# Apéndice E

Archivo DATA\_BUFFER.vhd

library ieee;

use ieee.std\_logic\_1164.all;

ENTITY DATA\_BUFFER IS

GENERIC(

data\_width: integer:=8

);

PORT(

DATAIN: IN STD\_LOGIC\_VECTOR(data\_width-1 DOWNTO 0);

EN: IN STD\_LOGIC;

CLK: IN STD\_LOGIC;

CLEAR: IN STD\_LOGIC;

DATAOUT: OUT STD\_LOGIC\_VECTOR(data\_width-1 DOWNTO 0)

);

END DATA\_BUFFER;

ARCHITECTURE BEH OF DATA\_BUFFER IS

BEGIN

STORE: PROCESS (CLK,EN,CLEAR)

BEGIN

IF (CLEAR='1') THEN

DATAOUT<=(OTHERS=>'0');

ELSIF (RISING\_EDGE(CLK) AND EN='1') THEN

DATAOUT<=DATAIN;

END IF;

END PROCESS;

END BEH;

# Apéndice F

Archivo HexaPackage.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

--use work.HexaPackage.all;

PACKAGE HexaPackage IS

CONSTANT BITS : INTEGER:= 10;

function bit2bool(b: std\_logic) return boolean;

CONSTANT CENTROX : SIGNED:= TO\_SIGNED(320,BITS);

CONSTANT CENTROY : SIGNED:= TO\_SIGNED(240,BITS);

--------------------------------------------------------------------------------------------------------------------

----------------------------------------------COMPONENTES-----------------------------------------------------------

--------------------------------------------------------------------------------------------------------------------

COMPONENT PLL IS

PORT

(

areset : IN STD\_LOGIC := '0';

inclk0 : IN STD\_LOGIC := '0';

c0 : OUT STD\_LOGIC

);

END COMPONENT;

COMPONENT CONTROLADOR\_VGA IS

GENERIC( BITS: INTEGER:=10);

PORT(

CLK : IN STD\_LOGIC;

CLR : IN STD\_LOGIC;

RED\_IN : IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

GREEN\_IN : IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

BLUE\_IN : IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

HSYNC : OUT STD\_LOGIC;

VSYNC : OUT STD\_LOGIC;

VGA\_SYNC\_N : OUT STD\_LOGIC;

VGA\_BLANK\_N : OUT STD\_LOGIC;

X : OUT UNSIGNED(BITS-1 DOWNTO 0);

Y : OUT UNSIGNED(BITS-1 DOWNTO 0);

RED : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

GREEN : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

BLUE : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

END\_FRAME : OUT STD\_LOGIC;

PRE\_FRAME : OUT STD\_LOGIC

);

END COMPONENT;

component DesplazaXYalCentro is

port(

Clk : in std\_logic;

Xin : in unsigned(9 downto 0);

Yin : in unsigned(9 downto 0);

Xout : out signed(9 downto 0);

Yout : out signed(9 downto 0)

);

end component;

component rotar is

port(

clk : in std\_logic;

Actualizar : in std\_logic;

X\_in : in signed(9 downto 0);

Y\_in : in signed(9 downto 0);

angulo : in unsigned(9 downto 0);

X\_out : out signed(9 downto 0);

Y\_out : out signed(9 downto 0)

);

end component;

component conv\_hexagonal is

port(

clk : in std\_logic;

x : in signed (9 downto 0);

y : in signed (9 downto 0);

cuadrante : out unsigned (2 downto 0);

radio : out unsigned (9 downto 0)

);

end component;

COMPONENT GENERADOR\_PAREDES IS

PORT(

CLK : IN STD\_LOGIC;

RAND : IN STD\_LOGIC\_VECTOR(63 DOWNTO 0);

MANTIENE : IN STD\_LOGIC;

ADDR : IN STD\_LOGIC\_VECTOR(5 DOWNTO 0);

DATA : OUT STD\_LOGIC\_VECTOR(5 DOWNTO 0)

);

END COMPONENT;

COMPONENT PAREDES IS

port(

clk : in std\_logic;

reset : in std\_logic;

actualizar : in std\_logic;

dificultad : in std\_logic\_vector(1 downto 0);

data : in std\_logic\_vector(5 downto 0);

cuadrante : in unsigned(2 downto 0);

radio : in unsigned(9 downto 0);

mantiene : out std\_logic;

data\_addr : out std\_logic\_vector(5 downto 0);

cuadrante\_out : out std\_logic\_vector(2 downto 0);

visible : out std\_logic;

isla : out std\_logic

);

END COMPONENT;

component control\_giro is

port(

clk : in std\_logic;

reset : in std\_logic;

Actualizar : in std\_logic;

Random : in std\_logic\_vector(63 downto 0);

Subir\_velocidad : in std\_logic;

velocidad\_prueba : out unsigned(5 downto 0);

angulo : out unsigned(9 downto 0)

);

end component;

component Juego\_Fsm is

port(

Clk : in std\_logic;

Frame\_inicio : in std\_logic;

Frame\_fin : in std\_logic;

Nuevo\_juego : in std\_logic;

Colision : in std\_logic;

Reset : in std\_logic;

Actualizar : out std\_logic;

Rst : out std\_logic;

idle\_led : out std\_logic;

Game\_over : out std\_logic

);

end component;

component CONTADOR is

generic(max:integer:=16);

port

(

-- Input ports

Clk : in std\_logic;

EN : in std\_logic;

Rst : in std\_logic;

-- Output ports

ovf: out std\_logic

);

end component;

component LFSR\_64 is

port

(

-- Input ports

Clk : in std\_logic;

Set : in std\_logic;

En : in std\_logic;

-- Output ports

b : out std\_logic\_vector(63 downto 0)

);

end component;

component senocoseno is

port(

clk : in std\_logic;

angulo : in unsigned(9 downto 0);

seno : out signed(11 downto 0);

coseno : out signed(11 downto 0)

);

end component;

COMPONENT DIBUJO IS

PORT(

CLK : IN STD\_LOGIC;

ALTERNA : IN STD\_LOGIC;

CCOLOR : IN STD\_LOGIC;

--RESET : IN STD\_LOGIC;

DPAREDES : IN STD\_LOGIC;

DISLA : IN STD\_LOGIC;

DJUGADOR : IN STD\_LOGIC;

CUADRANTE: IN STD\_LOGIC\_VECTOR(2 DOWNTO 0);

RAND : IN STD\_LOGIC\_VECTOR(63 DOWNTO 0);

ROJO : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

VERDE : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

AZUL : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0)

);

end component;

component jugador is

port(

Clk : in std\_logic;

Actualizar : in std\_logic;

Vuelve\_jugador : in std\_logic;

Dificultad : in std\_logic\_vector(1 downto 0);

Boton\_der : in std\_logic;

Boton\_izq : in std\_logic;

X\_actual : out signed(9 downto 0);

Y\_actual : out signed(9 downto 0);

X\_anterior : out signed(9 downto 0);

Y\_anterior : out signed(9 downto 0)

);

end component;

COMPONENT CIRCULO IS

PORT(

CLK : IN STD\_LOGIC;

X1 : IN SIGNED(9 DOWNTO 0);

Y1 : IN SIGNED(9 DOWNTO 0);

X2 : IN SIGNED(9 DOWNTO 0);

Y2 : IN SIGNED(9 DOWNTO 0);

DIBUJA: OUT STD\_LOGIC

);

END COMPONENT;

component Dificultad is

port(

Clk : in std\_logic;

switches : in std\_logic\_vector(1 downto 0);

gameover : in std\_logic;

Dificultad : out std\_logic\_vector(1 downto 0);

Leds\_Rojo : out std\_logic\_vector(7 downto 0);

Leds\_Verde : out std\_logic\_vector(7 downto 0)

);

end component;

component Compara\_coordenadas is

port(

--inputs

Xin1 : in signed(9 downto 0);

Yin1 : in signed(9 downto 0);

Xin2 : in signed(9 downto 0);

Yin2 : in signed(9 downto 0);

--outputs

Comparacion : out std\_logic

);

end component;

component FF\_AND is

port(

Clk : in std\_logic;

IN1 : in std\_logic;

IN2 : in std\_logic;

OUT1 : out std\_logic

);

end component;

--------------------------------------------------------------------------------------------------------------------

END PACKAGE;

--------------------------------------------------------------------------------------------------------------------

PACKAGE BODY HexaPackage IS

function bit2bool(b: std\_logic) return boolean is

begin

if (b='1') then

return true;

else

return false;

end if;

end function bit2bool;

END PACKAGE BODY;

# Apéndice G

Archivo senocoseno.vhd

--LOOK-UP TABLE DE LAS FUNCIONES SENO Y COSENO.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity senocoseno is

port(

clk : in std\_logic;

angulo : in unsigned(9 downto 0);

seno : out signed(11 downto 0);

coseno : out signed(11 downto 0)

);

end senocoseno;

architecture beh of senocoseno is

constant data\_width : natural := 12;

constant addr\_width : natural := 10;

constant mem\_size : natural := 2\*\*addr\_width;

subtype rom\_word is signed(data\_width-1 downto 0);

type mem\_type is array (mem\_size-1 downto 0) of rom\_word;

constant mem : mem\_type :=

(0000 => to\_signed(0,12),

0001 => to\_signed(12,12),

0002 => to\_signed(25,12),

0003 => to\_signed(37,12),

0004 => to\_signed(50,12),

0005 => to\_signed(62,12),

0006 => to\_signed(75,12),

0007 => to\_signed(87,12),

0008 => to\_signed(100,12),

0009 => to\_signed(113,12),

0010 => to\_signed(125,12),

0011 => to\_signed(138,12),

0012 => to\_signed(150,12),

0013 => to\_signed(163,12),

0014 => to\_signed(175,12),

0015 => to\_signed(188,12),

0016 => to\_signed(200,12),

0017 => to\_signed(213,12),

0018 => to\_signed(225,12),

0019 => to\_signed(238,12),

0020 => to\_signed(250,12),

0021 => to\_signed(263,12),

0022 => to\_signed(275,12),

0023 => to\_signed(288,12),

0024 => to\_signed(300,12),

0025 => to\_signed(312,12),

0026 => to\_signed(325,12),

0027 => to\_signed(337,12),

0028 => to\_signed(350,12),

0029 => to\_signed(362,12),

0030 => to\_signed(374,12),

0031 => to\_signed(387,12),

0032 => to\_signed(399,12),

0033 => to\_signed(411,12),

0034 => to\_signed(424,12),

0035 => to\_signed(436,12),

0036 => to\_signed(448,12),

0037 => to\_signed(460,12),

0038 => to\_signed(473,12),

0039 => to\_signed(485,12),

0040 => to\_signed(497,12),

0041 => to\_signed(509,12),

0042 => to\_signed(521,12),

0043 => to\_signed(534,12),

0044 => to\_signed(546,12),

0045 => to\_signed(558,12),

0046 => to\_signed(570,12),

0047 => to\_signed(582,12),

0048 => to\_signed(594,12),

0049 => to\_signed(606,12),

0050 => to\_signed(618,12),

0051 => to\_signed(630,12),

0052 => to\_signed(642,12),

0053 => to\_signed(654,12),

0054 => to\_signed(666,12),

0055 => to\_signed(678,12),

0056 => to\_signed(689,12),

0057 => to\_signed(701,12),

0058 => to\_signed(713,12),

0059 => to\_signed(725,12),

0060 => to\_signed(737,12),

0061 => to\_signed(748,12),

0062 => to\_signed(760,12),

0063 => to\_signed(772,12),

0064 => to\_signed(783,12),

0065 => to\_signed(795,12),

0066 => to\_signed(806,12),

0067 => to\_signed(818,12),

0068 => to\_signed(829,12),

0069 => to\_signed(841,12),

0070 => to\_signed(852,12),

0071 => to\_signed(864,12),

0072 => to\_signed(875,12),

0073 => to\_signed(886,12),

0074 => to\_signed(898,12),

0075 => to\_signed(909,12),

0076 => to\_signed(920,12),

0077 => to\_signed(932,12),

0078 => to\_signed(943,12),

0079 => to\_signed(954,12),

0080 => to\_signed(965,12),

0081 => to\_signed(976,12),

0082 => to\_signed(987,12),

0083 => to\_signed(998,12),

0084 => to\_signed(1009,12),

0085 => to\_signed(1020,12),

0086 => to\_signed(1031,12),

0087 => to\_signed(1042,12),

0088 => to\_signed(1052,12),

0089 => to\_signed(1063,12),

0090 => to\_signed(1074,12),

0091 => to\_signed(1085,12),

0092 => to\_signed(1095,12),

0093 => to\_signed(1106,12),

0094 => to\_signed(1116,12),

0095 => to\_signed(1127,12),

0096 => to\_signed(1137,12),

0097 => to\_signed(1148,12),

0098 => to\_signed(1158,12),

0099 => to\_signed(1168,12),

0100 => to\_signed(1179,12),

0101 => to\_signed(1189,12),

0102 => to\_signed(1199,12),

0103 => to\_signed(1209,12),

0104 => to\_signed(1219,12),

0105 => to\_signed(1230,12),

0106 => to\_signed(1240,12),

0107 => to\_signed(1250,12),

0108 => to\_signed(1259,12),

0109 => to\_signed(1269,12),

0110 => to\_signed(1279,12),

0111 => to\_signed(1289,12),

0112 => to\_signed(1299,12),

0113 => to\_signed(1308,12),

0114 => to\_signed(1318,12),

0115 => to\_signed(1328,12),

0116 => to\_signed(1337,12),

0117 => to\_signed(1347,12),

0118 => to\_signed(1356,12),

0119 => to\_signed(1366,12),

0120 => to\_signed(1375,12),

0121 => to\_signed(1384,12),

0122 => to\_signed(1393,12),

0123 => to\_signed(1403,12),

0124 => to\_signed(1412,12),

0125 => to\_signed(1421,12),

0126 => to\_signed(1430,12),

0127 => to\_signed(1439,12),

0128 => to\_signed(1448,12),

0129 => to\_signed(1457,12),

0130 => to\_signed(1465,12),

0131 => to\_signed(1474,12),

0132 => to\_signed(1483,12),

0133 => to\_signed(1491,12),

0134 => to\_signed(1500,12),

0135 => to\_signed(1509,12),

0136 => to\_signed(1517,12),

0137 => to\_signed(1525,12),

0138 => to\_signed(1534,12),

0139 => to\_signed(1542,12),

0140 => to\_signed(1550,12),

0141 => to\_signed(1558,12),

0142 => to\_signed(1567,12),

0143 => to\_signed(1575,12),

0144 => to\_signed(1583,12),

0145 => to\_signed(1591,12),

0146 => to\_signed(1598,12),

0147 => to\_signed(1606,12),

0148 => to\_signed(1614,12),

0149 => to\_signed(1622,12),

0150 => to\_signed(1629,12),

0151 => to\_signed(1637,12),

0152 => to\_signed(1644,12),

0153 => to\_signed(1652,12),

0154 => to\_signed(1659,12),

0155 => to\_signed(1667,12),

0156 => to\_signed(1674,12),

0157 => to\_signed(1681,12),

0158 => to\_signed(1688,12),

0159 => to\_signed(1695,12),

0160 => to\_signed(1702,12),

0161 => to\_signed(1709,12),

0162 => to\_signed(1716,12),

0163 => to\_signed(1723,12),

0164 => to\_signed(1730,12),

0165 => to\_signed(1736,12),

0166 => to\_signed(1743,12),

0167 => to\_signed(1750,12),

0168 => to\_signed(1756,12),

0169 => to\_signed(1763,12),

0170 => to\_signed(1769,12),

0171 => to\_signed(1775,12),

0172 => to\_signed(1781,12),

0173 => to\_signed(1788,12),

0174 => to\_signed(1794,12),

0175 => to\_signed(1800,12),

0176 => to\_signed(1806,12),

0177 => to\_signed(1812,12),

0178 => to\_signed(1817,12),

0179 => to\_signed(1823,12),

0180 => to\_signed(1829,12),

0181 => to\_signed(1834,12),

0182 => to\_signed(1840,12),

0183 => to\_signed(1845,12),

0184 => to\_signed(1851,12),

0185 => to\_signed(1856,12),

0186 => to\_signed(1861,12),

0187 => to\_signed(1867,12),

0188 => to\_signed(1872,12),

0189 => to\_signed(1877,12),

0190 => to\_signed(1882,12),

0191 => to\_signed(1887,12),

0192 => to\_signed(1892,12),

0193 => to\_signed(1896,12),

0194 => to\_signed(1901,12),

0195 => to\_signed(1906,12),

0196 => to\_signed(1910,12),

0197 => to\_signed(1915,12),

0198 => to\_signed(1919,12),

0199 => to\_signed(1924,12),

0200 => to\_signed(1928,12),

0201 => to\_signed(1932,12),

0202 => to\_signed(1936,12),

0203 => to\_signed(1940,12),

0204 => to\_signed(1944,12),

0205 => to\_signed(1948,12),

0206 => to\_signed(1952,12),

0207 => to\_signed(1956,12),

0208 => to\_signed(1959,12),

0209 => to\_signed(1963,12),

0210 => to\_signed(1966,12),

0211 => to\_signed(1970,12),

0212 => to\_signed(1973,12),

0213 => to\_signed(1977,12),

0214 => to\_signed(1980,12),

0215 => to\_signed(1983,12),

0216 => to\_signed(1986,12),

0217 => to\_signed(1989,12),

0218 => to\_signed(1992,12),

0219 => to\_signed(1995,12),

0220 => to\_signed(1998,12),

0221 => to\_signed(2000,12),

0222 => to\_signed(2003,12),

0223 => to\_signed(2006,12),

0224 => to\_signed(2008,12),

0225 => to\_signed(2011,12),

0226 => to\_signed(2013,12),

0227 => to\_signed(2015,12),

0228 => to\_signed(2017,12),

0229 => to\_signed(2019,12),

0230 => to\_signed(2021,12),

0231 => to\_signed(2023,12),

0232 => to\_signed(2025,12),

0233 => to\_signed(2027,12),

0234 => to\_signed(2029,12),

0235 => to\_signed(2031,12),

0236 => to\_signed(2032,12),

0237 => to\_signed(2034,12),

0238 => to\_signed(2035,12),

0239 => to\_signed(2036,12),

0240 => to\_signed(2038,12),

0241 => to\_signed(2039,12),

0242 => to\_signed(2040,12),

0243 => to\_signed(2041,12),

0244 => to\_signed(2042,12),

0245 => to\_signed(2043,12),

0246 => to\_signed(2044,12),

0247 => to\_signed(2044,12),

0248 => to\_signed(2045,12),

0249 => to\_signed(2046,12),

0250 => to\_signed(2046,12),

0251 => to\_signed(2047,12),

0252 => to\_signed(2047,12),

0253 => to\_signed(2047,12),

0254 => to\_signed(2047,12),

0255 => to\_signed(2047,12),

0256 => to\_signed(2047,12),

0257 => to\_signed(2047,12),

0258 => to\_signed(2047,12),

0259 => to\_signed(2047,12),

0260 => to\_signed(2047,12),

0261 => to\_signed(2047,12),

0262 => to\_signed(2046,12),

0263 => to\_signed(2046,12),

0264 => to\_signed(2045,12),

0265 => to\_signed(2044,12),

0266 => to\_signed(2044,12),

0267 => to\_signed(2043,12),

0268 => to\_signed(2042,12),

0269 => to\_signed(2041,12),

0270 => to\_signed(2040,12),

0271 => to\_signed(2039,12),

0272 => to\_signed(2038,12),

0273 => to\_signed(2036,12),

0274 => to\_signed(2035,12),

0275 => to\_signed(2034,12),

0276 => to\_signed(2032,12),

0277 => to\_signed(2031,12),

0278 => to\_signed(2029,12),

0279 => to\_signed(2027,12),

0280 => to\_signed(2025,12),

0281 => to\_signed(2023,12),

0282 => to\_signed(2021,12),

0283 => to\_signed(2019,12),

0284 => to\_signed(2017,12),

0285 => to\_signed(2015,12),

0286 => to\_signed(2013,12),

0287 => to\_signed(2011,12),

0288 => to\_signed(2008,12),

0289 => to\_signed(2006,12),

0290 => to\_signed(2003,12),

0291 => to\_signed(2000,12),

0292 => to\_signed(1998,12),

0293 => to\_signed(1995,12),

0294 => to\_signed(1992,12),

0295 => to\_signed(1989,12),

0296 => to\_signed(1986,12),

0297 => to\_signed(1983,12),

0298 => to\_signed(1980,12),

0299 => to\_signed(1977,12),

0300 => to\_signed(1973,12),

0301 => to\_signed(1970,12),

0302 => to\_signed(1966,12),

0303 => to\_signed(1963,12),

0304 => to\_signed(1959,12),

0305 => to\_signed(1956,12),

0306 => to\_signed(1952,12),

0307 => to\_signed(1948,12),

0308 => to\_signed(1944,12),

0309 => to\_signed(1940,12),

0310 => to\_signed(1936,12),

0311 => to\_signed(1932,12),

0312 => to\_signed(1928,12),

0313 => to\_signed(1924,12),

0314 => to\_signed(1919,12),

0315 => to\_signed(1915,12),

0316 => to\_signed(1910,12),

0317 => to\_signed(1906,12),

0318 => to\_signed(1901,12),

0319 => to\_signed(1896,12),

0320 => to\_signed(1892,12),

0321 => to\_signed(1887,12),

0322 => to\_signed(1882,12),

0323 => to\_signed(1877,12),

0324 => to\_signed(1872,12),

0325 => to\_signed(1867,12),

0326 => to\_signed(1861,12),

0327 => to\_signed(1856,12),

0328 => to\_signed(1851,12),

0329 => to\_signed(1845,12),

0330 => to\_signed(1840,12),

0331 => to\_signed(1834,12),

0332 => to\_signed(1829,12),

0333 => to\_signed(1823,12),

0334 => to\_signed(1817,12),

0335 => to\_signed(1812,12),

0336 => to\_signed(1806,12),

0337 => to\_signed(1800,12),

0338 => to\_signed(1794,12),

0339 => to\_signed(1788,12),

0340 => to\_signed(1781,12),

0341 => to\_signed(1775,12),

0342 => to\_signed(1769,12),

0343 => to\_signed(1763,12),

0344 => to\_signed(1756,12),

0345 => to\_signed(1750,12),

0346 => to\_signed(1743,12),

0347 => to\_signed(1736,12),

0348 => to\_signed(1730,12),

0349 => to\_signed(1723,12),

0350 => to\_signed(1716,12),

0351 => to\_signed(1709,12),

0352 => to\_signed(1702,12),

0353 => to\_signed(1695,12),

0354 => to\_signed(1688,12),

0355 => to\_signed(1681,12),

0356 => to\_signed(1674,12),

0357 => to\_signed(1667,12),

0358 => to\_signed(1659,12),

0359 => to\_signed(1652,12),

0360 => to\_signed(1644,12),

0361 => to\_signed(1637,12),

0362 => to\_signed(1629,12),

0363 => to\_signed(1622,12),

0364 => to\_signed(1614,12),

0365 => to\_signed(1606,12),

0366 => to\_signed(1598,12),

0367 => to\_signed(1591,12),

0368 => to\_signed(1583,12),

0369 => to\_signed(1575,12),

0370 => to\_signed(1567,12),

0371 => to\_signed(1558,12),

0372 => to\_signed(1550,12),

0373 => to\_signed(1542,12),

0374 => to\_signed(1534,12),

0375 => to\_signed(1525,12),

0376 => to\_signed(1517,12),

0377 => to\_signed(1509,12),

0378 => to\_signed(1500,12),

0379 => to\_signed(1491,12),

0380 => to\_signed(1483,12),

0381 => to\_signed(1474,12),

0382 => to\_signed(1465,12),

0383 => to\_signed(1457,12),

0384 => to\_signed(1448,12),

0385 => to\_signed(1439,12),

0386 => to\_signed(1430,12),

0387 => to\_signed(1421,12),

0388 => to\_signed(1412,12),

0389 => to\_signed(1403,12),

0390 => to\_signed(1393,12),

0391 => to\_signed(1384,12),

0392 => to\_signed(1375,12),

0393 => to\_signed(1366,12),

0394 => to\_signed(1356,12),

0395 => to\_signed(1347,12),

0396 => to\_signed(1337,12),

0397 => to\_signed(1328,12),

0398 => to\_signed(1318,12),

0399 => to\_signed(1308,12),

0400 => to\_signed(1299,12),

0401 => to\_signed(1289,12),

0402 => to\_signed(1279,12),

0403 => to\_signed(1269,12),

0404 => to\_signed(1259,12),

0405 => to\_signed(1250,12),

0406 => to\_signed(1240,12),

0407 => to\_signed(1230,12),

0408 => to\_signed(1219,12),

0409 => to\_signed(1209,12),

0410 => to\_signed(1199,12),

0411 => to\_signed(1189,12),

0412 => to\_signed(1179,12),

0413 => to\_signed(1168,12),

0414 => to\_signed(1158,12),

0415 => to\_signed(1148,12),

0416 => to\_signed(1137,12),

0417 => to\_signed(1127,12),

0418 => to\_signed(1116,12),

0419 => to\_signed(1106,12),

0420 => to\_signed(1095,12),

0421 => to\_signed(1085,12),

0422 => to\_signed(1074,12),

0423 => to\_signed(1063,12),

0424 => to\_signed(1052,12),

0425 => to\_signed(1042,12),

0426 => to\_signed(1031,12),

0427 => to\_signed(1020,12),

0428 => to\_signed(1009,12),

0429 => to\_signed(998,12),

0430 => to\_signed(987,12),

0431 => to\_signed(976,12),

0432 => to\_signed(965,12),

0433 => to\_signed(954,12),

0434 => to\_signed(943,12),

0435 => to\_signed(932,12),

0436 => to\_signed(920,12),

0437 => to\_signed(909,12),

0438 => to\_signed(898,12),

0439 => to\_signed(886,12),

0440 => to\_signed(875,12),

0441 => to\_signed(864,12),

0442 => to\_signed(852,12),

0443 => to\_signed(841,12),

0444 => to\_signed(829,12),

0445 => to\_signed(818,12),

0446 => to\_signed(806,12),

0447 => to\_signed(795,12),

0448 => to\_signed(783,12),

0449 => to\_signed(772,12),

0450 => to\_signed(760,12),

0451 => to\_signed(748,12),

0452 => to\_signed(737,12),

0453 => to\_signed(725,12),

0454 => to\_signed(713,12),

0455 => to\_signed(701,12),

0456 => to\_signed(689,12),

0457 => to\_signed(678,12),

0458 => to\_signed(666,12),

0459 => to\_signed(654,12),

0460 => to\_signed(642,12),

0461 => to\_signed(630,12),

0462 => to\_signed(618,12),

0463 => to\_signed(606,12),

0464 => to\_signed(594,12),

0465 => to\_signed(582,12),

0466 => to\_signed(570,12),

0467 => to\_signed(558,12),

0468 => to\_signed(546,12),

0469 => to\_signed(534,12),

0470 => to\_signed(521,12),

0471 => to\_signed(509,12),

0472 => to\_signed(497,12),

0473 => to\_signed(485,12),

0474 => to\_signed(473,12),

0475 => to\_signed(460,12),

0476 => to\_signed(448,12),

0477 => to\_signed(436,12),

0478 => to\_signed(424,12),

0479 => to\_signed(411,12),

0480 => to\_signed(399,12),

0481 => to\_signed(387,12),

0482 => to\_signed(374,12),

0483 => to\_signed(362,12),

0484 => to\_signed(350,12),

0485 => to\_signed(337,12),

0486 => to\_signed(325,12),

0487 => to\_signed(312,12),

0488 => to\_signed(300,12),

0489 => to\_signed(288,12),

0490 => to\_signed(275,12),

0491 => to\_signed(263,12),

0492 => to\_signed(250,12),

0493 => to\_signed(238,12),

0494 => to\_signed(225,12),

0495 => to\_signed(213,12),

0496 => to\_signed(200,12),

0497 => to\_signed(188,12),

0498 => to\_signed(175,12),

0499 => to\_signed(163,12),

0500 => to\_signed(150,12),

0501 => to\_signed(138,12),

0502 => to\_signed(125,12),

0503 => to\_signed(113,12),

0504 => to\_signed(100,12),

0505 => to\_signed(87,12),

0506 => to\_signed(75,12),

0507 => to\_signed(62,12),

0508 => to\_signed(50,12),

0509 => to\_signed(37,12),

0510 => to\_signed(25,12),

0511 => to\_signed(12,12),

0512 => to\_signed(0,12),

0513 => to\_signed(-12,12),

0514 => to\_signed(-25,12),

0515 => to\_signed(-37,12),

0516 => to\_signed(-50,12),

0517 => to\_signed(-62,12),

0518 => to\_signed(-75,12),

0519 => to\_signed(-87,12),

0520 => to\_signed(-100,12),

0521 => to\_signed(-113,12),

0522 => to\_signed(-125,12),

0523 => to\_signed(-138,12),

0524 => to\_signed(-150,12),

0525 => to\_signed(-163,12),

0526 => to\_signed(-175,12),

0527 => to\_signed(-188,12),

0528 => to\_signed(-200,12),

0529 => to\_signed(-213,12),

0530 => to\_signed(-225,12),

0531 => to\_signed(-238,12),

0532 => to\_signed(-250,12),

0533 => to\_signed(-263,12),

0534 => to\_signed(-275,12),

0535 => to\_signed(-288,12),

0536 => to\_signed(-300,12),

0537 => to\_signed(-312,12),

0538 => to\_signed(-325,12),

0539 => to\_signed(-337,12),

0540 => to\_signed(-350,12),

0541 => to\_signed(-362,12),

0542 => to\_signed(-374,12),

0543 => to\_signed(-387,12),

0544 => to\_signed(-399,12),

0545 => to\_signed(-411,12),

0546 => to\_signed(-424,12),

0547 => to\_signed(-436,12),

0548 => to\_signed(-448,12),

0549 => to\_signed(-460,12),

0550 => to\_signed(-473,12),

0551 => to\_signed(-485,12),

0552 => to\_signed(-497,12),

0553 => to\_signed(-509,12),

0554 => to\_signed(-521,12),

0555 => to\_signed(-534,12),

0556 => to\_signed(-546,12),

0557 => to\_signed(-558,12),

0558 => to\_signed(-570,12),

0559 => to\_signed(-582,12),

0560 => to\_signed(-594,12),

0561 => to\_signed(-606,12),

0562 => to\_signed(-618,12),

0563 => to\_signed(-630,12),

0564 => to\_signed(-642,12),

0565 => to\_signed(-654,12),

0566 => to\_signed(-666,12),

0567 => to\_signed(-678,12),

0568 => to\_signed(-689,12),

0569 => to\_signed(-701,12),

0570 => to\_signed(-713,12),

0571 => to\_signed(-725,12),

0572 => to\_signed(-737,12),

0573 => to\_signed(-748,12),

0574 => to\_signed(-760,12),

0575 => to\_signed(-772,12),

0576 => to\_signed(-783,12),

0577 => to\_signed(-795,12),

0578 => to\_signed(-806,12),

0579 => to\_signed(-818,12),

0580 => to\_signed(-829,12),

0581 => to\_signed(-841,12),

0582 => to\_signed(-852,12),

0583 => to\_signed(-864,12),

0584 => to\_signed(-875,12),

0585 => to\_signed(-886,12),

0586 => to\_signed(-898,12),

0587 => to\_signed(-909,12),

0588 => to\_signed(-920,12),

0589 => to\_signed(-932,12),

0590 => to\_signed(-943,12),

0591 => to\_signed(-954,12),

0592 => to\_signed(-965,12),

0593 => to\_signed(-976,12),

0594 => to\_signed(-987,12),

0595 => to\_signed(-998,12),

0596 => to\_signed(-1009,12),

0597 => to\_signed(-1020,12),

0598 => to\_signed(-1031,12),

0599 => to\_signed(-1042,12),

0600 => to\_signed(-1052,12),

0601 => to\_signed(-1063,12),

0602 => to\_signed(-1074,12),

0603 => to\_signed(-1085,12),

0604 => to\_signed(-1095,12),

0605 => to\_signed(-1106,12),

0606 => to\_signed(-1116,12),

0607 => to\_signed(-1127,12),

0608 => to\_signed(-1137,12),

0609 => to\_signed(-1148,12),

0610 => to\_signed(-1158,12),

0611 => to\_signed(-1168,12),

0612 => to\_signed(-1179,12),

0613 => to\_signed(-1189,12),

0614 => to\_signed(-1199,12),

0615 => to\_signed(-1209,12),

0616 => to\_signed(-1219,12),

0617 => to\_signed(-1230,12),

0618 => to\_signed(-1240,12),

0619 => to\_signed(-1250,12),

0620 => to\_signed(-1259,12),

0621 => to\_signed(-1269,12),

0622 => to\_signed(-1279,12),

0623 => to\_signed(-1289,12),

0624 => to\_signed(-1299,12),

0625 => to\_signed(-1308,12),

0626 => to\_signed(-1318,12),

0627 => to\_signed(-1328,12),

0628 => to\_signed(-1337,12),

0629 => to\_signed(-1347,12),

0630 => to\_signed(-1356,12),

0631 => to\_signed(-1366,12),

0632 => to\_signed(-1375,12),

0633 => to\_signed(-1384,12),

0634 => to\_signed(-1393,12),

0635 => to\_signed(-1403,12),

0636 => to\_signed(-1412,12),

0637 => to\_signed(-1421,12),

0638 => to\_signed(-1430,12),

0639 => to\_signed(-1439,12),

0640 => to\_signed(-1448,12),

0641 => to\_signed(-1457,12),

0642 => to\_signed(-1465,12),

0643 => to\_signed(-1474,12),

0644 => to\_signed(-1483,12),

0645 => to\_signed(-1491,12),

0646 => to\_signed(-1500,12),

0647 => to\_signed(-1509,12),

0648 => to\_signed(-1517,12),

0649 => to\_signed(-1525,12),

0650 => to\_signed(-1534,12),

0651 => to\_signed(-1542,12),

0652 => to\_signed(-1550,12),

0653 => to\_signed(-1558,12),

0654 => to\_signed(-1567,12),

0655 => to\_signed(-1575,12),

0656 => to\_signed(-1583,12),

0657 => to\_signed(-1591,12),

0658 => to\_signed(-1598,12),

0659 => to\_signed(-1606,12),

0660 => to\_signed(-1614,12),

0661 => to\_signed(-1622,12),

0662 => to\_signed(-1629,12),

0663 => to\_signed(-1637,12),

0664 => to\_signed(-1644,12),

0665 => to\_signed(-1652,12),

0666 => to\_signed(-1659,12),

0667 => to\_signed(-1667,12),

0668 => to\_signed(-1674,12),

0669 => to\_signed(-1681,12),

0670 => to\_signed(-1688,12),

0671 => to\_signed(-1695,12),

0672 => to\_signed(-1702,12),

0673 => to\_signed(-1709,12),

0674 => to\_signed(-1716,12),

0675 => to\_signed(-1723,12),

0676 => to\_signed(-1730,12),

0677 => to\_signed(-1736,12),

0678 => to\_signed(-1743,12),

0679 => to\_signed(-1750,12),

0680 => to\_signed(-1756,12),

0681 => to\_signed(-1763,12),

0682 => to\_signed(-1769,12),

0683 => to\_signed(-1775,12),

0684 => to\_signed(-1781,12),

0685 => to\_signed(-1788,12),

0686 => to\_signed(-1794,12),

0687 => to\_signed(-1800,12),

0688 => to\_signed(-1806,12),

0689 => to\_signed(-1812,12),

0690 => to\_signed(-1817,12),

0691 => to\_signed(-1823,12),

0692 => to\_signed(-1829,12),

0693 => to\_signed(-1834,12),

0694 => to\_signed(-1840,12),

0695 => to\_signed(-1845,12),

0696 => to\_signed(-1851,12),

0697 => to\_signed(-1856,12),

0698 => to\_signed(-1861,12),

0699 => to\_signed(-1867,12),

0700 => to\_signed(-1872,12),

0701 => to\_signed(-1877,12),

0702 => to\_signed(-1882,12),

0703 => to\_signed(-1887,12),

0704 => to\_signed(-1892,12),

0705 => to\_signed(-1896,12),

0706 => to\_signed(-1901,12),

0707 => to\_signed(-1906,12),

0708 => to\_signed(-1910,12),

0709 => to\_signed(-1915,12),

0710 => to\_signed(-1919,12),

0711 => to\_signed(-1924,12),

0712 => to\_signed(-1928,12),

0713 => to\_signed(-1932,12),

0714 => to\_signed(-1936,12),

0715 => to\_signed(-1940,12),

0716 => to\_signed(-1944,12),

0717 => to\_signed(-1948,12),

0718 => to\_signed(-1952,12),

0719 => to\_signed(-1956,12),

0720 => to\_signed(-1959,12),

0721 => to\_signed(-1963,12),

0722 => to\_signed(-1966,12),

0723 => to\_signed(-1970,12),

0724 => to\_signed(-1973,12),

0725 => to\_signed(-1977,12),

0726 => to\_signed(-1980,12),

0727 => to\_signed(-1983,12),

0728 => to\_signed(-1986,12),

0729 => to\_signed(-1989,12),

0730 => to\_signed(-1992,12),

0731 => to\_signed(-1995,12),

0732 => to\_signed(-1998,12),

0733 => to\_signed(-2000,12),

0734 => to\_signed(-2003,12),

0735 => to\_signed(-2006,12),

0736 => to\_signed(-2008,12),

0737 => to\_signed(-2011,12),

0738 => to\_signed(-2013,12),

0739 => to\_signed(-2015,12),

0740 => to\_signed(-2017,12),

0741 => to\_signed(-2019,12),

0742 => to\_signed(-2021,12),

0743 => to\_signed(-2023,12),

0744 => to\_signed(-2025,12),

0745 => to\_signed(-2027,12),

0746 => to\_signed(-2029,12),

0747 => to\_signed(-2031,12),

0748 => to\_signed(-2032,12),

0749 => to\_signed(-2034,12),

0750 => to\_signed(-2035,12),

0751 => to\_signed(-2036,12),

0752 => to\_signed(-2038,12),

0753 => to\_signed(-2039,12),

0754 => to\_signed(-2040,12),

0755 => to\_signed(-2041,12),

0756 => to\_signed(-2042,12),

0757 => to\_signed(-2043,12),

0758 => to\_signed(-2044,12),

0759 => to\_signed(-2044,12),

0760 => to\_signed(-2045,12),

0761 => to\_signed(-2046,12),

0762 => to\_signed(-2046,12),

0763 => to\_signed(-2047,12),

0764 => to\_signed(-2047,12),

0765 => to\_signed(-2047,12),

0766 => to\_signed(-2047,12),

0767 => to\_signed(-2047,12),

0768 => to\_signed(-2047,12),

0769 => to\_signed(-2047,12),

0770 => to\_signed(-2047,12),

0771 => to\_signed(-2047,12),

0772 => to\_signed(-2047,12),

0773 => to\_signed(-2047,12),

0774 => to\_signed(-2046,12),

0775 => to\_signed(-2046,12),

0776 => to\_signed(-2045,12),

0777 => to\_signed(-2044,12),

0778 => to\_signed(-2044,12),

0779 => to\_signed(-2043,12),

0780 => to\_signed(-2042,12),

0781 => to\_signed(-2041,12),

0782 => to\_signed(-2040,12),

0783 => to\_signed(-2039,12),

0784 => to\_signed(-2038,12),

0785 => to\_signed(-2036,12),

0786 => to\_signed(-2035,12),

0787 => to\_signed(-2034,12),

0788 => to\_signed(-2032,12),

0789 => to\_signed(-2031,12),

0790 => to\_signed(-2029,12),

0791 => to\_signed(-2027,12),

0792 => to\_signed(-2025,12),

0793 => to\_signed(-2023,12),

0794 => to\_signed(-2021,12),

0795 => to\_signed(-2019,12),

0796 => to\_signed(-2017,12),

0797 => to\_signed(-2015,12),

0798 => to\_signed(-2013,12),

0799 => to\_signed(-2011,12),

0800 => to\_signed(-2008,12),

0801 => to\_signed(-2006,12),

0802 => to\_signed(-2003,12),

0803 => to\_signed(-2000,12),

0804 => to\_signed(-1998,12),

0805 => to\_signed(-1995,12),

0806 => to\_signed(-1992,12),

0807 => to\_signed(-1989,12),

0808 => to\_signed(-1986,12),

0809 => to\_signed(-1983,12),

0810 => to\_signed(-1980,12),

0811 => to\_signed(-1977,12),

0812 => to\_signed(-1973,12),

0813 => to\_signed(-1970,12),

0814 => to\_signed(-1966,12),

0815 => to\_signed(-1963,12),

0816 => to\_signed(-1959,12),

0817 => to\_signed(-1956,12),

0818 => to\_signed(-1952,12),

0819 => to\_signed(-1948,12),

0820 => to\_signed(-1944,12),

0821 => to\_signed(-1940,12),

0822 => to\_signed(-1936,12),

0823 => to\_signed(-1932,12),

0824 => to\_signed(-1928,12),

0825 => to\_signed(-1924,12),

0826 => to\_signed(-1919,12),

0827 => to\_signed(-1915,12),

0828 => to\_signed(-1910,12),

0829 => to\_signed(-1906,12),

0830 => to\_signed(-1901,12),

0831 => to\_signed(-1896,12),

0832 => to\_signed(-1892,12),

0833 => to\_signed(-1887,12),

0834 => to\_signed(-1882,12),

0835 => to\_signed(-1877,12),

0836 => to\_signed(-1872,12),

0837 => to\_signed(-1867,12),

0838 => to\_signed(-1861,12),

0839 => to\_signed(-1856,12),

0840 => to\_signed(-1851,12),

0841 => to\_signed(-1845,12),

0842 => to\_signed(-1840,12),

0843 => to\_signed(-1834,12),

0844 => to\_signed(-1829,12),

0845 => to\_signed(-1823,12),

0846 => to\_signed(-1817,12),

0847 => to\_signed(-1812,12),

0848 => to\_signed(-1806,12),

0849 => to\_signed(-1800,12),

0850 => to\_signed(-1794,12),

0851 => to\_signed(-1788,12),

0852 => to\_signed(-1781,12),

0853 => to\_signed(-1775,12),

0854 => to\_signed(-1769,12),

0855 => to\_signed(-1763,12),

0856 => to\_signed(-1756,12),

0857 => to\_signed(-1750,12),

0858 => to\_signed(-1743,12),

0859 => to\_signed(-1736,12),

0860 => to\_signed(-1730,12),

0861 => to\_signed(-1723,12),

0862 => to\_signed(-1716,12),

0863 => to\_signed(-1709,12),

0864 => to\_signed(-1702,12),

0865 => to\_signed(-1695,12),

0866 => to\_signed(-1688,12),

0867 => to\_signed(-1681,12),

0868 => to\_signed(-1674,12),

0869 => to\_signed(-1667,12),

0870 => to\_signed(-1659,12),

0871 => to\_signed(-1652,12),

0872 => to\_signed(-1644,12),

0873 => to\_signed(-1637,12),

0874 => to\_signed(-1629,12),

0875 => to\_signed(-1622,12),

0876 => to\_signed(-1614,12),

0877 => to\_signed(-1606,12),

0878 => to\_signed(-1598,12),

0879 => to\_signed(-1591,12),

0880 => to\_signed(-1583,12),

0881 => to\_signed(-1575,12),

0882 => to\_signed(-1567,12),

0883 => to\_signed(-1558,12),

0884 => to\_signed(-1550,12),

0885 => to\_signed(-1542,12),

0886 => to\_signed(-1534,12),

0887 => to\_signed(-1525,12),

0888 => to\_signed(-1517,12),

0889 => to\_signed(-1509,12),

0890 => to\_signed(-1500,12),

0891 => to\_signed(-1491,12),

0892 => to\_signed(-1483,12),

0893 => to\_signed(-1474,12),

0894 => to\_signed(-1465,12),

0895 => to\_signed(-1457,12),

0896 => to\_signed(-1448,12),

0897 => to\_signed(-1439,12),

0898 => to\_signed(-1430,12),

0899 => to\_signed(-1421,12),

0900 => to\_signed(-1412,12),

0901 => to\_signed(-1403,12),

0902 => to\_signed(-1393,12),

0903 => to\_signed(-1384,12),

0904 => to\_signed(-1375,12),

0905 => to\_signed(-1366,12),

0906 => to\_signed(-1356,12),

0907 => to\_signed(-1347,12),

0908 => to\_signed(-1337,12),

0909 => to\_signed(-1328,12),

0910 => to\_signed(-1318,12),

0911 => to\_signed(-1308,12),

0912 => to\_signed(-1299,12),

0913 => to\_signed(-1289,12),

0914 => to\_signed(-1279,12),

0915 => to\_signed(-1269,12),

0916 => to\_signed(-1259,12),

0917 => to\_signed(-1250,12),

0918 => to\_signed(-1240,12),

0919 => to\_signed(-1230,12),

0920 => to\_signed(-1219,12),

0921 => to\_signed(-1209,12),

0922 => to\_signed(-1199,12),

0923 => to\_signed(-1189,12),

0924 => to\_signed(-1179,12),

0925 => to\_signed(-1168,12),

0926 => to\_signed(-1158,12),

0927 => to\_signed(-1148,12),

0928 => to\_signed(-1137,12),

0929 => to\_signed(-1127,12),

0930 => to\_signed(-1116,12),

0931 => to\_signed(-1106,12),

0932 => to\_signed(-1095,12),

0933 => to\_signed(-1085,12),

0934 => to\_signed(-1074,12),

0935 => to\_signed(-1063,12),

0936 => to\_signed(-1052,12),

0937 => to\_signed(-1042,12),

0938 => to\_signed(-1031,12),

0939 => to\_signed(-1020,12),

0940 => to\_signed(-1009,12),

0941 => to\_signed(-998,12),

0942 => to\_signed(-987,12),

0943 => to\_signed(-976,12),

0944 => to\_signed(-965,12),

0945 => to\_signed(-954,12),

0946 => to\_signed(-943,12),

0947 => to\_signed(-932,12),

0948 => to\_signed(-920,12),

0949 => to\_signed(-909,12),

0950 => to\_signed(-898,12),

0951 => to\_signed(-886,12),

0952 => to\_signed(-875,12),

0953 => to\_signed(-864,12),

0954 => to\_signed(-852,12),

0955 => to\_signed(-841,12),

0956 => to\_signed(-829,12),

0957 => to\_signed(-818,12),

0958 => to\_signed(-806,12),

0959 => to\_signed(-795,12),

0960 => to\_signed(-783,12),

0961 => to\_signed(-772,12),

0962 => to\_signed(-760,12),

0963 => to\_signed(-748,12),

0964 => to\_signed(-737,12),

0965 => to\_signed(-725,12),

0966 => to\_signed(-713,12),

0967 => to\_signed(-701,12),

0968 => to\_signed(-689,12),

0969 => to\_signed(-678,12),

0970 => to\_signed(-666,12),

0971 => to\_signed(-654,12),

0972 => to\_signed(-642,12),

0973 => to\_signed(-630,12),

0974 => to\_signed(-618,12),

0975 => to\_signed(-606,12),

0976 => to\_signed(-594,12),

0977 => to\_signed(-582,12),

0978 => to\_signed(-570,12),

0979 => to\_signed(-558,12),

0980 => to\_signed(-546,12),

0981 => to\_signed(-534,12),

0982 => to\_signed(-521,12),

0983 => to\_signed(-509,12),

0984 => to\_signed(-497,12),

0985 => to\_signed(-485,12),

0986 => to\_signed(-473,12),

0987 => to\_signed(-460,12),

0988 => to\_signed(-448,12),

0989 => to\_signed(-436,12),

0990 => to\_signed(-424,12),

0991 => to\_signed(-411,12),

0992 => to\_signed(-399,12),

0993 => to\_signed(-387,12),

0994 => to\_signed(-374,12),

0995 => to\_signed(-362,12),

0996 => to\_signed(-350,12),

0997 => to\_signed(-337,12),

0998 => to\_signed(-325,12),

0999 => to\_signed(-312,12),

1000 => to\_signed(-300,12),

1001 => to\_signed(-288,12),

1002 => to\_signed(-275,12),

1003 => to\_signed(-263,12),

1004 => to\_signed(-250,12),

1005 => to\_signed(-238,12),

1006 => to\_signed(-225,12),

1007 => to\_signed(-213,12),

1008 => to\_signed(-200,12),

1009 => to\_signed(-188,12),

1010 => to\_signed(-175,12),

1011 => to\_signed(-163,12),

1012 => to\_signed(-150,12),

1013 => to\_signed(-138,12),

1014 => to\_signed(-125,12),

1015 => to\_signed(-113,12),

1016 => to\_signed(-100,12),

1017 => to\_signed(-87,12),

1018 => to\_signed(-75,12),

1019 => to\_signed(-62,12),

1020 => to\_signed(-50,12),

1021 => to\_signed(-37,12),

1022 => to\_signed(-25,12),

1023 => to\_signed(-12,12)

);

attribute ramstyle : string;

attribute ramstyle of mem : constant is "M4k";

begin

LookupTable : process (clk)

begin

if(rising\_edge(clk)) then

seno <= mem(to\_integer(angulo));

coseno <= mem(to\_integer(angulo+to\_unsigned(256,10)));

end if;

end process;

end beh;

# Apéndice H

Archivo rotar.vhf

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use work.HexaPackage.all;

entity rotar is

port(

clk : in std\_logic;

Actualizar : in std\_logic;

X\_in : in signed(9 downto 0);

Y\_in : in signed(9 downto 0);

angulo : in unsigned(9 downto 0);

X\_out : out signed(9 downto 0);

Y\_out : out signed(9 downto 0)

);

end rotar;

architecture beh of rotar is

signal seno : signed(11 downto 0);

signal coseno : signed(11 downto 0);

signal X\_aux : signed (21 downto 0);

signal Y\_aux : signed (21 downto 0);

begin

SC:senocoseno port map(clk=>clk,angulo=>angulo,seno=>seno,coseno=>coseno);

X\_aux <= X\_in\*coseno - Y\_in\*seno;

Y\_aux <= X\_in\*seno + Y\_in\*coseno;

Actualiza\_XY : process(clk)

begin

if (rising\_edge(clk)) then

x\_out <= X\_aux(21 downto 12);

y\_out <= y\_aux(21 downto 12);

end if;

end process;

end beh;

# Apéndice I

Archivo PRUEBA\_PLAYER.vhd

--PRUEBA DEL BLOQUE DIBUJO SÓLO.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use work.HexaPackage.all;

ENTITY PRUEBA\_PLAYER IS

PORT(

CLK25MHZ : IN STD\_LOGIC;

CLR : IN STD\_LOGIC;

DIF\_SWITCH : IN STD\_LOGIC\_VECTOR(1 DOWNTO 0);

Nuevo\_juego : in std\_logic;

Boton\_der : in std\_logic;

Boton\_izq : in std\_logic;

CLK\_VGA : OUT STD\_LOGIC;

HSYNC : OUT STD\_LOGIC;

VSYNC : OUT STD\_LOGIC;

VGA\_SYNC\_N : OUT STD\_LOGIC;

VGA\_BLANK\_N : OUT STD\_LOGIC;

ROJO,VERDE,AZUL : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0);

velocidad\_prueba : out unsigned(5 DOWNTO 0);

idle\_led : out std\_logic

);

END ENTITY;

ARCHITECTURE BEH OF PRUEBA\_PLAYER IS

SIGNAL RED\_OUT,GREEN\_OUT,BLUE\_OUT : STD\_LOGIC\_VECTOR(3 DOWNTO 0);

SIGNAL RED,GREEN,BLUE : STD\_LOGIC\_VECTOR(3 DOWNTO 0);

SIGNAL X,Y : UNSIGNED(9 DOWNTO 0);

SIGNAL XC,YC : SIGNED(9 downto 0);

SIGNAL XCR,YCR : SIGNED(9 downto 0);

SIGNAL END\_FRAME,PRE\_FRAME : STD\_LOGIC;

SIGNAL GAME\_OVER : STD\_LOGIC;

SIGNAL W\_DIFICULTAD : STD\_LOGIC\_VECTOR(1 DOWNTO 0);

SIGNAL Vuelve\_jugador,Actualizar : std\_logic;

SIGNAL angulo : unsigned(9 downto 0);

SIGNAL RESET : STD\_LOGIC;

SIGNAL RAND : STD\_LOGIC\_VECTOR(63 DOWNTO 0);

SIGNAL Subir\_velocidad : STD\_LOGIC;

signal XP\_actual : signed(9 downto 0);

signal YP\_actual : signed(9 downto 0);

signal XP\_ant : signed(9 downto 0);

signal YP\_ant : signed(9 downto 0);

signal cuadrante : unsigned (2 downto 0);

signal cuadranteAux : STD\_LOGIC\_VECTOR (2 downto 0);

signal radio : unsigned (9 downto 0);

signal cuadranteR : unsigned (2 downto 0);

signal radioR : unsigned (9 downto 0);

signal MANTENER : STD\_LOGIC;

signal ALTERNA : std\_logic;

SIGNAL DIBUJA\_PLAYER : STD\_LOGIC;

SIGNAL DIBUJA\_PAREDES : STD\_LOGIC;

SIGNAL DIBUJA\_ISLA : STD\_LOGIC;

SIGNAL DIRECCION : STD\_LOGIC\_VECTOR(5 DOWNTO 0);

SIGNAL DATOS : STD\_LOGIC\_VECTOR(5 DOWNTO 0);

SIGNAL CVISIBLE : STD\_LOGIC;

SIGNAL CNUEVO : STD\_LOGIC;

SIGNAL COMP1,COMP2 : STD\_LOGIC;

BEGIN

ROJO <= RED\_OUT & "0000";

VERDE<= GREEN\_OUT & "0000";

AZUL <= BLUE\_OUT & "0000";

CLK\_VGA<=CLK25MHZ;

DRIVER\_VGA: CONTROLADOR\_VGA

PORT MAP(

CLK =>CLK25MHZ,

CLR =>CLR,

RED\_IN =>RED,

GREEN\_IN =>GREEN,

BLUE\_IN =>BLUE,

HSYNC =>HSYNC,

VSYNC =>VSYNC,

VGA\_SYNC\_N =>VGA\_SYNC\_N,

VGA\_BLANK\_N =>VGA\_BLANK\_N,

X =>X,

Y =>Y,

RED =>RED\_OUT,

GREEN =>GREEN\_OUT,

BLUE =>BLUE\_OUT,

END\_FRAME =>END\_FRAME,

PRE\_FRAME =>PRE\_FRAME

);

Juego\_Fsm1: Juego\_Fsm port map(

Clk =>CLK25MHZ,

Frame\_inicio =>PRE\_FRAME,

Frame\_fin =>END\_FRAME,

Nuevo\_juego =>Nuevo\_juego,

Colision =>CNUEVO, --REVISAR

Reset =>CLR,

Actualizar =>Actualizar,

Rst =>RESET,

idle\_led =>idle\_led,

Game\_over =>GAME\_OVER

);

CoordenadasCentro: DesplazaXYalCentro port map(

Clk =>CLK25MHZ,

Xin =>X,

Yin =>Y,

Xout =>XC,

Yout =>YC

);

Rotar1: rotar port map(

clk =>CLK25MHZ,

Actualizar =>Actualizar,

X\_in =>XC,

Y\_in =>YC,

angulo =>angulo,

X\_out =>XCR,

Y\_out =>YCR

);

ContSpeedUp: CONTADOR generic map(600)

port map(

Clk =>CLK25MHZ,

EN =>Actualizar,

Rst =>RESET,

ovf =>Subir\_velocidad

);

ControlGiro1:control\_giro port map(

clk =>CLK25MHZ,

reset =>RESET,

Actualizar =>Actualizar,

Random =>RAND,

Subir\_velocidad =>Subir\_velocidad,

velocidad\_prueba =>velocidad\_prueba,

angulo =>angulo

);

Player: jugador port map(

Clk =>CLK25MHZ,

Actualizar =>Actualizar,

Vuelve\_jugador =>Vuelve\_jugador,

Dificultad =>W\_DIFICULTAD,

Boton\_der =>Boton\_der,

Boton\_izq =>Boton\_izq,

X\_actual =>XP\_actual,

Y\_actual =>YP\_actual,

X\_anterior =>XP\_ant,

Y\_anterior =>YP\_ant

);

Circle: CIRCULO PORT MAP(

CLK => CLK25MHZ,

X1 => XCR,

Y1 => YCR,

X2 => XP\_actual,

Y2 => YP\_actual,

DIBUJA => DIBUJA\_PLAYER

);

Gen\_paredes: GENERADOR\_PAREDES PORT MAP(

CLK =>CLK25MHZ,

RAND =>RAND,

MANTIENE =>MANTENER,

ADDR =>DIRECCION,

DATA =>DATOS

);

Pared: PAREDES PORT MAP(

CLK =>CLK25MHZ,

RESET =>Reset,

ACTUALIZAR =>Actualizar,

DIFICULTAD =>W\_DIFICULTAD,

DATA =>DATOS,

CUADRANTE =>cuadranteR,

RADIO =>radioR,

MANTIENE =>MANTENER,

DATA\_ADDR =>DIRECCION,

CUADRANTE\_OUT =>cuadranteAux,

VISIBLE =>DIBUJA\_PAREDES,

ISLA =>DIBUJA\_ISLA

);

Draw: DIBUJO PORT MAP(

CLK =>CLK25MHZ,

ALTERNA =>ALTERNA,

CCOLOR =>subir\_velocidad,

DPAREDES =>DIBUJA\_PAREDES,

DISLA =>DIBUJA\_ISLA,

DJUGADOR =>DIBUJA\_PLAYER,

CUADRANTE=>cuadranteAux,

RAND =>RAND,

ROJO =>RED,

VERDE =>GREEN,

AZUL =>BLUE

);

GeneradorAleatorio:LFSR\_64 port map(

Clk => CLK25MHZ,

Set => CLR,

En => '1',

b => RAND

);

Rect2Hex1: conv\_hexagonal port map(

clk =>CLK25MHZ,

x =>XC,

y =>YC,

cuadrante =>cuadrante,

radio => radio

);

Rect2Hex2: conv\_hexagonal port map(

clk =>CLK25MHZ,

x =>XCR,

y =>YCR,

cuadrante =>cuadranteR,

radio =>radioR

);

ContAlterna: CONTADOR generic map(60)

port map(

Clk =>CLK25MHZ,

EN =>Actualizar,

Rst =>RESET,

ovf =>ALTERNA

);

Dif: Dificultad port map(

Clk =>CLK25MHZ,

switches =>DIF\_SWITCH,

gameover =>GAME\_OVER,

Dificultad =>W\_DIFICULTAD,

Leds\_Rojo =>OPEN,

Leds\_Verde =>OPEN

);

--Comp\_cords1: Compara\_coordenadas port map(

-- --inputs

-- Xin1 =>XC,

-- Yin1 =>YC,

-- Xin2 =>XP\_actual,

-- Yin2 =>YP\_actual,

--

-- --outputs

-- Comparacion =>COMP1

-- );

--

--Comp\_cords2: Compara\_coordenadas port map(

-- --inputs

-- Xin1 =>XC,

-- Yin1 =>YC,

-- Xin2 =>XP\_ant,

-- Yin2 =>YP\_ant,

--

-- --outputs

-- Comparacion =>COMP2

-- );

FF\_AND1: FF\_AND port map(

Clk =>CLK25MHZ,

IN1 =>DIBUJA\_PLAYER,

IN2 =>DIBUJA\_PAREDES,

OUT1 =>CNUEVO

);

--FF\_AND2: FF\_AND port map(

-- Clk =>CLK25MHZ,

-- IN1 =>DIBUJA\_PLAYER,

-- IN2 =>DIBUJA\_PAREDES,

--

-- OUT1 =>open

-- );

--

END ARCHITECTURE;

# Apéndice J

Archivo paredes.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use work.HexaPackage.all;

-- carga las paredes desde generador de paredes, las mueve y las dibuja

-- tambien dibuja la isla

entity paredes IS

port(

clk : in std\_logic;

reset : in std\_logic;

actualizar : in std\_logic;

dificultad : in std\_logic\_vector(1 downto 0);

data : in std\_logic\_vector(5 downto 0);

cuadrante : in unsigned(2 downto 0);

radio : in unsigned(9 downto 0);

mantiene : out std\_logic;

data\_addr : out std\_logic\_vector(5 downto 0);

cuadrante\_out : out std\_logic\_vector(2 downto 0);

visible : out std\_logic;

isla : out std\_logic

);

end entity;

architecture beh of paredes is

type t\_2d\_a is array(0 to 127) of std\_logic\_vector(5 downto 0);

signal paredes\_a : t\_2d\_a; -- arreglo donde se cargan 8 grupos de paredes

signal centro : unsigned (13 downto 0); -- indica el centro de la pantalla (7,7)

signal offset : unsigned (11 downto 0); -- direccion de paredes\_a que tiene que dibujar (7,5)

signal puntero : unsigned (6 downto 0); -- puntero que pasa los datos desde generador de paredes y los pone en paredes\_a

signal mantiene\_aux : std\_logic;

begin

offset <= centro (13 downto 2) + resize(radio,12); -- distancia de la coordenada al centro de la pantalla

-- offset (7,5) = centro (7,5) + radio (5,5)

-- offset = centro + radio/32

--------------------------------------------------------------------------------

-- carga el valor inicial del reset y lo mueve dependiendo de la dificultad

mueve\_centro: process(clk, reset, dificultad, actualizar)

begin

if rising\_edge(clk) then

if (reset='1') then

centro <= to\_unsigned(6144,14); -- 96,0 (mitad de la segunda parte)

elsif (actualizar = '1') then

case dificultad is

when "00" => centro <= centro + to\_unsigned(6,14); -- 0,09375

when "01" => centro <= centro + to\_unsigned(9,14); -- 0,140625

when "10" => centro <= centro + to\_unsigned(12,14); -- 0,1875

when "11" => centro <= centro + to\_unsigned(15,14); -- 0,234375

end case;

end if;

end if;

end process;

--------------------------------------------------------------------------------

-- activa visible cuando debe dibujar una paredes

activa\_visible: process(clk, cuadrante, radio, offset)

variable aux : std\_logic\_vector (5 downto 0);

begin

if rising\_edge(clk) then

cuadrante\_out <= std\_logic\_vector(cuadrante);

visible <= '0';

isla <= '0';

if (radio < to\_unsigned(28,10)) then -- hexagono negro en el centro

isla <= '1';

elsif (radio < to\_unsigned(32,10)) then -- pared del hexagono central

visible <= '1';

else

aux := paredes\_a ( to\_integer(offset(11 downto 5)));

visible <= aux (to\_integer(cuadrante));

end if;

end if;

end process;

--------------------------------------------------------------------------------

-- carga del arreglo de paredes

data\_addr <= std\_logic\_vector(puntero(5 downto 0));

mantiene <= mantiene\_aux; -- mantiene la parte estocastica cuando

mantiene\_aux <= '1' when ((puntero (5 downto 0) /= to\_unsigned(0,6)) or -- esta en medio de pasar 4 grupos

(puntero = to\_unsigned(0,7) and bit2bool(centro(13))) or -- puntero esta al principio y centro en la segunda mitad

(puntero = to\_unsigned(64,7) and not(bit2bool(centro(13))))) -- puntero esta a la mitad y centro esta en la primer parte

else '0';

-- carga todo el arreglo con 0 al resetear y despues carga cada mitad del arreglo

carga\_paredes\_a: process (clk, reset, data)

begin

if rising\_edge(clk) then

if (reset = '1') then

puntero <= to\_unsigned(64,7);

for i in 0 to 127 loop

paredes\_a(i) <= std\_logic\_vector(to\_unsigned(0,6));

end loop;

elsif (mantiene\_aux = '1') then

paredes\_a(to\_integer(puntero))<= data;

puntero <= puntero + 1;

end if;

end if;

end process;

end architecture;

# Apéndice K

Archivo LFSR\_64.vhd

library ieee;

use ieee.std\_logic\_1164.all;

--CONTADOR LINEAR FEEDBACK SHIFT REGISTER.

entity LFSR\_64 is

port

(

-- Input ports

Clk : in std\_logic; --Entrada de reloj

Set : in std\_logic; --Entrada de seteo

En : in std\_logic;

-- Output ports

b : out std\_logic\_vector(63 downto 0)

);

end LFSR\_64;

architecture shift of LFSR\_64 is

component FF\_D\_RISING is

port

(

-- Input ports

D : in std\_logic;

Clk : in std\_logic; --Reloj

Set : in std\_logic; --Seteo asincrónico

Reset : in std\_logic;

En : in std\_logic;

-- Output ports

Q : out std\_logic

);

end component;

signal XOR1: std\_logic; --Salida de XOR

signal int: std\_logic\_vector(63 downto 0); --Conexiones internas del contador, sirven como salidas también.

begin

XOR1<=int(63) XOR int(62) xor int(60) XOR int(59);

LFSR: for i in 63 downto 0 generate

i64: if (i<64 and i>0) generate

bit32: FF\_D\_RISING port map(D=>int(i-1), Clk=>Clk, Set=>Set,Q=>int(i),reset=>'0',En=>En); --demas bits de salida

end generate i64;

i0: if(i=0) generate

bit0: FF\_D\_RISING port map(D=>XOR1,Clk=>Clk,Set=>Set,Q=>int(i),reset=>'0',En=>En); --Primer bit

end generate i0;

end generate LFSR;

b<=int;

end shift;

# Archivo L

Archivo jugador.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use work.HexaPackage.all;

entity jugador is

port(

Clk : in std\_logic;

Actualizar : in std\_logic;

Vuelve\_jugador : in std\_logic;

Dificultad : in std\_logic\_vector(1 downto 0);

Boton\_der : in std\_logic;

Boton\_izq : in std\_logic;

X\_actual : out signed(9 downto 0);

Y\_actual : out signed(9 downto 0);

X\_anterior : out signed(9 downto 0);

Y\_anterior : out signed(9 downto 0)

);

end jugador;

architecture beh of jugador is

signal angulo : unsigned(9 downto 0);

signal seno : signed(11 downto 0);

signal coseno : signed(11 downto 0);

signal velocidad : unsigned(9 downto 0);

signal angulo\_anterior : unsigned(9 downto 0);

signal x\_actual\_aux, y\_actual\_aux : signed(9 downto 0);

begin

SC:senocoseno port map(clk=>clk,angulo=>angulo,seno=>seno,coseno=>coseno);

Velocidad\_jugador: process (clk,dificultad)

begin

if rising\_edge(clk) then

case dificultad is

when "00" =>

velocidad <= to\_unsigned(12,10);

when "01" =>

velocidad <= to\_unsigned(14,10);

when "10" =>

velocidad <= to\_unsigned(16,10);

when "11" =>

velocidad <= to\_unsigned(18,10);

when others =>

velocidad <= to\_unsigned(12,10);

end case;

end if;

end process;

Actualizar\_posicion: process (clk,boton\_der,boton\_izq,vuelve\_jugador,actualizar)

begin

if rising\_edge(clk) then

if (actualizar='1') then

X\_anterior <= X\_actual\_aux;

Y\_anterior <= Y\_actual\_aux;

angulo\_anterior <= angulo;

if(boton\_der='1') then

angulo <= angulo + velocidad;

elsif(boton\_izq='1') then

angulo <= angulo - velocidad;

elsif(vuelve\_jugador='1') then

angulo <= angulo\_anterior;

end if;

end if;

end if;

end process;

Posicion\_jugador: process(clk)

begin

if(rising\_edge(clk)) then

x\_actual\_aux <= signed((11 DOWNTO 8 => coseno(11))&coseno(11 downto 6)) + signed((11 DOWNTO 7 => coseno(11))&coseno(11 downto 7));

y\_actual\_aux <= signed((11 DOWNTO 8 => seno(11))&seno(11 downto 6)) + signed((11 DOWNTO 7 => seno(11))&seno(11 downto 7));

end if;

end process;

x\_actual <= x\_actual\_aux;

y\_actual <= y\_actual\_aux;

end beh;

# Apéndice M

Archivo Juego\_Fsm.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity Juego\_Fsm is

port(

Clk : in std\_logic;

frame\_inicio : in std\_logic;

Frame\_fin : in std\_logic;

Nuevo\_juego : in std\_logic;

Colision : in std\_logic;

Reset : in std\_logic;

Actualizar : out std\_logic;

Rst : out std\_logic;

idle\_led : out std\_logic;

Game\_over : out std\_logic

);

end Juego\_Fsm;

architecture beh of Juego\_Fsm is

type FSM\_states is (IDLE, RESTART, JUGANDO, COLISIONDOBLE);

signal current\_state,next\_state:FSM\_states;

begin

PROXIMO\_ESTADO: process(current\_state, frame\_inicio, frame\_fin, clk, colision, nuevo\_juego)

begin

case current\_state is

when IDLE =>

if (nuevo\_juego='1' and frame\_inicio='1') then

next\_state <= RESTART;

else

next\_state <= IDLE;

end if;

when RESTART =>

if (frame\_fin='1') then

next\_state <= JUGANDO;

else

next\_state <= RESTART;

end if;

when JUGANDO =>

if (colision='1') then

next\_state <= COLISIONDOBLE;

else

next\_state <= JUGANDO;

end if;

when COLISIONDOBLE =>

if (frame\_fin='1') then

next\_state <= IDLE;

else

next\_state <= COLISIONDOBLE;

end if;

when others =>

next\_state <= IDLE;

end case;

end process;

ESTADO\_ACTUAL: process(clk,Reset)

begin

if(Reset='1') then

current\_state<=IDLE;

elsif(rising\_edge(Clk)) then

current\_state<=next\_state;

end if;

end process;

Salida: process(Reset,Current\_state,Clk,frame\_inicio,colision,frame\_fin)

begin

if (reset='1') then

Game\_over <= '1';

Actualizar <= '0';

rst <= '1';

idle\_led <= '0';

elsif (rising\_edge(clk)) then

Rst <= '0';

Game\_over <= '0';

Actualizar <= '0';

idle\_led <= '0';

case Current\_state is

when IDLE =>

Game\_over <= '0';

idle\_led <= '1';

when RESTART =>

Rst <= '1';

when JUGANDO =>

if (frame\_inicio='1') then

Actualizar <= '1';

end if;

when COLISIONDOBLE =>

Game\_over <= '1';

end case;

end if;

end process;

end beh;

# Apéndice N

Archivo GENERADOR\_PAREDES.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

ENTITY GENERADOR\_PAREDES IS

PORT(

CLK : IN STD\_LOGIC;

RAND : IN STD\_LOGIC\_VECTOR(63 DOWNTO 0);

MANTIENE : IN STD\_LOGIC;

ADDR : IN STD\_LOGIC\_VECTOR(5 DOWNTO 0);

DATA : OUT STD\_LOGIC\_VECTOR(5 DOWNTO 0)

);

END ENTITY;

ARCHITECTURE BEH OF GENERADOR\_PAREDES IS

TYPE RAND\_2D IS ARRAY(0 TO 63) OF STD\_LOGIC\_VECTOR(5 DOWNTO 0);

TYPE LABE\_2D IS ARRAY(0 TO 127) OF STD\_LOGIC\_VECTOR(5 DOWNTO 0);

SIGNAL RANDH : STD\_LOGIC\_VECTOR(63 DOWNTO 0);

SIGNAL PAREDES\_RAND : RAND\_2D;

SIGNAL PAREDES\_ALT : RAND\_2D;

SIGNAL PAREDES\_LABE : LABE\_2D;

BEGIN

MANTIENERAND:PROCESS(CLK,MANTIENE)

BEGIN

IF RISING\_EDGE(CLK) THEN

IF (MANTIENE='0') THEN

RANDH <= RAND;

END IF;

END IF;

END PROCESS;

PROCESS (ADDR, RANDH,PAREDES\_RAND,PAREDES\_ALT,PAREDES\_LABE)

VARIABLE INDICE: STD\_LOGIC\_VECTOR(6 DOWNTO 0);

BEGIN

CASE ADDR(5 DOWNTO 4) IS

WHEN "00" =>

INDICE:="0" & RANDH(1 DOWNTO 0) & ADDR(3 DOWNTO 0);

DATA <= PAREDES\_RAND(TO\_INTEGER(UNSIGNED(INDICE)));

WHEN "01" =>

INDICE:="0" & RANDH(3 DOWNTO 2) & ADDR(3 DOWNTO 0);

DATA <= PAREDES\_ALT(TO\_INTEGER(UNSIGNED(INDICE)));

WHEN "10" =>

INDICE:="0" & RANDH(5 DOWNTO 4) & ADDR(3 DOWNTO 0);

DATA <= PAREDES\_RAND(TO\_INTEGER(UNSIGNED(INDICE)));

WHEN "11" =>

INDICE:=RANDH(8 DOWNTO 6) & ADDR(3 DOWNTO 0);

DATA <= PAREDES\_LABE(TO\_INTEGER(UNSIGNED(INDICE)));

END CASE;

END PROCESS;

PAREDES\_RAND(0) <= "101010";

PAREDES\_RAND(1) <= "000000";

PAREDES\_RAND(2) <= "010101";

PAREDES\_RAND(3) <= "000000";

PAREDES\_RAND(4) <= "101010";

PAREDES\_RAND(5) <= "000000";

PAREDES\_RAND(6) <= "010101";

PAREDES\_RAND(7) <= "000000";

PAREDES\_RAND(8) <= "101010";

PAREDES\_RAND(9) <= "000000";

PAREDES\_RAND(10) <= "010101";

PAREDES\_RAND(11) <= "000000";

PAREDES\_RAND(12) <= "101010";

PAREDES\_RAND(13) <= "000000";

PAREDES\_RAND(14) <= "000000";

PAREDES\_RAND(15) <= "000000";

PAREDES\_RAND(16) <= "010101";

PAREDES\_RAND(17) <= "000000";

PAREDES\_RAND(18) <= "101010";

PAREDES\_RAND(19) <= "000000";

PAREDES\_RAND(20) <= "010101";

PAREDES\_RAND(21) <= "000000";

PAREDES\_RAND(22) <= "101010";

PAREDES\_RAND(23) <= "000000";

PAREDES\_RAND(24) <= "010101";

PAREDES\_RAND(25) <= "000000";

PAREDES\_RAND(26) <= "101010";

PAREDES\_RAND(27) <= "000000";

PAREDES\_RAND(28) <= "010101";

PAREDES\_RAND(29) <= "000000";

PAREDES\_RAND(30) <= "000000";

PAREDES\_RAND(31) <= "000000";

PAREDES\_RAND(32) <= "101010";

PAREDES\_RAND(33) <= "000000";

PAREDES\_RAND(34) <= "000000";

PAREDES\_RAND(35) <= "010101";

PAREDES\_RAND(36) <= "000000";

PAREDES\_RAND(37) <= "000000";

PAREDES\_RAND(38) <= "101110";

PAREDES\_RAND(39) <= "000000";

PAREDES\_RAND(40) <= "000000";

PAREDES\_RAND(41) <= "011011";

PAREDES\_RAND(42) <= "000000";

PAREDES\_RAND(43) <= "000000";

PAREDES\_RAND(44) <= "010101";

PAREDES\_RAND(45) <= "000000";

PAREDES\_RAND(46) <= "000000";

PAREDES\_RAND(47) <= "000000";

PAREDES\_RAND(48) <= "010101";

PAREDES\_RAND(49) <= "000000";

PAREDES\_RAND(50) <= "000000";

PAREDES\_RAND(51) <= "101010";

PAREDES\_RAND(52) <= "000000";

PAREDES\_RAND(53) <= "000000";

PAREDES\_RAND(54) <= "111010";

PAREDES\_RAND(55) <= "000000";

PAREDES\_RAND(56) <= "000000";

PAREDES\_RAND(57) <= "101101";

PAREDES\_RAND(58) <= "000000";

PAREDES\_RAND(59) <= "000000";

PAREDES\_RAND(60) <= "101010";

PAREDES\_RAND(61) <= "000000";

PAREDES\_RAND(62) <= "000000";

PAREDES\_RAND(63) <= "000000";

PAREDES\_ALT(0) <= "011111";

PAREDES\_ALT(1) <= "000000";

PAREDES\_ALT(2) <= "000000";

PAREDES\_ALT(3) <= "000000";

PAREDES\_ALT(4) <= "111011";

PAREDES\_ALT(5) <= "000000";

PAREDES\_ALT(6) <= "000000";

PAREDES\_ALT(7) <= "000000";

PAREDES\_ALT(8) <= "011111";

PAREDES\_ALT(9) <= "000000";

PAREDES\_ALT(10) <= "000000";

PAREDES\_ALT(11) <= "000000";

PAREDES\_ALT(12) <= "111011";

PAREDES\_ALT(13) <= "000000";

PAREDES\_ALT(14) <= "000000";

PAREDES\_ALT(15) <= "000000";

PAREDES\_ALT(16) <= "110111";

PAREDES\_ALT(17) <= "000000";

PAREDES\_ALT(18) <= "000000";

PAREDES\_ALT(19) <= "000000";

PAREDES\_ALT(20) <= "111110";

PAREDES\_ALT(21) <= "000000";

PAREDES\_ALT(22) <= "000000";

PAREDES\_ALT(23) <= "000000";

PAREDES\_ALT(24) <= "110111";

PAREDES\_ALT(25) <= "000000";

PAREDES\_ALT(26) <= "000000";

PAREDES\_ALT(27) <= "000000";

PAREDES\_ALT(28) <= "111110";

PAREDES\_ALT(29) <= "000000";

PAREDES\_ALT(30) <= "000000";

PAREDES\_ALT(31) <= "000000";

PAREDES\_ALT(32) <= "111101";

PAREDES\_ALT(33) <= "000000";

PAREDES\_ALT(34) <= "000000";

PAREDES\_ALT(35) <= "000000";

PAREDES\_ALT(36) <= "101111";

PAREDES\_ALT(37) <= "000000";

PAREDES\_ALT(38) <= "000000";

PAREDES\_ALT(39) <= "000000";

PAREDES\_ALT(40) <= "111101";

PAREDES\_ALT(41) <= "000000";

PAREDES\_ALT(42) <= "000000";

PAREDES\_ALT(43) <= "000000";

PAREDES\_ALT(44) <= "101111";

PAREDES\_ALT(45) <= "000000";

PAREDES\_ALT(46) <= "000000";

PAREDES\_ALT(47) <= "000000";

PAREDES\_ALT(48) <= "111011";

PAREDES\_ALT(49) <= "000000";

PAREDES\_ALT(50) <= "000000";

PAREDES\_ALT(51) <= "000000";

PAREDES\_ALT(52) <= "101111";

PAREDES\_ALT(53) <= "000000";

PAREDES\_ALT(54) <= "000000";

PAREDES\_ALT(55) <= "000000";

PAREDES\_ALT(56) <= "111110";

PAREDES\_ALT(57) <= "000000";

PAREDES\_ALT(58) <= "000000";

PAREDES\_ALT(59) <= "000000";

PAREDES\_ALT(60) <= "111011";

PAREDES\_ALT(61) <= "000000";

PAREDES\_ALT(62) <= "000000";

PAREDES\_ALT(63) <= "000000";

PAREDES\_LABE(0) <= "011111";

PAREDES\_LABE(1) <= "000000";

PAREDES\_LABE(2) <= "101111";

PAREDES\_LABE(3) <= "000000";

PAREDES\_LABE(4) <= "110111";

PAREDES\_LABE(5) <= "000000";

PAREDES\_LABE(6) <= "111011";

PAREDES\_LABE(7) <= "000000";

PAREDES\_LABE(8) <= "111101";

PAREDES\_LABE(9) <= "000000";

PAREDES\_LABE(10) <= "111110";

PAREDES\_LABE(11) <= "000000";

PAREDES\_LABE(12) <= "011111";

PAREDES\_LABE(13) <= "000000";

PAREDES\_LABE(14) <= "101111";

PAREDES\_LABE(15) <= "000000";

PAREDES\_LABE(16) <= "111011";

PAREDES\_LABE(17) <= "000000";

PAREDES\_LABE(18) <= "110111";

PAREDES\_LABE(19) <= "000000";

PAREDES\_LABE(20) <= "101111";

PAREDES\_LABE(21) <= "000000";

PAREDES\_LABE(22) <= "011111";

PAREDES\_LABE(23) <= "000000";

PAREDES\_LABE(24) <= "111110";

PAREDES\_LABE(25) <= "000000";

PAREDES\_LABE(26) <= "111101";

PAREDES\_LABE(27) <= "000000";

PAREDES\_LABE(28) <= "111011";

PAREDES\_LABE(29) <= "000000";

PAREDES\_LABE(30) <= "110111";

PAREDES\_LABE(31) <= "000000";

PAREDES\_LABE(32) <= "110110";

PAREDES\_LABE(33) <= "100100";

PAREDES\_LABE(34) <= "101101";

PAREDES\_LABE(35) <= "100100";

PAREDES\_LABE(36) <= "110110";

PAREDES\_LABE(37) <= "100100";

PAREDES\_LABE(38) <= "101101";

PAREDES\_LABE(39) <= "100100";

PAREDES\_LABE(40) <= "110110";

PAREDES\_LABE(41) <= "100100";

PAREDES\_LABE(42) <= "101101";

PAREDES\_LABE(43) <= "100100";

PAREDES\_LABE(44) <= "110110";

PAREDES\_LABE(45) <= "100100";

PAREDES\_LABE(46) <= "101101";

PAREDES\_LABE(47) <= "000000";

PAREDES\_LABE(48) <= "101101";

PAREDES\_LABE(49) <= "100100";

PAREDES\_LABE(50) <= "110110";

PAREDES\_LABE(51) <= "100100";

PAREDES\_LABE(52) <= "101101";

PAREDES\_LABE(53) <= "100100";

PAREDES\_LABE(54) <= "110110";

PAREDES\_LABE(55) <= "100100";

PAREDES\_LABE(56) <= "101101";

PAREDES\_LABE(57) <= "100100";

PAREDES\_LABE(58) <= "110110";

PAREDES\_LABE(59) <= "100100";

PAREDES\_LABE(60) <= "101101";

PAREDES\_LABE(61) <= "100100";

PAREDES\_LABE(62) <= "110110";

PAREDES\_LABE(63) <= "000000";

PAREDES\_LABE(64) <= "110110";

PAREDES\_LABE(65) <= "010010";

PAREDES\_LABE(66) <= "001001";

PAREDES\_LABE(67) <= "100100";

PAREDES\_LABE(68) <= "010010";

PAREDES\_LABE(69) <= "001001";

PAREDES\_LABE(70) <= "100100";

PAREDES\_LABE(71) <= "010010";

PAREDES\_LABE(72) <= "001001";

PAREDES\_LABE(73) <= "100100";

PAREDES\_LABE(74) <= "010010";

PAREDES\_LABE(75) <= "001001";

PAREDES\_LABE(76) <= "100100";

PAREDES\_LABE(77) <= "010010";

PAREDES\_LABE(78) <= "001001";

PAREDES\_LABE(79) <= "000000";

PAREDES\_LABE(80) <= "011011";

PAREDES\_LABE(81) <= "010010";

PAREDES\_LABE(82) <= "100100";

PAREDES\_LABE(83) <= "001001";

PAREDES\_LABE(84) <= "010010";

PAREDES\_LABE(85) <= "100100";

PAREDES\_LABE(86) <= "001001";

PAREDES\_LABE(87) <= "010010";

PAREDES\_LABE(88) <= "100100";

PAREDES\_LABE(89) <= "001001";

PAREDES\_LABE(90) <= "010010";

PAREDES\_LABE(91) <= "100100";

PAREDES\_LABE(92) <= "001001";

PAREDES\_LABE(93) <= "010010";

PAREDES\_LABE(94) <= "100100";

PAREDES\_LABE(95) <= "000000";

PAREDES\_LABE(96) <= "101111";

PAREDES\_LABE(97) <= "100000";

PAREDES\_LABE(98) <= "100000";

PAREDES\_LABE(99) <= "100000";

PAREDES\_LABE(100) <= "100000";

PAREDES\_LABE(101) <= "111110";

PAREDES\_LABE(102) <= "100000";

PAREDES\_LABE(103) <= "100000";

PAREDES\_LABE(104) <= "100000";

PAREDES\_LABE(105) <= "100000";

PAREDES\_LABE(106) <= "101111";

PAREDES\_LABE(107) <= "100000";

PAREDES\_LABE(108) <= "100000";

PAREDES\_LABE(109) <= "100000";

PAREDES\_LABE(110) <= "111011";

PAREDES\_LABE(111) <= "000000";

PAREDES\_LABE(112) <= "100011";

PAREDES\_LABE(113) <= "110111";

PAREDES\_LABE(114) <= "000001";

PAREDES\_LABE(115) <= "001001";

PAREDES\_LABE(116) <= "011101";

PAREDES\_LABE(117) <= "001001";

PAREDES\_LABE(118) <= "000001";

PAREDES\_LABE(119) <= "110111";

PAREDES\_LABE(120) <= "000001";

PAREDES\_LABE(121) <= "001001";

PAREDES\_LABE(122) <= "011101";

PAREDES\_LABE(123) <= "001001";

PAREDES\_LABE(124) <= "000001";

PAREDES\_LABE(125) <= "110111";

PAREDES\_LABE(126) <= "000000";

PAREDES\_LABE(127) <= "000000";

END ARCHITECTURE;

# Apéndice Ñ

Archivo FF\_D\_RISING.vhd

library ieee;

use ieee.std\_logic\_1164.all;

--Modelo de Flip Flop D

entity FF\_D\_RISING is

port

(

-- Input ports

D : in std\_logic;

Clk : in std\_logic; --Reloj

Set : in std\_logic; --Seteo asincrónico

Reset : in std\_logic;

En : in std\_logic;

-- Output ports

Q : out std\_logic

);

end FF\_D\_RISING;

architecture flow of FF\_D\_RISING is

begin

ff\_d\_clk: process (Clk, Set, Reset,En)

begin

if (En='0') then null;

elsif (Set='1') then

Q<='1';

elsif (Reset='1') then

Q<='0';

elsif (rising\_edge(Clk)) then

Q<=D;

end if;

end process;

end flow;

# Apéndice O

Archivo FF\_AND.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity FF\_AND is

port(

Clk : in std\_logic;

IN1 : in std\_logic;

IN2 : in std\_logic;

OUT1 : out std\_logic

);

end FF\_AND;

architecture beh of FF\_AND is

signal FF\_OUT1, FF\_OUT2 : std\_logic;

component FF\_D\_RISING is

port(

D : in std\_logic;

Clk : in std\_logic;

Set : in std\_logic;

Reset : in std\_logic;

En : in std\_logic;

Q : out std\_logic

);

end component;

begin

FF1: FF\_D\_RISING port map(

D =>IN1,

Clk =>CLK,

Set =>'0',

Reset =>'0',

En =>'1',

Q =>FF\_OUT1

);

FF2: FF\_D\_RISING port map(

D =>FF\_OUT1,

Clk =>CLK,

Set =>'0',

Reset =>'0',

En =>'1',

Q =>FF\_OUT2

);

OUT1 <= IN2 and FF\_OUT2;

end beh;

# Apéndice P

Archivo Dificultad.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity Dificultad is

port(

Clk : in std\_logic;

switches : in std\_logic\_vector(1 downto 0);

gameover : in std\_logic;

Dificultad : out std\_logic\_vector(1 downto 0);

Leds\_Rojo : out std\_logic\_vector(7 downto 0);

Leds\_Verde : out std\_logic\_vector(7 downto 0)

);

end Dificultad;

architecture beh of Dificultad is

begin

Setea\_dificultad: Process(Clk,gameover,switches)

begin

if (rising\_edge(Clk)) then

if(gameover='0') then

Leds\_Verde<=(others=>'1');

elsif (gameover='1') then

Leds\_Verde<=(others=>'0');

if (switches="00") then

Leds\_Rojo<= "00000011";

elsif (switches = "01") then

Leds\_Rojo<= "00001111";

elsif (switches = "10") then

Leds\_Rojo<= "00111111";

elsif (switches = "11") then

Leds\_Rojo<= "11111111";

end if;

dificultad<=switches;

end if;

end if;

end process;

end beh;

# Apéndice Q

Archivo DIBUJO.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use work.HexaPackage.all;

ENTITY DIBUJO IS

PORT(

CLK : IN STD\_LOGIC;

ALTERNA : IN STD\_LOGIC;

CCOLOR : IN STD\_LOGIC;

--RESET : IN STD\_LOGIC;

DPAREDES : IN STD\_LOGIC;

DISLA : IN STD\_LOGIC;

DJUGADOR : IN STD\_LOGIC;

CUADRANTE: IN STD\_LOGIC\_VECTOR(2 DOWNTO 0);

RAND : IN STD\_LOGIC\_VECTOR(63 DOWNTO 0);

ROJO : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

VERDE : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

AZUL : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0)

);

END ENTITY;

ARCHITECTURE BEH OF DIBUJO IS

TYPE A\_2D IS ARRAY(0 TO 15) OF STD\_LOGIC\_VECTOR(11 DOWNTO 0);

SIGNAL INV\_COLOR: STD\_LOGIC;

CONSTANT ESQUEMAS\_COLOR: A\_2D :=

(0=>X"311" , 1=>X"000" , 2=>X"F11" , 3=>X"EEE" ,

4=>X"131" , 5=>X"000" , 6=>X"1F1" , 7=>X"EEE" ,

8=>X"113" , 9=>X"000" , 10=>X"11F" , 11=>X"EEE" ,

12=>X"313" , 13=>X"000" , 14=>X"F1F" , 15=>X"EEE" );

SIGNAL ESQUEMA: UNSIGNED(1 DOWNTO 0);

BEGIN

INVIERTECOLOR:PROCESS (CLK)

BEGIN

IF (RISING\_EDGE(CLK)) THEN

IF (ALTERNA='1') THEN

INV\_COLOR<= NOT(INV\_COLOR);

END IF;

END IF;

END PROCESS;

CAMBIACOLORES:PROCESS (CLK)

BEGIN

IF RISING\_EDGE(CLK) THEN

IF (CCOLOR='1') THEN

ESQUEMA <= UNSIGNED(RAND(18 DOWNTO 17));

END IF;

END IF;

END PROCESS;

DEFINEDIBUJO:PROCESS(CLK,DJUGADOR,DISLA,DPAREDES,CUADRANTE,INV\_COLOR)

VARIABLE INDICE: INTEGER;

VARIABLE DATO : STD\_LOGIC\_VECTOR(11 DOWNTO 0);

BEGIN

IF (RISING\_EDGE(CLK)) THEN

IF(DJUGADOR='1') THEN

INDICE := TO\_INTEGER(ESQUEMA)\*4+3;

DATO := ESQUEMAS\_COLOR(INDICE);

ROJO <= DATO(3 DOWNTO 0);

VERDE <= DATO(7 DOWNTO 4);

AZUL <= DATO(11 DOWNTO 8);

ELSIF(DISLA='1') THEN

INDICE := TO\_INTEGER(ESQUEMA)\*4+1;

DATO := ESQUEMAS\_COLOR(INDICE);

ROJO <= DATO(3 DOWNTO 0);

VERDE <= DATO(7 DOWNTO 4);

AZUL <= DATO(11 DOWNTO 8);

ELSIF(DPAREDES='1') THEN

INDICE := TO\_INTEGER(ESQUEMA)\*4+2;

DATO := ESQUEMAS\_COLOR(INDICE);

ROJO <= DATO(3 DOWNTO 0);

VERDE <= DATO(7 DOWNTO 4);

AZUL <= DATO(11 DOWNTO 8);

ELSE

IF(bit2bool(CUADRANTE(0)) XOR bit2bool(INV\_COLOR)) THEN

INDICE := TO\_INTEGER(ESQUEMA)\*4+0;

DATO := ESQUEMAS\_COLOR(INDICE);

ROJO <= DATO(3 DOWNTO 0);

VERDE <= DATO(7 DOWNTO 4);

AZUL <= DATO(11 DOWNTO 8);

ELSE

INDICE := TO\_INTEGER(ESQUEMA)\*4+1;

DATO := ESQUEMAS\_COLOR(INDICE);

ROJO <= DATO(3 DOWNTO 0);

VERDE <= DATO(7 DOWNTO 4);

AZUL <= DATO(11 DOWNTO 8);

END IF;

END IF;

END IF;

END PROCESS;

END ARCHITECTURE;

# Apéndice R

Archivo DesplazaXYalCentro.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use work.HexaPackage.all;

entity DesplazaXYalCentro is

port(

Clk : in std\_logic;

Xin : in unsigned(9 downto 0);

Yin : in unsigned(9 downto 0);

Xout : out signed(9 downto 0);

Yout : out signed(9 downto 0)

);

end DesplazaXYalCentro;

architecture beh of DesplazaXYalCentro is

begin

Desplazar: process(Clk)

begin

if rising\_edge(clk) then

Xout <= signed(std\_logic\_vector(Xin)) - CENTROX;

Yout <= signed(std\_logic\_vector(Yin)) - CENTROY;

end if;

end process;

end beh;

# Apéndice S

Archivo DEC\_HEX\_7SEG.vhd

library ieee;

use ieee.std\_logic\_1164.all;

entity DEC\_HEX\_7SEG is

port

(

in1 : in std\_logic\_vector(3 downto 0);

out1 : out std\_logic\_vector(0 to 6)

);

end DEC\_HEX\_7SEG;

architecture DECOD of DEC\_HEX\_7SEG is

begin

with in1 select

out1 <= "1000000" when "0000", --0

"1111001" when "0001", --1

"0100100" when "0010", --2

"0110000" when "0011", --3

"0011001" when "0100", --4

"0010010" when "0101", --5

"0000010" when "0110", --6

"1111000" when "0111", --7

"0000000" when "1000", --8

"0011000" when "1001", --9

"0001000" when "1010", --A

"0000011" when "1011", --B

"1000110" when "1100", --C

"0100001" when "1101", --D

"0000110" when "1110", --E

"0001110" when "1111", --F

"1111111" when others;

end DECOD;

# Apéndice T

Archivo conv\_hexagonal.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity conv\_hexagonal is

port(

clk : in std\_logic;

x : in signed (9 downto 0);

y : in signed (9 downto 0);

cuadrante : out unsigned (2 downto 0);

radio : out unsigned (9 downto 0)

);

end conv\_hexagonal;

architecture beh of conv\_hexagonal is

constant raiz\_3int : integer := 1773;

constant raiz\_3 : signed (11 downto 0) := to\_signed(raiz\_3int,12);

signal r\_3\_x\_completo : signed (21 downto 0);

signal r\_3\_x : signed (11 downto 0);

signal r\_3\_x\_neg : signed (11 downto 0);

signal y\_ext : signed (11 downto 0);

signal y\_neg : signed (11 downto 0);

signal radio\_aux : signed (11 downto 0);

begin

r\_3\_x\_completo <= x \* raiz\_3;

r\_3\_x <= r\_3\_x\_completo (21 downto 10);

r\_3\_x\_neg <= -r\_3\_x;

y\_ext <= (y(9) & y(9) & y);

y\_neg <= -y\_ext;

convertir : process (clk)

begin

if (rising\_edge(clk)) then

if (y\_ext > to\_signed(0,12)) then

if (y\_ext < r\_3\_x) then -- cuadrante 0

cuadrante <= to\_unsigned(0,3);

radio\_aux <= y\_ext + r\_3\_x;

radio <= unsigned(radio\_aux (10 downto 1));

elsif (y\_ext < r\_3\_x\_neg) then -- cuadrante 2

cuadrante <= to\_unsigned(2,3);

radio\_aux <= y\_ext + r\_3\_x\_neg;

radio <= unsigned(radio\_aux (10 downto 1));

else -- cuadrante 1

cuadrante <= to\_unsigned(1,3);

-- radio <= unsigned(std\_logic\_vector(y));

radio\_aux <= y\_ext + y\_ext;

radio <= unsigned(radio\_aux (10 downto 1));

end if;

else

if (y\_ext > r\_3\_x) then -- cuadrante 3

cuadrante <= to\_unsigned(3,3);

radio\_aux <= y\_neg + r\_3\_x\_neg;

radio <= unsigned(radio\_aux (10 downto 1));

elsif (y\_ext > r\_3\_x\_neg) then -- cuadrante 5

cuadrante <= to\_unsigned(5,3);

radio\_aux <= y\_neg + r\_3\_x;

radio <= unsigned(radio\_aux (10 downto 1));

else -- cuadrante 4

cuadrante <= to\_unsigned(4,3);

-- radio <= unsigned(std\_logic\_vector(-y));

radio\_aux <= y\_neg + y\_neg;

radio <= unsigned(radio\_aux (10 downto 1));

end if;

end if;

-- radio <= unsigned( resize(x\*x + y\*y,10));

end if;

end process;

end beh;

# Apéndice U

Archivo CONTROLADOR\_VGA.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use work.HexaPackage.all;

ENTITY CONTROLADOR\_VGA IS

GENERIC( BITS: INTEGER:=10);

PORT(

CLK : IN STD\_LOGIC;

CLR : IN STD\_LOGIC;

RED\_IN : IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

GREEN\_IN : IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

BLUE\_IN : IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

HSYNC : OUT STD\_LOGIC;

VSYNC : OUT STD\_LOGIC;

VGA\_SYNC\_N: OUT STD\_LOGIC;

VGA\_BLANK\_N:OUT STD\_LOGIC;

X : OUT UNSIGNED(BITS-1 DOWNTO 0);

Y : OUT UNSIGNED(BITS-1 DOWNTO 0);

RED : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

GREEN : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

BLUE : OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

END\_FRAME: OUT STD\_LOGIC;

PRE\_FRAME: OUT STD\_LOGIC

);

END ENTITY;

ARCHITECTURE BEH OF CONTROLADOR\_VGA IS

CONSTANT HPIXELS : UNSIGNED:= TO\_UNSIGNED(800,BITS); --SYNC\_PULSE +BP +AREA VISIBLE +FP

CONSTANT vLINES : UNSIGNED:= TO\_UNSIGNED(525,BITS);

CONSTANT HPULSE : UNSIGNED:= TO\_UNSIGNED(96,BITS); --96

CONSTANT VPULSE : UNSIGNED:= TO\_UNSIGNED(2,BITS); --2

CONSTANT HBP : UNSIGNED:= TO\_UNSIGNED(144,BITS); --48

CONSTANT VBP : UNSIGNED:= TO\_UNSIGNED(35,BITS); --33

CONSTANT HAREA : UNSIGNED:= TO\_UNSIGNED(640,BITS); --640

CONSTANT VAREA : UNSIGNED:= TO\_UNSIGNED(480,BITS); --480

CONSTANT HFP : UNSIGNED:= TO\_UNSIGNED(784,BITS); --16

CONSTANT VFP : UNSIGNED:= TO\_UNSIGNED(515,BITS); --10

----PARA PROBAR 800X600 -- SE NECESITA GENERAR RELOJ DE 40 MHZ

--

--CONSTANT HPIXELS : UNSIGNED:= TO\_UNSIGNED(1056,BITS); --SYNC\_PULSE +BP +AREA VISIBLE +FP

--CONSTANT vLINES : UNSIGNED:= TO\_UNSIGNED(628,BITS);

--CONSTANT HPULSE : UNSIGNED:= TO\_UNSIGNED(128,BITS); --128

--CONSTANT VPULSE : UNSIGNED:= TO\_UNSIGNED(4,BITS); --4

--CONSTANT HBP : UNSIGNED:= TO\_UNSIGNED(216,BITS); --88

--CONSTANT VBP : UNSIGNED:= TO\_UNSIGNED(27,BITS); --23

--CONSTANT HAREA : UNSIGNED:= TO\_UNSIGNED(800,BITS); --800

--CONSTANT VAREA : UNSIGNED:= TO\_UNSIGNED(600,BITS); --600

--CONSTANT HFP : UNSIGNED:= TO\_UNSIGNED(1016,BITS); --40

--CONSTANT VFP : UNSIGNED:= TO\_UNSIGNED(627,BITS); --1

SIGNAL HC : UNSIGNED(BITS-1 DOWNTO 0);

SIGNAL VC : UNSIGNED(BITS-1 DOWNTO 0);

SIGNAL IN\_FRAME : STD\_LOGIC;

SIGNAL VGA\_SYNC,VGA\_BLANK: STD\_LOGIC;

BEGIN

HSYNC<='0' WHEN (HC<HPULSE) ELSE '1';

VSYNC<='0' WHEN (VC<VPULSE) ELSE '1';

VGA\_SYNC <= '1' WHEN (VC >= VBP AND VC < VFP) ELSE '0';

VGA\_BLANK<= '1' WHEN (HC >= HBP AND HC < HFP) ELSE '0';

IN\_FRAME <= '1' WHEN (VGA\_SYNC='1' AND VGA\_BLANK='1') ELSE '0';

PRE\_FRAME <= '1' WHEN (HC=(BITS-1 DOWNTO 0 => '0') AND VC=(BITS-1 DOWNTO 0 => '0')) ELSE '0';

END\_FRAME <= '1' WHEN (HC=(BITS-1 DOWNTO 0 => '0') AND VC=VFP) ELSE '0';

VGA\_BLANK\_N<=VGA\_BLANK;

VGA\_SYNC\_N<=VGA\_SYNC;

SUMACONTADORES:PROCESS (CLK,CLR,HC,VC)

BEGIN

IF (CLR='1') THEN

HC<=(OTHERS=>'0');

VC<=(OTHERS=>'0');

ELSIF (RISING\_EDGE(CLK)) THEN

IF(HC < HPIXELS-1) THEN

HC<=HC+1;

ELSE

HC<=(OTHERS=>'0');

IF(VC < VLINES-1) THEN

VC<=VC+1;

ELSE

VC<=(OTHERS=>'0');

END IF;

END IF;

END IF;

END PROCESS;

SETEACOORDENADAS:PROCESS (CLK)

BEGIN

IF(RISING\_EDGE(CLK)) THEN

IF(VC >= VBP AND VC < VFP) THEN

Y <= VC-VBP;

ELSE

Y <= (OTHERS=>'0');

END IF;

IF(HC >= HBP AND HC < HFP) THEN

X <= HC-HBP;

ELSE

X <= (OTHERS=>'0');

END IF;

END IF;

END PROCESS;

SETEACOLORES:PROCESS (CLK)

BEGIN

IF RISING\_EDGE(CLK) THEN

IF (IN\_FRAME='1') THEN

RED <= RED\_IN;

GREEN <= GREEN\_IN;

BLUE <= BLUE\_IN;

ELSE

RED <= (OTHERS => '0');

GREEN <= (OTHERS => '0');

BLUE <= (OTHERS => '0');

END IF;

END IF;

END PROCESS;

END ARCHITECTURE;

# Apéndice V

Archivo control\_giro.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity control\_giro is

port(

clk : in std\_logic;

reset : in std\_logic;

Actualizar : in std\_logic;

Random : in std\_logic\_vector(63 downto 0);

subir\_velocidad : in std\_logic;

velocidad\_prueba : out unsigned(5 downto 0);

angulo : out unsigned(9 downto 0)

);

end control\_giro;

architecture beh of control\_giro is

signal velocidad : unsigned(9 downto 0);

signal cambiar\_giro : unsigned(8 downto 0);

signal direccion : std\_logic;

signal angulo\_aux : unsigned(9 downto 0);

begin

velocidad\_prueba<=velocidad(5 downto 0);

AUMENTAR\_VELOCIDAD : process (clk)

begin

if (rising\_edge(clk)) then

if (reset='1') then

velocidad <= to\_unsigned(1,10);

elsif (subir\_velocidad='1') then

velocidad <= velocidad + 1;

--velocidad <= to\_unsigned(0,10);

end if;

end if;

end process;

CAMBIAR\_DIRECCION : process (clk,reset,actualizar,cambiar\_giro)

begin

if (rising\_edge(clk)) then

if (reset='1') then

cambiar\_giro <= to\_unsigned(0,9);

elsif (actualizar='1') then

if (cambiar\_giro = to\_unsigned(0,9)) then

cambiar\_giro <= unsigned(random(16 downto 9)) + to\_unsigned(60,9);

direccion <= not(direccion);

else

cambiar\_giro <= cambiar\_giro - to\_unsigned(1,9);

end if;

end if;

end if;

end process;

AUMENTAR\_ANGULO : process (clk, actualizar, direccion,reset)

begin

if (rising\_edge(clk)) then

if (reset='1') then

angulo\_aux<= to\_unsigned(0,10);

elsif (actualizar='1') then

if (direccion='1') then

angulo\_aux <= angulo\_aux + velocidad;

else

angulo\_aux <= angulo\_aux - velocidad;

end if;

end if;

end if;

end process;

angulo <= angulo\_aux;

end beh;

# Apéndice W

Archivo CONTADOR.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity CONTADOR is

generic(max:integer:=16);

port

(

-- Input ports

Clk : in std\_logic; --Reloj

EN : in std\_logic; --Entrada de habilitación

Rst : in std\_logic; --

-- Output ports

ovf : out std\_logic --Overflow.

);

end CONTADOR;

architecture beh of CONTADOR is

signal ovf\_aux: std\_logic;

function Log2( input:integer ) return integer is

variable temp,log:integer;

begin

temp:=input;

log:=0;

while (temp /= 0) loop

temp:=temp/2;

log:=log+1;

end loop;

return log;

end function Log2;

constant bits: integer:= Log2(max-1);

signal count: unsigned(bits-1 downto 0);

begin

clock\_event: process (Clk, Rst, count, EN)

begin

if(Rst='1') then --Reset Asincrónico

count <= (others=>'0');

ovf<='0';

elsif(rising\_edge(Clk)) then

ovf<='0';

if(to\_integer(count) = max-1) then

ovf <= '1';

count <= (others=>'0');

elsif(EN='1') then

ovf<='0';

count <= count +1;

end if;

end if;

end process;

end beh;

# Apéndice X

Archivo Compara\_coordenadas.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity Compara\_coordenadas is

port(

--inputs

Xin1 : in signed(9 downto 0);

Yin1 : in signed(9 downto 0);

Xin2 : in signed(9 downto 0);

Yin2 : in signed(9 downto 0);

--outputs

Comparacion : out std\_logic

);

end Compara\_coordenadas;

architecture beh of Compara\_coordenadas is

begin

Comparacion <= '1' when (Xin1 = Xin2 and Yin1 = Yin2) else '0';

end beh;

# Apéndice Y

Archivo CIRCULO.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

ENTITY CIRCULO IS

PORT(

CLK : IN STD\_LOGIC;

X1 : IN SIGNED(9 DOWNTO 0);

Y1 : IN SIGNED(9 DOWNTO 0);

X2 : IN SIGNED(9 DOWNTO 0);

Y2 : IN SIGNED(9 DOWNTO 0);

DIBUJA: OUT STD\_LOGIC

);

END ENTITY;

ARCHITECTURE BEH OF CIRCULO IS

CONSTANT RADIO : SIGNED(9 DOWNTO 0) := TO\_SIGNED(2,10);

SIGNAL DX,DY : SIGNED(9 DOWNTO 0);

BEGIN

CALCULA:PROCESS(CLK,DX,DY)

BEGIN

IF RISING\_EDGE(CLK) THEN

DX<=X1-X2;

DY<=Y1-Y2;

IF (DX\*DX+DY\*DY<RADIO\*RADIO) THEN

DIBUJA <= '1';

ELSE

DIBUJA <= '0';

END IF;

END IF;

END PROCESS;

END ARCHITECTURE;

# Apéndice Z

Archivo RS232\_RX.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

ENTITY RS232\_RX IS

PORT(

CLK: IN STD\_LOGIC;

RST: IN STD\_LOGIC;

RX: IN STD\_LOGIC;

FSEL: IN STD\_LOGIC\_VECTOR(1 DOWNTO 0);

DATA\_OK: OUT STD\_LOGIC;

DATAOUT: OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END RS232\_RX;

ARCHITECTURE BEH OF RS232\_RX IS

COMPONENT RS232\_RX\_FSM IS

PORT(

CLK: IN STD\_LOGIC;

RST: IN STD\_LOGIC;

TX: IN STD\_LOGIC;

SHIFT: OUT STD\_LOGIC;

DATA\_OK: OUT STD\_LOGIC

);

END COMPONENT;

COMPONENT SHIFT\_REGISTER\_RX IS

GENERIC(

data\_width: integer:=8

);

PORT(

CLK: IN STD\_LOGIC;

RST: IN STD\_LOGIC;

DATAIN: IN STD\_LOGIC;

SHIFT: IN STD\_LOGIC;

DATAOUT: OUT STD\_LOGIC\_VECTOR(data\_width-1 DOWNTO 0)

);

END COMPONENT;

COMPONENT DATA\_BUFFER IS

GENERIC(

data\_width: integer:=8

);

PORT(

DATAIN: IN STD\_LOGIC\_VECTOR(data\_width-1 DOWNTO 0);

EN: IN STD\_LOGIC;

CLK: IN STD\_LOGIC;

CLEAR: IN STD\_LOGIC;

DATAOUT: OUT STD\_LOGIC\_VECTOR(data\_width-1 DOWNTO 0)

);

END COMPONENT;

SIGNAL DOK, SHIFT, PCHECK: STD\_LOGIC;

SIGNAL PBIT: STD\_LOGIC;

SIGNAL DATA\_BUFF: STD\_LOGIC\_VECTOR(7 DOWNTO 0);

BEGIN

RX\_FSM1: RS232\_RX\_FSM PORT MAP( CLK=>CLK,

RST=>RST,

TX=>RX,

SHIFT=>SHIFT,

DATA\_OK=>DOK);

SR1: SHIFT\_REGISTER\_RX PORT MAP( CLK=>CLK,

RST=>RST,

DATAIN=>RX,

SHIFT=>SHIFT,

DATAOUT=>DATA\_BUFF);

DBUFFER: DATA\_BUFFER PORT MAP( DATAIN => DATA\_BUFF,

EN => DOK,

CLK => CLK,

DATAOUT => DATAOUT,

CLEAR => '0');

PROCESS (CLK)

VARIABLE AUX: STD\_LOGIC;

BEGIN

IF (RISING\_EDGE(CLK)) THEN

DATA\_OK <= AUX;

AUX := DOK;

END IF;

END PROCESS;

END BEH;

# Presupuesto

Detalle de horas trabajadas

|  |  |
| --- | --- |
| Tarea | Tiempo [Horas] |
| Diseño y desarrollo de sistema implementado en FPGA | 24 |
| Diseño y desarrollo de sistema implementado en Android | 8 |
| Configuración de módulo Bluetooth | 1 |
| Depuración | 15 |
| Programación de sistema completo en FPGA | 0.5 |
| Documentación | 8 |
|  | 57 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ítem** | **Cantidad** | **Costo unitario [U$D]** | **Subtotal[U$D]** | **Costo total [U$D]** |
| Costo de diseño |  |  |  | 570,00 |
| Personal | 57 | 10,00 | 570,00 |  |
| Costo de materiales |  |  |  | 55,00 |
| Placa de desarrollo amortizada | 1 | 50,00 | 50,00 |  |
| Módulo Bluetooth | 1 | 5,00 | 5,00 |  |
|  |  |  |  |  |
|  |  |  | Total (S/Imp): | **625,00** |
|  |  |  | Total (C/Imp): | **756,00** |

Para la realización del proyecto se debe abonar el 35% (treinta y cinco) del costo total antes de dar comienzo con el mismo, mientras que el saldo restante se podrá cancelar mediante un plan de pago a convenir.

El pago puede ser realizado en dólares estadounidenses o en pesos argentinos tomando la cotización del dólar del día en que se efectúe el pago.

El presente presupuesto tiene un plazo de validez de 15 (quince) días desde que se realiza la entrega al cliente.