
Alumno: SUÁREZ NIÑO, ALEXANDRO RENÉ

Práctica de Verificación

Curso: 2021-2022

En esta parte de la asignatura el alumno debe realizar, a partir de la especificación de un sistema digital, una verificación funcional completa basada en simulación.

El trabajo constará de los siguientes pasos:

1. Matriz de cobertura funcional, según se describe en teoría:
 - a. Eje Y: Lista exhaustiva de las funcionalidades a verificar, basada en la descripción funcional del sistema. El requisito de esta lista es que, si todos estas funcionalidades son correctas en la implementación del sistema verificado, el sistema funcionará correctamente de acuerdo a su especificación.
 - b. Eje X: Lista de tests necesarios para realizar la verificación de las funcionalidades en eje Y.
 - c. En la matriz se marca, para cada entrada en el eje Y, qué test (eje X) la verifica.

[Entregable: Documento]
2. Descripción de cada test. Se define, para cada test en la Matriz, su funcionalidad: generación de estímulos y comprobación de salidas.
[Entregable: Documento]
3. Descripción del o los testbench, explicando qué estructura tendrá, cómo se van a ejecutar los distintos tests, cómo se vana comprobar y reportar los resultados.
[Entregable: Documento]
4. Desarrollo del código VHDL del/los testbench e implementación de los tests. Sin necesidad de disponer del código RTL del DUT, el alumno realizará simulaciones para comprobar que los recursos de verificación implementados funcionan correctamente.
[Entregable: Código y evidencia de las simulaciones de comprobación el test bench]
5. Ejecutar los tests sobre las distintas versiones del código del DUT (Device Under Test) entregado por el profesor.

- a. Los tests deben ejecutarse de manera autónoma, con un resultado mostrado en consola de PASO/FALLO para cada uno, y para la batería completa de tests.
- b. Cada test incluirá mensajes en consola que ilustren el desarrollo del mismo.
- c. En caso de fallo, un mensaje en consola apuntará a la causa del mismo (qué condición ha provocado el fallo).

[Entregable: Ficheros con la salida por consola resultante de la ejecución de los tests, para cada versión de código suministrada]

Desarrollo del trabajo:

El alumno completará cada paso y entregará el entregable asociado antes de pasar al siguiente.

El profesor está disponible para resolver cuantas dudas se planteen sobre cómo afrontar el trabajo. Pero no se trata de un proceso iterativo en el que el profesor revisa versiones de preliminares de cada entregable para que el alumno lo vaya mejorando implementando las sugerencias del profesor. Si se hiciera así, el alumno no haría un trabajo creativo, sino que simplemente transcribiría las sucesivas indicaciones del profesor.

Para el desarrollo del VHDL del testbench el alumno deberá investigar los distintos recursos que el lenguaje VHDL ofrece para implementar los mecanismos de generación de estímulos y comprobación de salidas como ASSERT, manejo de ficheros de texto etc. El profesor estará disponible para explicar el uso de los mismos.

Para los pasos 1-4 sólo se necesita el presente enunciado con la especificación del sistema. Para el paso 5 recibirá el código VHDL (RTL) del sistema a verificar.

Fechas límite:

Paso 1: 24/11

Paso 2: 30/11

Paso 3: 6/12

Paso 4: 17/12

Paso 5: 10/1

Evaluación:

Para aprobar esta sección de la asignatura será condición necesaria (no suficiente) completar los pasos 1-5.

Para cada paso el entregable debe suficiente calidad y acercarse a su objetivo.

Respecto al test bench, es imprescindible que la ejecución sea autónoma y que detecte varios errores funcionales introducidos en el código del sistema a verificar.

ESPECIFICACIÓN FUNCIONAL

El sistema digital a verificar consiste en un controlador de un periférico de salida (un anillo de LEDs RGB) con como periféricos de entrada dos codificadores rotatorios (Control V y Control I).



El anillo de LEDs muestra un número de LEDs encendidos (entre 1 y 11) que rotan continuamente con color configurable e intensidad y velocidad de giro controlables.

Entradas del sistema a verificar (DUT):

- CONTROL_V_A, CONTROL_V_B: Señales de entrada desde el rotatorio CONTROL V
- CONTROL_I_A, CONTROL_I_B: Señales de entrada desde el rotatorio CONTROL I
- NUM_LED[3:0]: Número de LEDs encendidos.

Salidas del sistema a verificar (DUT):

- LED_DOUT: Dato serie de entrada al anillo de LEDs

El rotatorio “CONTROL V” controla la velocidad y sentido del giro de los LEDs. Tras Reset los leds quedan quietos. Cada paso de giro a la derecha se incrementa la velocidad de giro a la derecha, o se reduce la velocidad de giro a la izquierda. Recíprocamente, cada paso de giro a la izquierda se incrementa la velocidad de giro a la izquierda, o se reduce la velocidad de giro a la derecha.

La frecuencia máxima de giro del anillo es 1Hz. El control de velocidad es lineal, con un valor 0 (estático), 8 valores de giro a la derecha y 8 de giro a la izquierda.

El “CONTROL I” controla la intensidad de iluminación de los LEDs. Tras Reset los LEDs estarán apagados. Cada paso de giro a la derecha incrementa en 1/16 de la escala 0 a 255 la intensidad de brillo (0, 16, 32, ... 240, 255). Cada paso de giro a la izquierda decrementa con los mismos pasos. La intensidad se satura en 255 por arriba y en 0 por abajo.

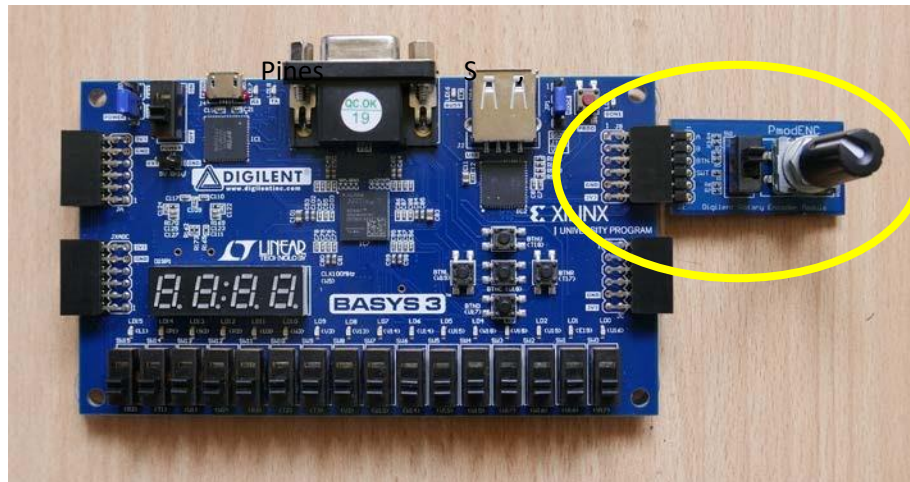
El número de LEDs encendidos se controla por un valor de entrada de 4 bits (NUM_LED). Los valores válidos son entre 0 y 12. “0” mantiene todos los leds apagados, “12” todos los LEDs encendidos

El color de los LEDs es fijo, blanco (R, G y B de igual valor).

Los detalles de los periféricos “Codificador Rotatorio” y “Anillo de LEDs” se incluyen como apéndices. Ambos son periféricos utilizados en el laboratorio de Electrónica Digital II de 2º curso del Grado.

APÉNDICE 1: DESCRIPCIÓN DEL CODIFICADOR ROTATORIO

En la siguiente imagen podéis ver el codificador rotatorio montado en una placa Basys3.



La pequeña placa Pmod tiene los siguientes pines (su ubicación se distingue también en la imagen anterior):

Pin	Signal	Description
1	A	Output of button A in the encoder shaft
2	B	Output of button B in the encoder shaft
3	BTN	Output of the integral push button in the encoder shaft
4	SWT	Output of the on board switch
5	GND	Power Supply Ground
6	VCC	Positive Power Supply (3.3/5V)

De cara a la verificación nuestro sistema sólo usará las señales A y B.

Hay varios tipos de codificadores rotatorios. El que se usa se basa en la generación de dos señales A y B que provienen de dos contactos internos. Cuando se hace girar el mando del codificador, las señales A y B generan una secuencia cuyo orden depende del sentido del giro. Cuando se gira en sentido anti-horario, la primera señal que se activa es la A. En sentido horario será la B.

En el siguiente enlace podéis ver una explicación de funcionamiento. Centraos, fundamentalmente, en la parte que va desde el minuto 1 al 2:40. El que usamos en la práctica es muy similar, salvo que el paso por los contactos A y B provoca un '0' y, cuando no se está girando están a '1'. La posición de los contactos también es la contraria (en el del vídeo, el giro horario activa primero el pin A) <https://www.youtube.com/watch?v=v4BbSzJ-hz4>

En este enlace tenéis un video explicativo con el rotatorio montado en la placa Basys3 (también en

:

https://unicanmy.sharepoint.com/:v:/g/personal/fernandv_unican_es/EfPt_bPYegt9KjwbQ1pi1GT8BIG9Enk0Hcl99VclK1znLw?e=C1c2e8).

Hay dos aspectos que debemos tener en cuenta a la hora de computar pasos de giro:

- El giro en cualquier sentido se puede abortar antes de completarlo (volviendo a la posición de partida) y no debería ser computado. Esto se muestra en el vídeo anteriormente referenciado.
- Las señales A y B son susceptibles de tener rebotes en sus cambios de valor. Ahora bien, cuando una de las señales tiene rebotes la otra está estable. Por lo tanto, evitando el fenómeno explicado en el punto anterior se evita también éste.

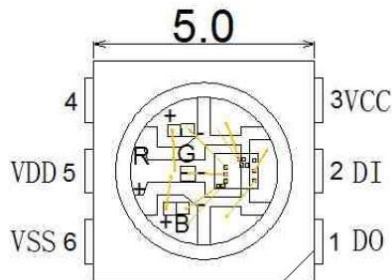
APÉNDICE 2: DESCRIPCIÓN DEL ANILLO DE LEDS

El anillo de led a controlar es el WS2812 de Adafruit. Se trata de un anillo de 12 leds RGB

<https://www.adafruit.com/product/1643> <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>

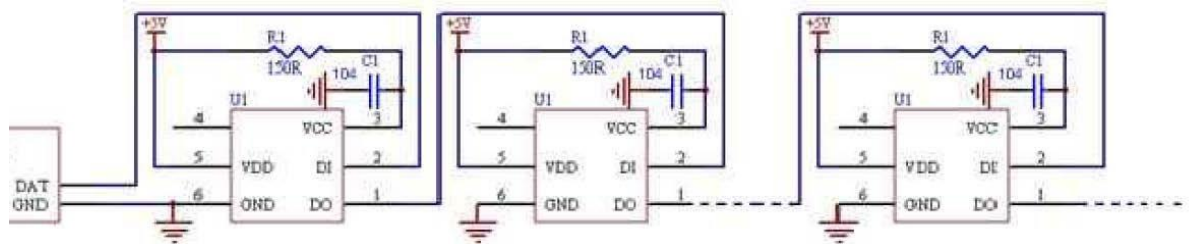


Cada led RGB es un chip con los siguientes pines:



NO.	Symbol	Function description
1	DOUT	Control data signal output
2	DIN	Control data signal input
3	VCC	Power supply control circuit
4	NC	
5	VDD	Power supply LED
6	VSS	Ground

El anillo es, por lo tanto, una placa PCB circular que conecta en serie 12 chips RGB. El conexionado en serie se puede ver en la siguiente figura:

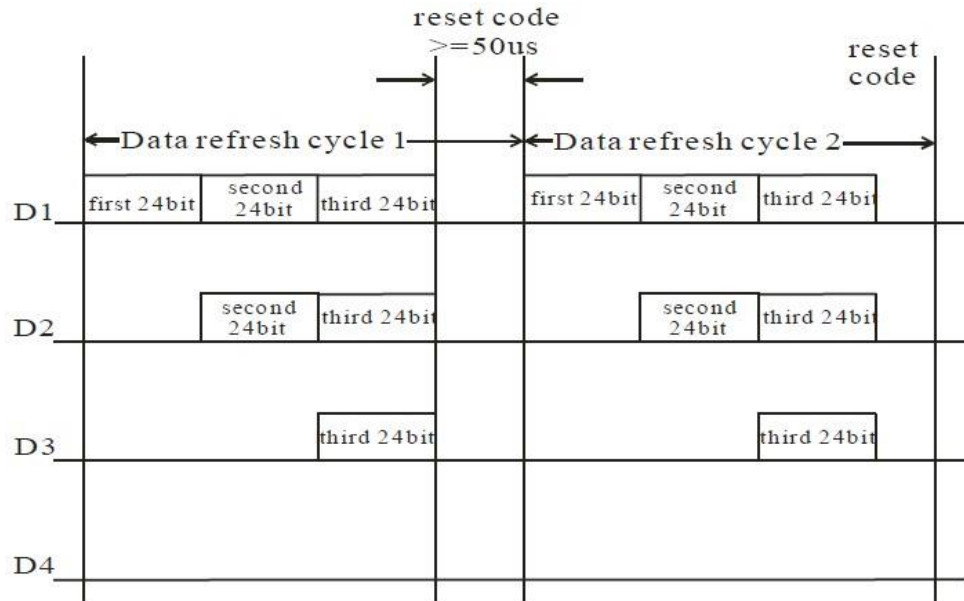


Cada chip RGB contiene un microcontrolador para su control y configuración. Para determinar el color de un led, debemos pasarle 24 bits en el orden (GRB) que se ve en el siguiente esquema:

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Note: Follow the order of GRB to sent data and the high bit sent at first.

Para pasar los colores a los distintos leds se sigue el siguiente protocolo:



Debemos enviar toda la ristra de bits de color. Cada chip RGB se queda con los primeros 24 bits y envía el resto al siguiente. Cuando acaban de recibir datos en su entrada DI y ésta permanece a '0' durante, al menos, $50\mu s$, el chip establece el color indicado. Dejará iluminado ese color hasta que se le indique otro.

Cada bit se indica con un formato PWM con la siguiente forma y valores temporales:

Bit 0	
$T_{0H} = 0.35\mu s$ $T_{0L} = 0.9\mu s$ $T_0 = T_{0H} + T_{0L} = 1,25\mu s$ (fr = 800 KHz)	0 code
Bit 1	
$T_{1H} = 0.7\mu s$ $T_{1L} = 0.55\mu s$ $T_1 = T_{1H} + T_{1L} = 1,25\mu s$ (fr = 800 KHz)	1 code