

Electrónica Digital. Curso 2019/20

Práctica 2 – El Coche Fantástico

Objetivo de la práctica

El objetivo de la segunda práctica de Electrónica Digital es que el alumno profundice en el diseño e implementación de alguno de los componentes básicos que podemos encontrar en la mayoría de sistemas digitales, tales como divisores de frecuencia, contadores o decodificadores, entre otros. Procederemos además a la implementación de un fichero de *Test Bench* para realizar la simulación del sistema, antes de su descarga en la FPGA, así como un fichero de *restricciones XDC*.

Para ello se plantea el diseño de un sistema que emule el funcionamiento de las luces frontales del famoso *coche fantástico* de Michael Knight, que se muestra en la Figura 1. En este caso, emplearemos los LEDs presentes en las placas de FPGA del laboratorio, tal y como se destaca en la Figura 2.

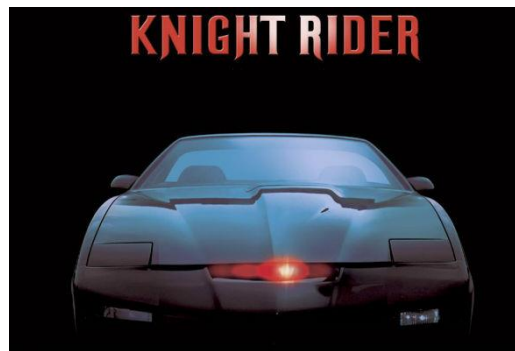


Figura 1: Luces frontales del coche fantástico que se desean emular

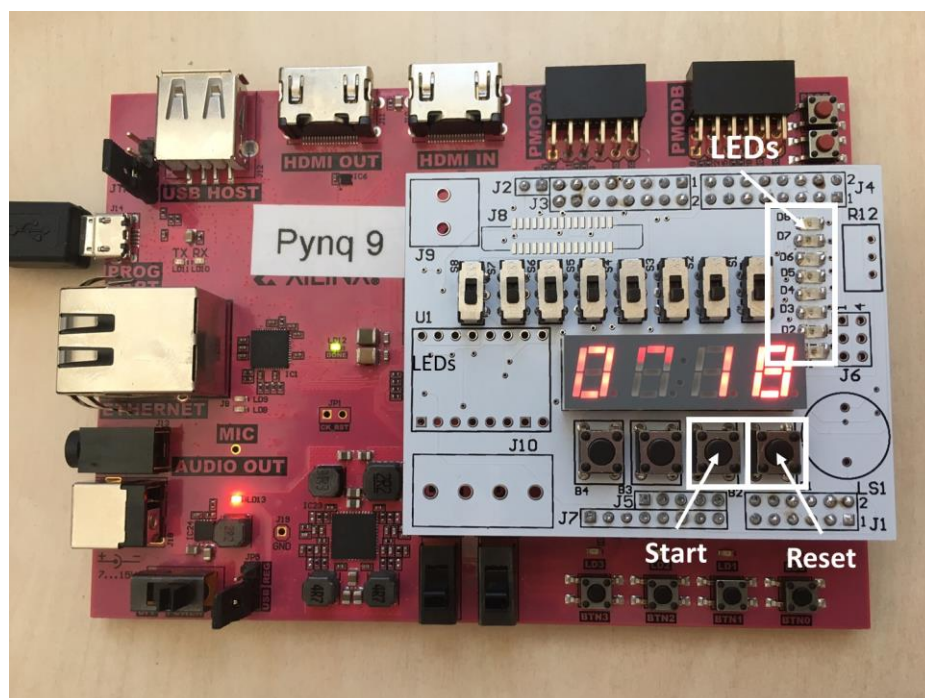


Figura 2: Ubicación de los LEDs en las placas del laboratorio

El funcionamiento esperado es el siguiente:

- Los LEDs deben encenderse primero de abajo a arriba y después de arriba abajo, sin repetición en los extremos.
- Sólo uno de los LEDs debe estar encendido a la vez.
- El tiempo total para recorrer los 8 LEDs, ida y vuelta, será de **1 segundo**.
- El usuario deberá pulsar un botón de la placa (el que se destaca en la Figura 2, etiquetado como *Start*) para indicar al sistema que su funcionamiento debe empezar. Antes de pulsar este botón, los LEDs deberán reflejar el estado de *reset* (sólo el de abajo estará encendido).
- El patrón de funcionamiento del circuito se repetirá indefinidamente, hasta que se pulse el botón de *reset*.

El bloque a diseñar debe tener las siguientes entradas y salidas:

Nombre de la Señal	Dirección	Ancho (bits)
clk	Input	1
reset	Input	1
start	Input	1
leds	Output	8

Simulación del Circuito

Una vez se ha finalizado el diseño, procederemos a su simulación. Para ello tenemos que escribir un fichero de *Test Bench*. La *entity* de un fichero de *Test Bench* tiene la particularidad de no tener ni entradas ni salidas:

```
entity rider_tb is
end rider_tb;
```

En su interior instanciaremos el bloque que queremos simular (*design under test*), tal y como se muestra en la Figura 3.

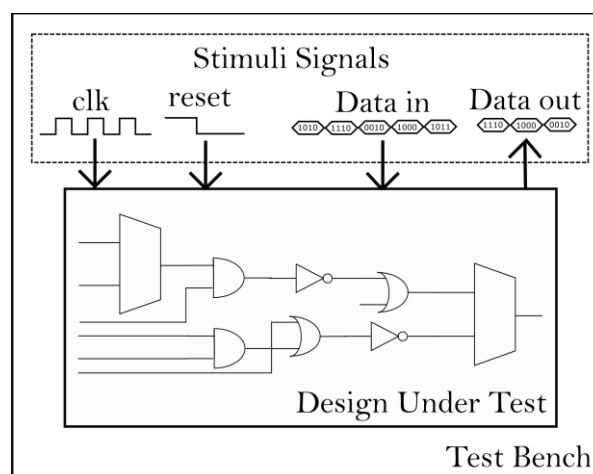


Figura 3: Arquitectura genérica de un fichero de Test Bench

Este fichero deberá contener procesos para simular la generación del reloj y de las señales de control del sistema (señales reset y start).

```
-- Generación del reloj  
  
clk_proc: process  
  
begin  
  
    clk <= '1';  
  
    wait for clk_period/2;  
  
    clk <= '0';  
  
    wait for clk_period/2;  
  
end process;
```

```
-- Generación del reset y boton  
  
stimuli_proc: process  
  
begin  
  
    .....  
  
    wait;  
  
end process;
```

Es importante destacar que los ficheros de *Test Bench* pueden contener procesos dentro de los cuales existen instrucciones VHDL **no sintetizables**, tales como **wait** o **wait for**. Estos procesos no tienen lista de sensibilidad. Estas instrucciones **no se podrán emplear nunca** en nuestros diseños digitales.

Cheatsheet sobre testbenches VHDL

- Entity sin puertos (vacía)
- Procesos sin lista de sensibilidad (se redisparan siempre al llegar al final)
- Construcciones no sintetizables, con concepto de tiempo (wait, wait for)

Los *testbench* son artificios que permiten probar (simulando) nuestros diseños, no especificaciones de circuitos digitales. Es en el único archivo de código de la asignatura Electrónica Digital donde se permite no seguir la norma sobre diseño secuencial síncrono (ver *template* en el guion de la práctica anterior).

Alternativas de diseño

¿Se os ocurren otras formas de implementar el mismo circuito?, ¿Cuáles son las ventajas e inconvenientes de cada arquitectura? Probad a realizar otras implementaciones y simular su funcionamiento.

Implementación sobre la FPGA

Para proceder a implementar nuestro diseño debemos generar un fichero de restricciones XDC, en el que se recojan las asignaciones de los puertos de entrada y salida del bloque diseñado, a pines de la FPGA. En el siguiente cuadro se recogen las asignaciones necesarias para cada una de las placas:

Nombre de la Señal	Dirección	PIN
clk	Input	H16
reset	Input	V17
start	Input	V18
Leds<0>	Output	U5
Leds<1>	Output	V5
Leds<2>	Output	V6
Leds<3>	Output	U7
Leds<4>	Output	T14
Leds<5>	Output	U12
Leds<6>	Output	U13
Leds<7>	Output	V13

Figura 2: Pines a emplear en nuestro diseño para la placa **Pynq-Z1** y la shield de prácticas

A modo de ejemplo, una asignación dentro del fichero XDC tendrá la forma:

```
set_property -dict { PACKAGE_PIN H16 IOSTANDARD LVCMOS33 } [get_ports { clk }];
```

Se recuerda cómo podemos ver el resumen de nuestro diseño, en el que se indica la lógica empleada en cada caso.

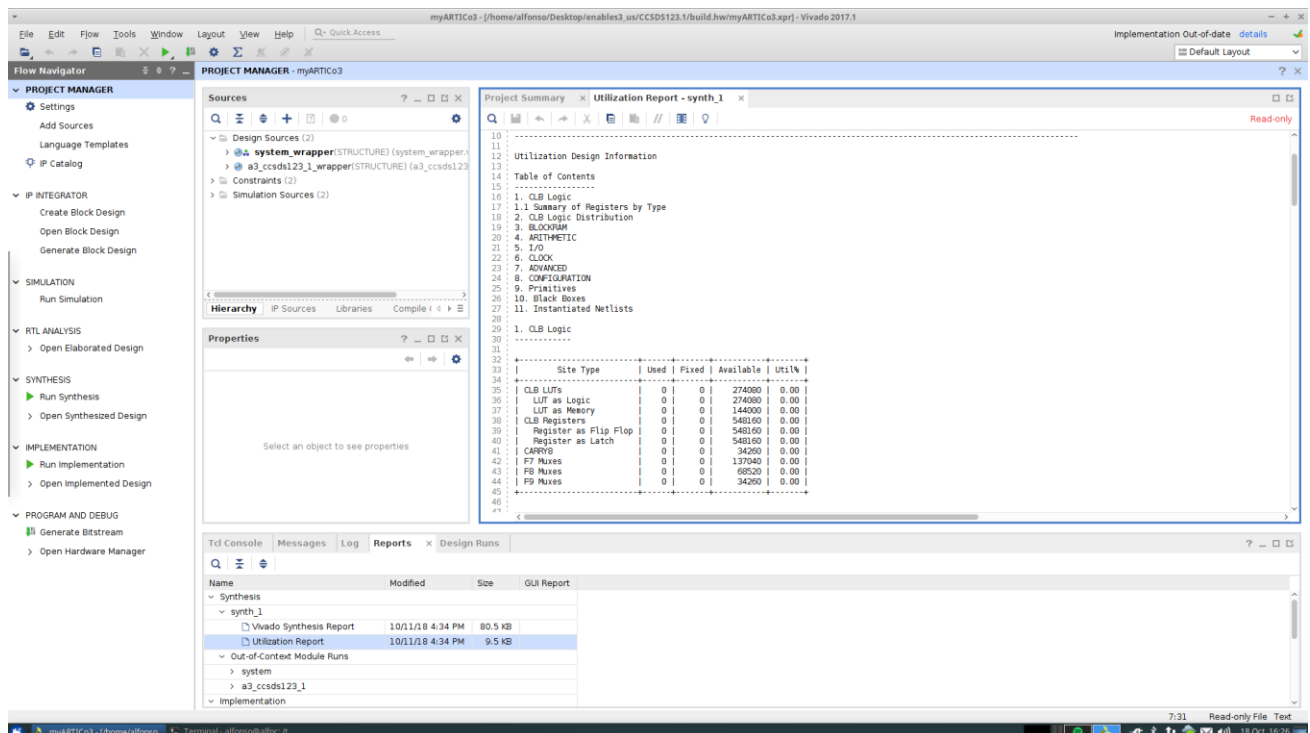


Figura 4. Resumen del diseño