

Complementos de Electrónica

Curso 2017/2018

Flappy Madrid



Francisco José Díaz Romero

Francisco Javier Ortiz Bonilla

Alberto Fuentes Muñoz

Índice

1.	Diagrama de bloques y funcionamiento.....	3
2.	Diagramas de bolas de los diferentes bloques	5
2.1.	Pájaro	5
2.2.	Columna	6
2.3.	Gestor para el acceso a las ROM de las columnas	7
2.4.	Gestor de partida	8
3.	Lista de warnings.....	9
4.	Recursos utilizados de la FPGA.....	10

1. Diagrama de bloques y funcionamiento

A continuación se muestra el diagrama de bloques del circuito completo, el cual se adjunta también en la imagen **esquema.jpg** donde se puede ver mejor.

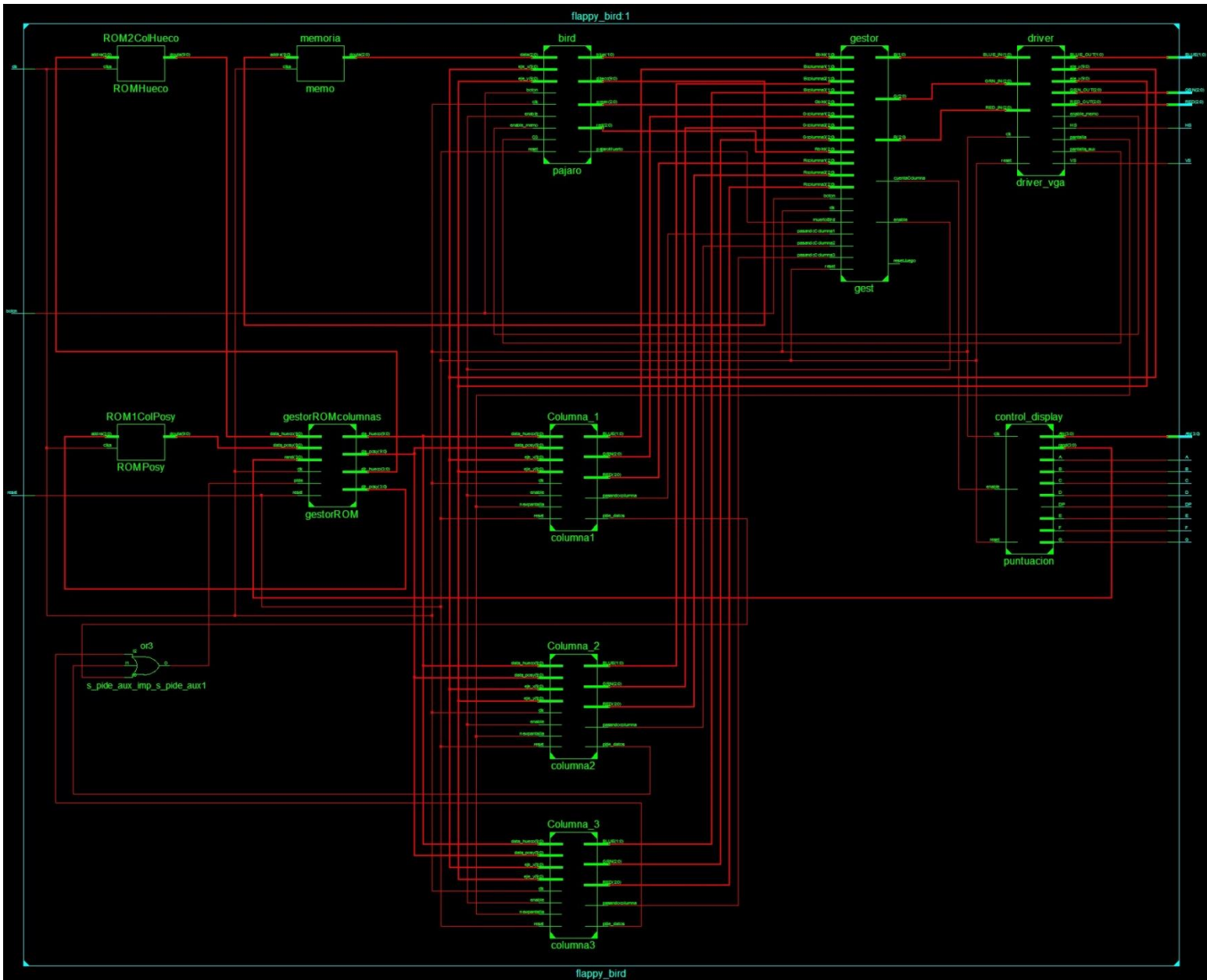


Ilustración 1: Diagrama de bloques del circuito.

Los bloques del sistema son los siguientes:

- Memorias ROM
 - **memoria:** Memoria ROM donde se almacena el sprite del pájaro. En nuestro caso, corresponde al escudo del Real Madrid.
 - **ROM1ColPosy:** Memoria ROM donde se almacenan diferentes posiciones verticales en las que comenzará el hueco de cada columna.

- **ROM2ColHueco:** Memoria ROM donde se almacenan diferentes anchuras de hueco para las columnas.

Las columnas, cada vez que vayan a aparecer en la pantalla, obtendrán de la ROM un valor de anchura de hueco y coordenada vertical donde empieza este.

- **bird**

Es el bloque correspondiente al pájaro. Controla el movimiento de este mediante la entrada del botón con el que se mueve, y da como salida lo que se tiene que pintar en la pantalla con respecto al pájaro.

Mediante la entrada `muertoBird` sabe si el pájaro ha muerto o no.

- **Columna**

Es el bloque correspondiente a una columna. Controla el movimiento horizontal de estas y da como salida lo que se tiene que pintar en la pantalla con respecto a una columna.

En el circuito existen 3 columnas, y mediante un generic (`retardoInicio`), se indica a cada una un retraso inicial de forma que salgan una tras otra, y no todas a la vez.

```
Generic (coordenada_pajaro : unsigned(9 downto 0) := to_unsigned(100,10);
        retardo : integer := 642;
        retardoInicio : integer := 642);
```

Se controla si el pájaro está pasando a través de una columna (la columna estaría en la posición horizontal del pájaro: `coordenada_pajaro`) y de esta forma se pueden contar las columnas que se han conseguido pasar en una partida.

Cada vez que una columna vuelve al principio de la pantalla, lee un par de datos nuevos de las memorias `ROM1ColPosy` y `ROM2ColHueco`.

- **gestorROMColumnas**

Es el encargado de acceder a las memorias `ROM1ColPosy` y `ROM2ColHueco` en nombre de las columnas y de proporcionar los datos a estas.

- **gestor**

Es el bloque que maneja el estado de la partida y decide qué se pinta en la pantalla.

Recibe las salidas RGB de cada columna y de bird, y en función de estas determina qué se pinta (salida RGB que se envía al driver vga) y si el pájaro muere (se choca con una columna o con el suelo).

También se encarga de sumar 1 a la puntuación cada vez que el pájaro pasa por una columna.

- **driver**

Driver para el protocolo VGA reciclado de la práctica 2.

- **control_display**

Bloque reciclado de la práctica 1. Se usa para mostrar en el display de 7 segmentos la puntuación.

2. Diagramas de bolas de los diferentes bloques

2.1. Pájaro

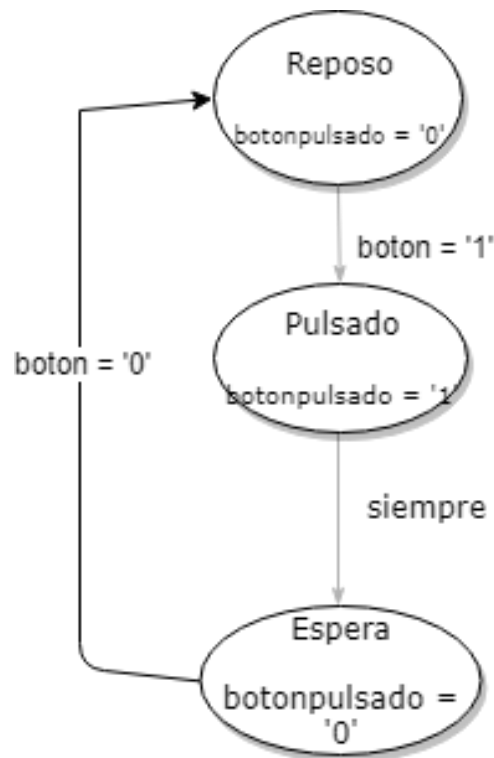


Ilustración 2: Diagrama de bolas para la pulsación del botón en bird.vhd

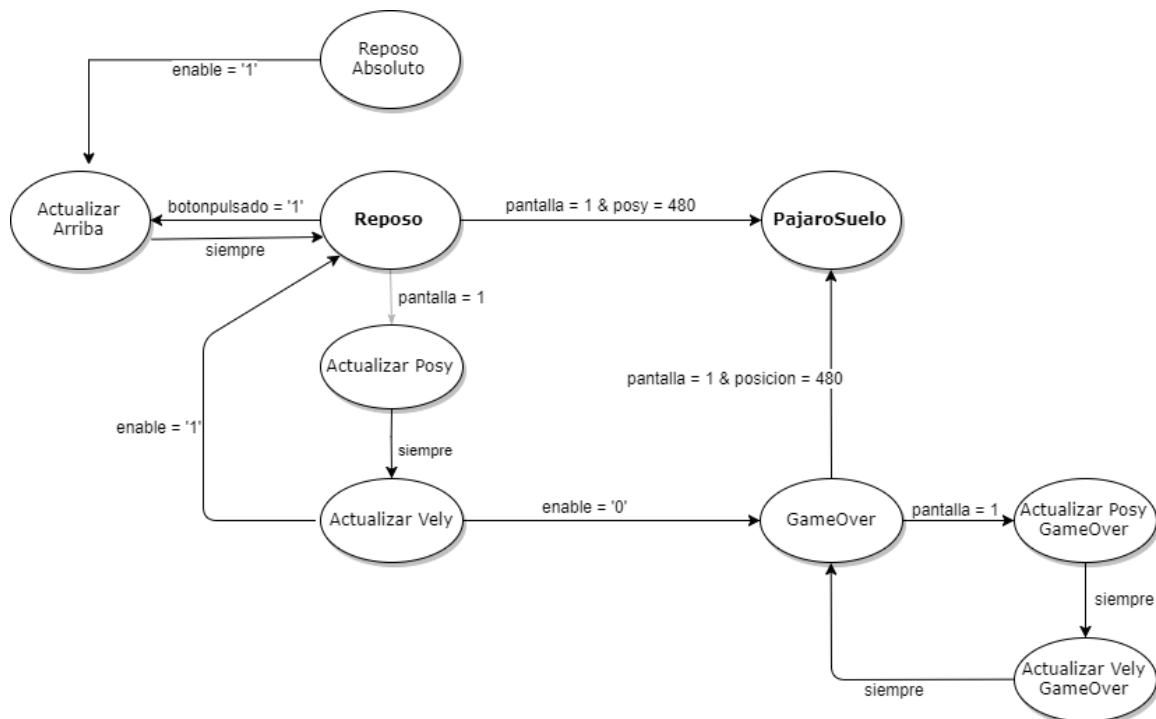


Ilustración 3: Diagrama de bolas de la máquina de estados de bird.vhd

2.2. Columna

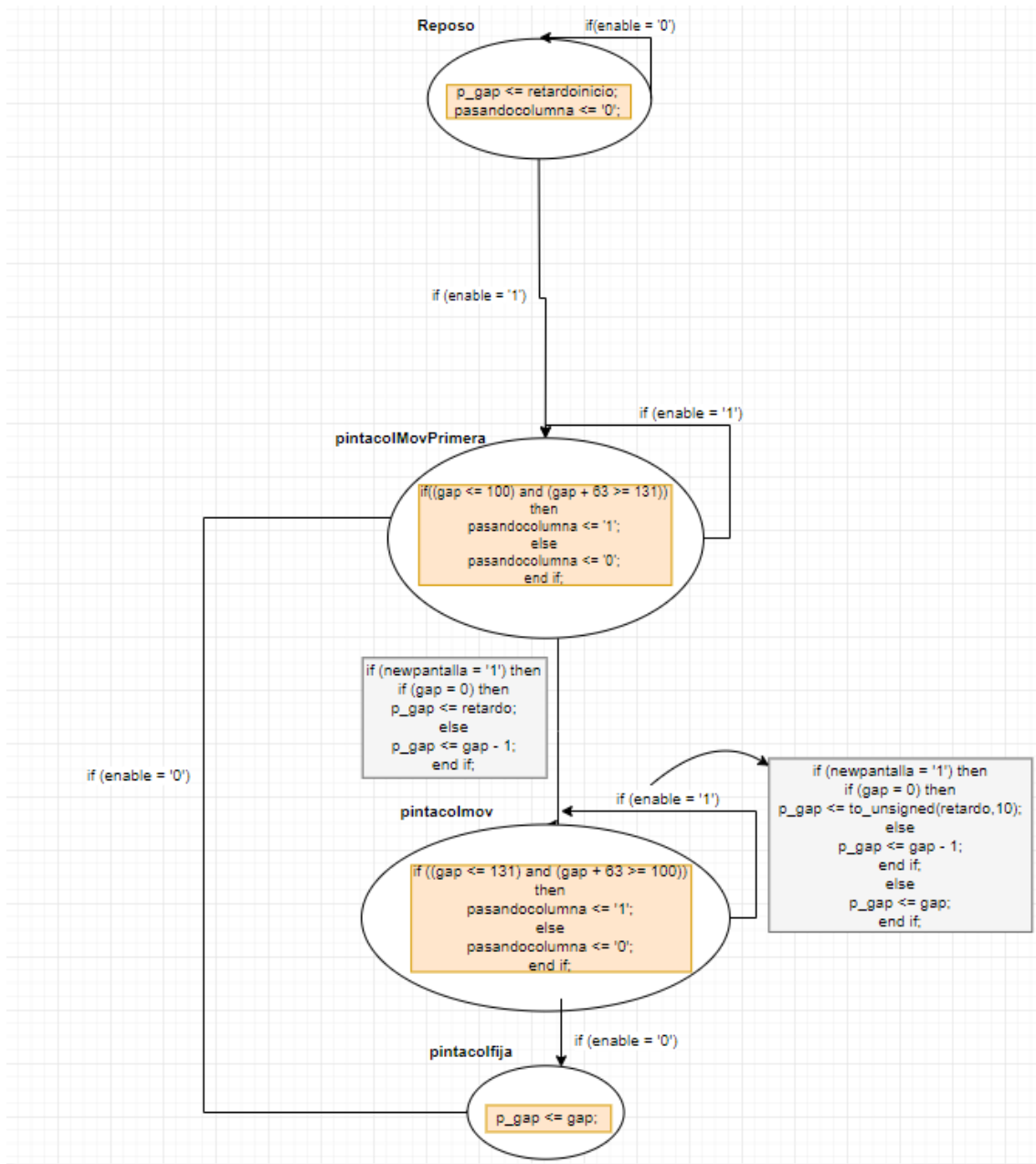


Ilustración 4: Diagrama de bolas de la máquina de estados de columna.vhd

2.3. Gestor para el acceso a las ROM de las columnas

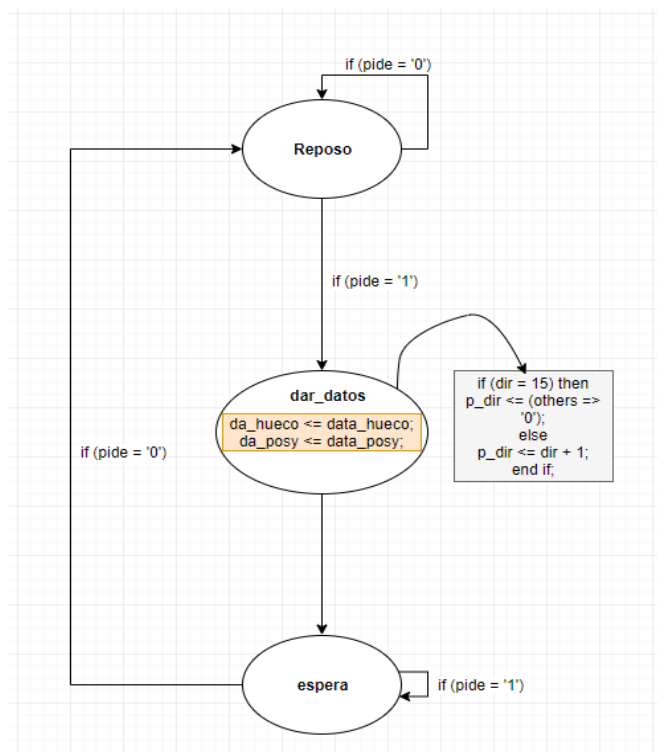
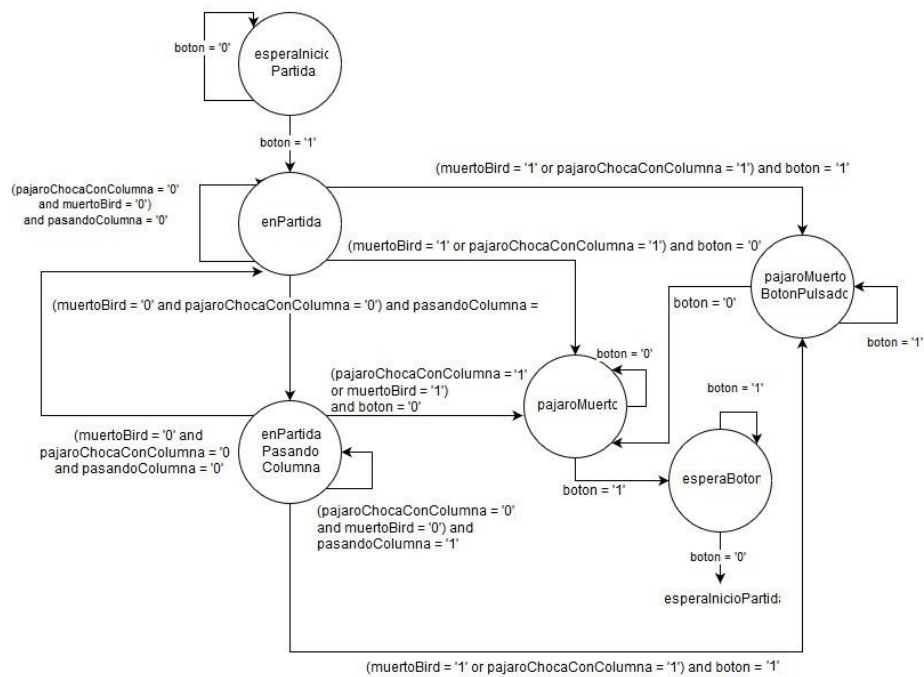


Ilustración 5: Diagrama de bolas de la máquina de estados de gestorROMcolumnas.vhd

En los estados de Reposo y Espera las salidas da_hueco y da_posy están proporcionando el último valor leído de la memoria. Es en el estado dar_datos en el que se incrementa la dirección de memoria y cambian los valores de las salidas da_hueco y da_posy.

2.4. Gestor de partida



Salidas de los estados

esperarInicioPartida
 cuentaColumna = '0'
 resetJuego = '0'
 enable = '0'

enPartida
 cuentaColumna = '0'
 resetJuego = '0'
 enable = '1'

enPartidaPasandoColumna
 if (pasandoColumna = '0')
 cuentaColumna = '1'
 else cuentaColumna = '0'
 resetJuego = '0'
 enable = '1'

pajaritoMuertoBotonPulsado
 cuentaColumna = '0'
 resetJuego = '0'
 enable = '0'

pajaritoMuerto
 cuentaColumna = '0'
 resetJuego = '0'
 enable = '0'

esperaBoton
 cuentaColumna = '0'
 resetJuego = '0'
 enable = '0'

Ilustración 6: Diagrama de bolas de la máquina de estados de gestor.vhd

3. Lista de warnings

```
Warnings
WARNING:Xst:2211 - "C:/Users/Javi/Desktop/TODO 2/TODO 2/flappy_bird.vhd" line 267: Instantiating black box module <memoria>.
WARNING:Xst:2211 - "C:/Users/Javi/Desktop/TODO 2/TODO 2/flappy_bird.vhd" line 274: Instantiating black box module <ROM1ColPosy>.
WARNING:Xst:2211 - "C:/Users/Javi/Desktop/TODO 2/TODO 2/flappy_bird.vhd" line 281: Instantiating black box module <ROM2ColHueco>.
WARNING:Xst:753 - "C:/Users/Javi/Desktop/TODO 2/TODO 2/control_display.vhd" line 155: Unconnected output port 'sat' of component 'cont_digito'.
WARNING:PhysDesignRules:812 - Dangling pin <DOB2> on
block:<ROMHueco/U0/xst_blk_mem_generator/gnativebmg.native_blk_mem_gen/valid.
cstr/ramloop[0].ram.r/s3_init.ram/spram.ram.B>:<RAMB16_RAMB16B>.
WARNING:PhysDesignRules:812 - Dangling pin <DOB3> on
block:<ROMHueco/U0/xst_blk_mem_generator/gnativebmg.native_blk_mem_gen/valid.
cstr/ramloop[0].ram.r/s3_init.ram/spram.ram.B>:<RAMB16_RAMB16B>.
WARNING:PhysDesignRules:812 - Dangling pin <DOB4> on
block:<ROMHueco/U0/xst_blk_mem_generator/gnativebmg.native_blk_mem_gen/valid.
cstr/ramloop[0].ram.r/s3_init.ram/spram.ram.B>:<RAMB16_RAMB16B>.
WARNING:PhysDesignRules:812 - Dangling pin <DOB5> on
block:<ROMHueco/U0/xst_blk_mem_generator/gnativebmg.native_blk_mem_gen/valid.
cstr/ramloop[0].ram.r/s3_init.ram/spram.ram.B>:<RAMB16_RAMB16B>.
WARNING:PhysDesignRules:812 - Dangling pin <DOB6> on
block:<ROMHueco/U0/xst_blk_mem_generator/gnativebmg.native_blk_mem_gen/valid.
cstr/ramloop[0].ram.r/s3_init.ram/spram.ram.B>:<RAMB16_RAMB16B>.
```

Ilustración 7: Warnings obtenidos al generar el fichero de programación.

Los 3 primeros warnings son debidos a instanciar las memorias ROM como cajas negras, y el cuarto por haber dejado una salida abierta (de forma intencionada).

El resto de warnings son iguales y hay 30 de ellos para cada una de las ROM (pines <DOB2>, <DOB3>...<DOB31> de cada ROM).

4. Recursos utilizados de la FPGA

The screenshot displays the Xilinx ISE Design Suite interface. The left pane shows the Hierarchy view with the project structure. The right pane shows the Design Summary (Synthesized) report, which includes a Project Status table, a Device Utilization Summary table, and a Detailed Reports table.

Project Status (02/23/2018 - 18:27:27)

Property	Value	Property	Value
Project File:	Trabajo.xise	Parser Errors:	No Errors
Module Name:	flappy_bird	Implementation State:	Synthesized
Target Device:	xc3s100e-5cp132	Errors:	
Product Version:	ISE 14.7	Warnings:	
Design Goal:	Balanced	Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	Timing Constraints:	
Environment:	System Settings	Final Timing Score:	

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	41	960	4%
Number of Slice Flip Flops	48	1920	2%
Number of 4 input LUTs	54	1920	2%
Number of bonded IOBs	4	83	4%
Number of BRAMs	2	4	50%
Number of GCLKs	2	24	8%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Success	02-23-2018 18:27:27			