

Laboratorio de Desarrollo de Hardware

Memoria de prácticas

Plataforma Papillo

Ángel López Santiago

Contenido

Objetivos	2
¿Qué es PAPILIO?	2
¿Qué es ZPUino?	2
Ejercicio 1: Diseñando circuitos básicos sobre la FPGA	2
Ejercicio 2: Controlar el encendido apagado desde el PC vía puerto serie	5
Ejercicio 3: Añadir periféricos	6

Objetivos

Nuestro objetivo es conocer la plataforma PAPILIO y conocer qué se puede hacer desde el entorno Design Lab sobre las placas PAPILLO como diseñar circuitos a nivel de captura de esquemáticos, desarrollar Sketches para ZPUINO o modificar el SoC ZPUINO añadiendo algún un nuevo periférico.

¿Qué es PAPILIO?

PAPILIO es una plataforma de placas de desarrollo hardware abiertas basadas en FPGAs de XILINX.

Estas placas cuentan con placas de expansión adaptadas al conector de expansión de PAPILIO llamadas WINGS.

¿Qué es ZPUino?

ZPUino es un SoC implementado en VHDL (soft core) basado en el microprocesador ZPU de Zylins.

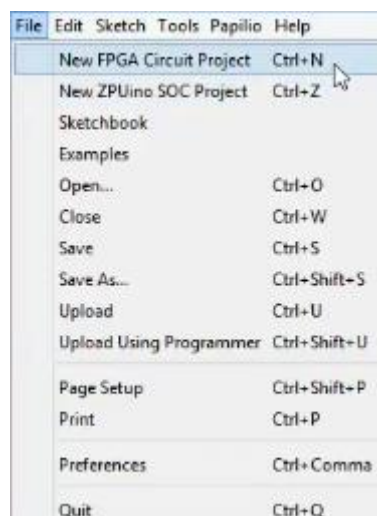
Al ser diseño soft-core, se pueden añadir nuevos periféricos al SoC según las necesidades.

Se puede trabajar con ZPUino de modo equivalente a Arduino ya que se ha adaptado el IDE para ser compatible con ZPUino. Como veremos, el entorno también permitira modificar el SoC a nivel hardware para añadir nuevos periféricos al mismo.

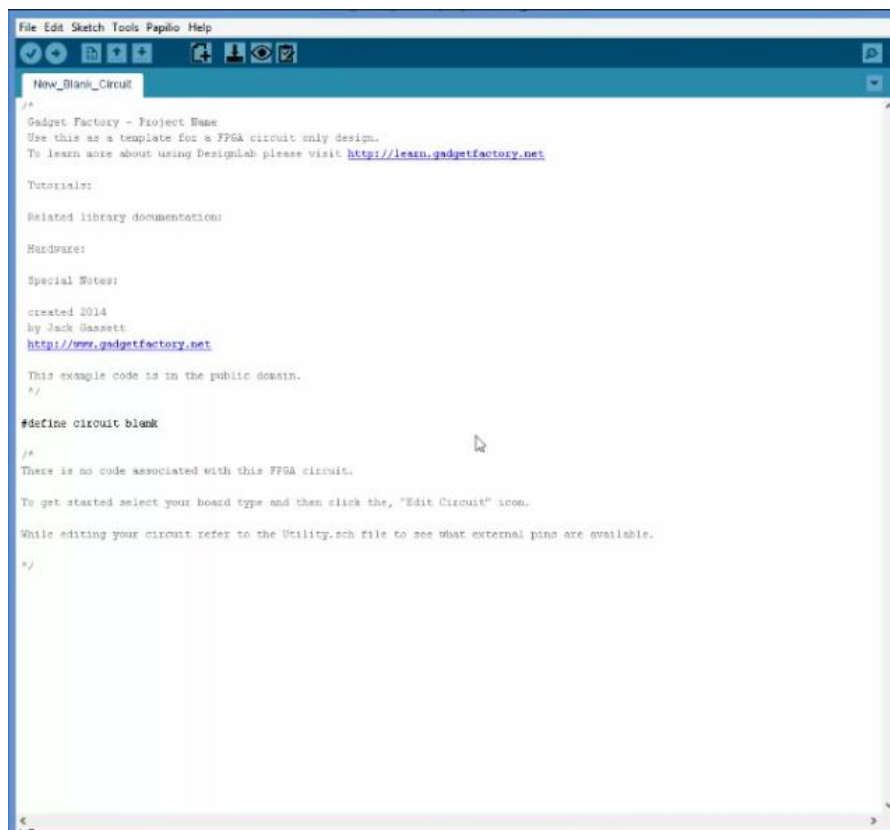
Ejercicio 1: Diseñando circuitos básicos sobre la FPGA

Desde Design Lab es posible diseñar circuitos para implementarlos en la FPGA de la placa de una manera relativamente simple.

Para crear un nuevo proyecto de circuito de FPGA seleccionamos “New FPGA Circuit Project” en el menú FILE.

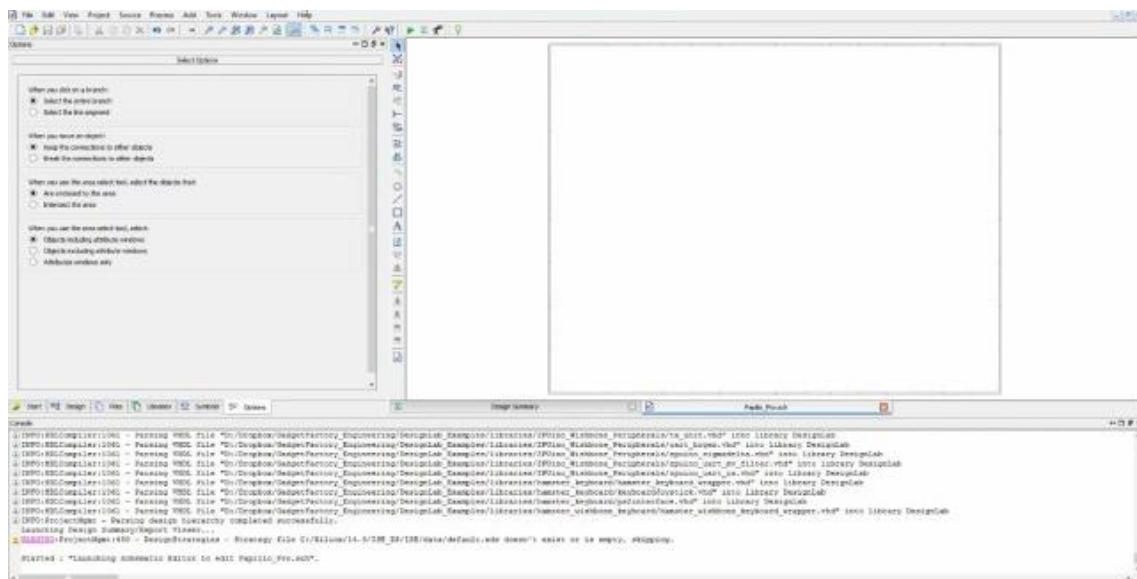


Esto abrirá una ventana con una plantilla de circuito vacío.



Una vez guardada la plantilla dispondremos a editar el circuito. Para ello pulsamos "Edit" para abrir el ISE Schematic Editor.

Una vez abierto ISE Schematic Editor, buscamos un archivo llamado "Papilio_Pro (Papilio_Pro.sch)" y hacemos doble click en él. Esto abrirá el editor esquemático.



Para este primer ejemplo, vamos a diseñar un circuito simple, un inversor.

Para ellos nos dirigimos a la pestaña symbols y seleccionamos el inversor. Una vez seleccionado le conectamos una marca de E/S a cada extremo.

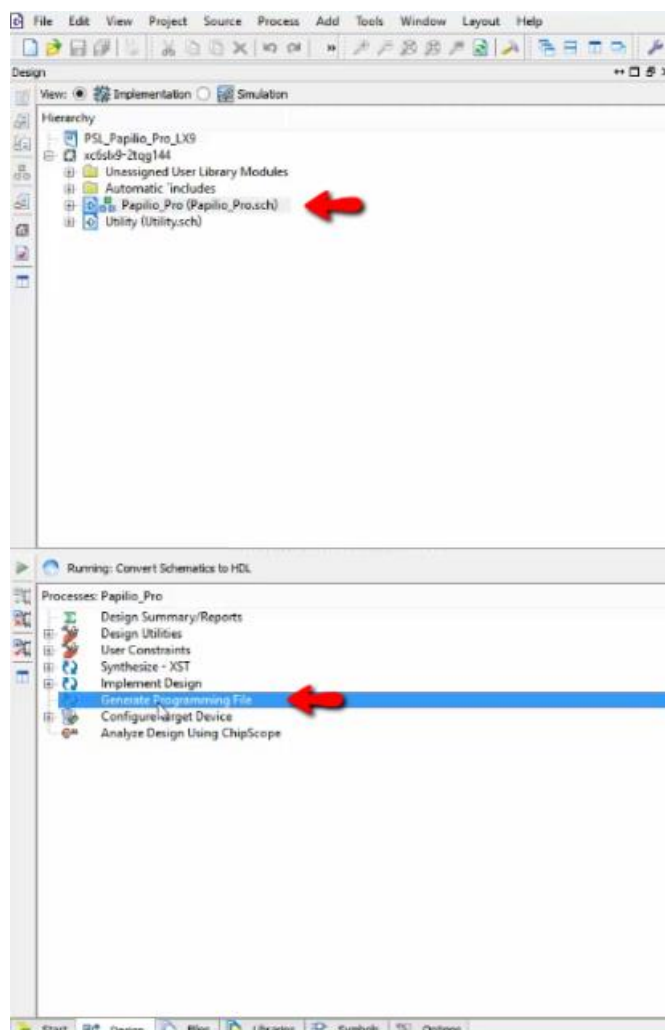


Las marcas de E/S sirven para asignar que pines externos vamos a conectar a la FPGA.

Para este ejercicio asignaremos las entradas Wing_AL0 y Wing_AI1.

Una vez hecho esto procederemos a sintetizar el circuito.

Para ello volveremos a la pestaña Desing, seleccionaremos el archivo Papilio_Pro (Papilio_Pro.sch) y daremos doble click a generate program file.



Para que nuestro circuito se carga correctamente a nuestra placa Papilio, debemos dirigirnos a la carpeta de nuestro proyecto y entrar en el directorio /circuit/500K/, borrar papilio_one_500k.bit y renombrar Papilio_One_500K.bit a papilio_one_500k.bit

Una vez hecho esto volveremos a nuestro proyecto de Papilio y seleccionaremos Load Circuit para cargar nuestro circuito a la FPGA.



Si todo ha salido correctamente, podremos probar nuestro circuito simple. Conectaremos al pin Wing_AI0 la salida de un pulsador y a la salida Wing_AI1 un led.

Si todo sale bien podremos ver que mientras no pulsemos el pulsador el led permanecerá encendido, cuando lo pulsemos, se apagará.

Ejercicio 2: Controlar el encendido apagado desde el PC vía puerto serie

Ejercicio parecido al de la práctica de Arduino.

Para ello usamos este código:

```
HardwareSerial mySerial1 (Wishbone (5));

int led = 13;

void setup(){
    pinMode (led, OUTPUT);
    mySerial1.begin(9600);
}
void loop(){
    if(mySerial1.available()){
        char c = mySerial1.read();
        if (c == 'H'){
            digitalWrite(led, HIGH);
        }else if( c == 'L'){
            digitalWrite(led, LOW);
        }
    }
}
```

También dispones de este código Python

```
import serial
Zpuino = serial.Serial('/dev/ttyUSB0', 9600);
print ("Starting o Inicializando")
while True:
    comando = raw_input('Introduce un comando H o L: ')
    Zpuino.write (comando)
    if comando == 'H':
        print('LED ENCENDIDO')
    elif comando == 'L':
        print('LED APAGADO')
Zpuino.close()
```

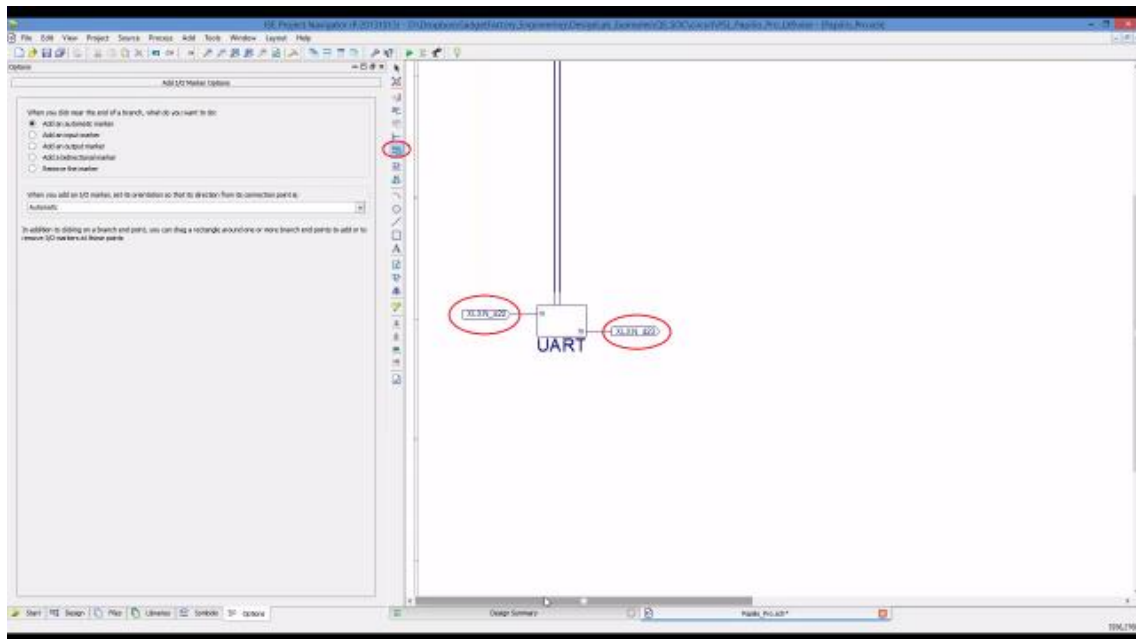
Ejercicio 3: Añadir periféricos

Vamos a añadir una UART a nuestro SoC ZPUino.

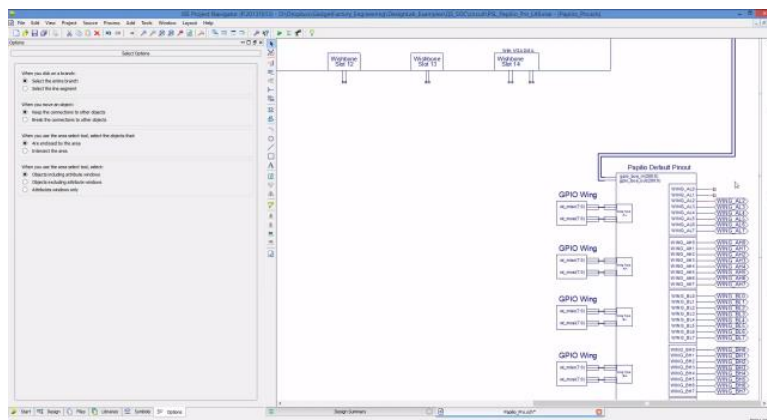
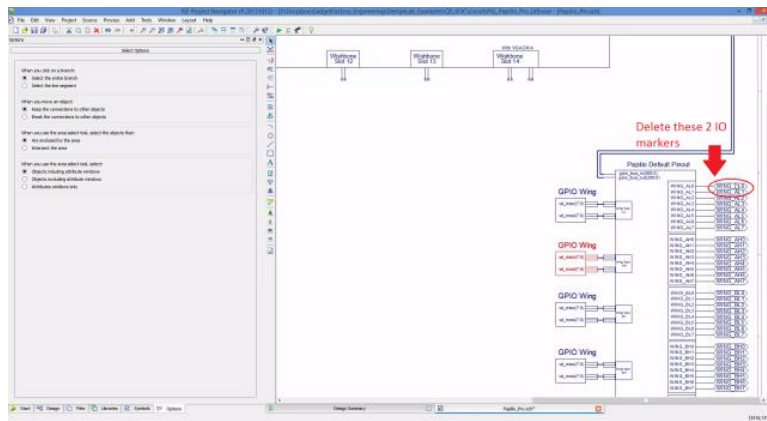
Para ello, hacemos click en New ZPUino SoC Project y procederemos a editar el circuito como en el ejercicio 1.

Una vez abierto el esquemático del circuito en Xilinx ISE Editor, nos dirigiremos a la pestaña symbols y añadiremos la UART al puerto Whishbone 5.

Una vez añadida le asociaremos dos marcas E/S.



Recordando eliminar antes las que vamos a usar.



Una vez sintetizado conectaremos una las salidas seleccionadas un cable TTL-RS232-USB para hacer de puerto serie.



Usaremos los códigos del ejercicio anterior para comprobar su funcionamiento.