



Protocolos de Comunicaciones

Trabajo Práctico N° 1

Proyecto de Leds

Alumnos:

Taborda Galeano, Andrea Elizabeth

Docente:

Finochietto, Jorge

Sposetti, Facundo

Año 2025

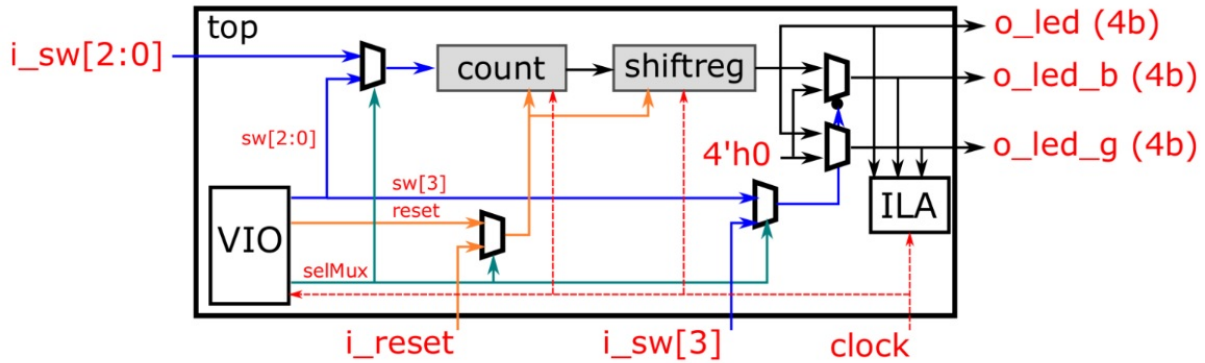
Índice

Ejercicio 1

2

Enunciado

Implementar el siguiente circuito en verilog



- Incluir los módulos VIO e ILA para control y debug del módulo Top.
- Los nombres en **ROJO** son puertos.
- i_reset es el reset del sistema, el cual pone a cero el contador e inicializa el shift register (SR).
- $i_sw[0]$ controla el enable (1) del contador. En estado cero (0) detiene el funcionamiento sin alterar el estado actual del contador y del SR.
- El SR se desplaza únicamente cuando el contador llegó a algún límite R0-R3. La selección del límite se realiza con la señal $i_sw[2:1]$.
- La elección del límite se puede realizar en cualquier momento del funcionamiento.
- $i_sw[3]$ elige el color de los leds RGB.

Ejercicio 1

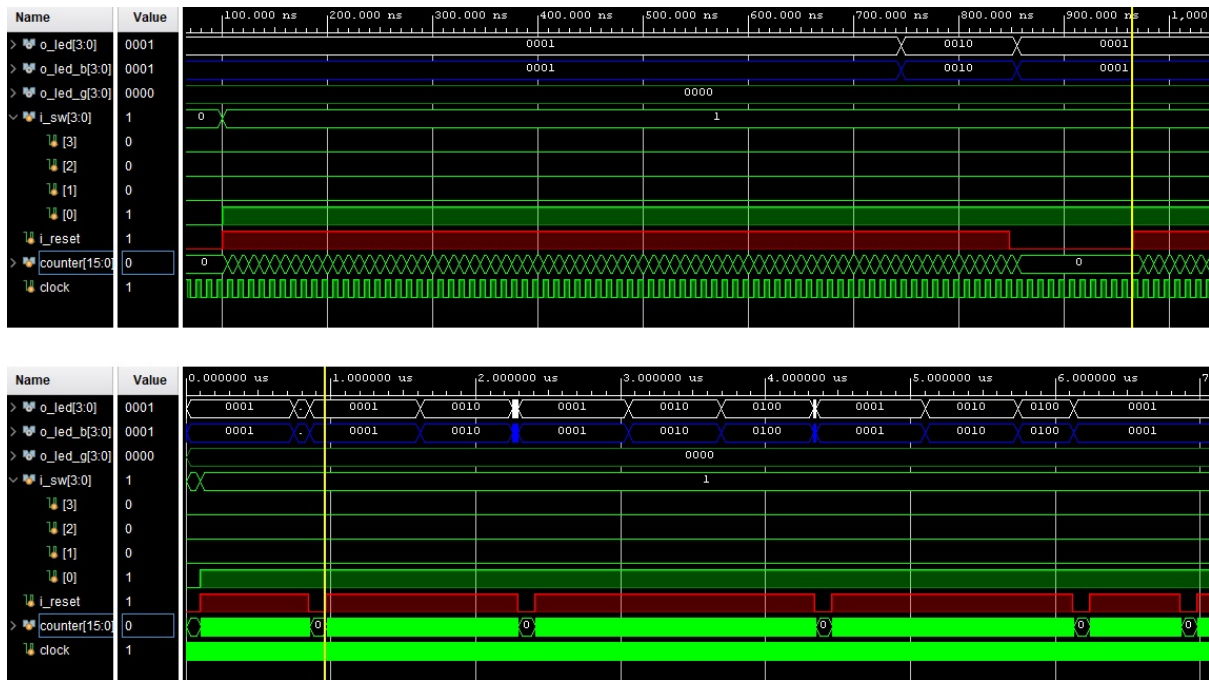
En este trabajo se implementaron distintos test con el objetivo de verificar de forma específica el comportamiento de diversas señales involucradas en el funcionamiento del sistema completo.

Test 1

Este test tiene como objetivo verificar el comportamiento esperado de i_reset , el cual debe poner el contador en cero e inicializar el *shift register*.

El test comienza configurando la velocidad R0 y utilizando los LEDs azules. Luego de salir del estado de reset, se habilita el contador. A intervalos aleatorios, se aplica un

nuevo reset para verificar el correcto reinicio del sistema. Esto se comprueba observando el valor de `o_led`, que debe ser 0001 inmediatamente después del reset; de no ser así, el test se considera fallido.



Como se observa, una vez que la señal de reset (en rojo) deja de aplicarse, el contador vuelve a cero y el *shift register* retorna a su valor inicial. Este comportamiento confirma que ambos módulos responden correctamente al evento de reinicio. El proceso se repite cinco veces a lo largo del test, con resets aplicados en momentos aleatorios, para validar la robustez y la consistencia del sistema ante múltiples eventos de reinicio.

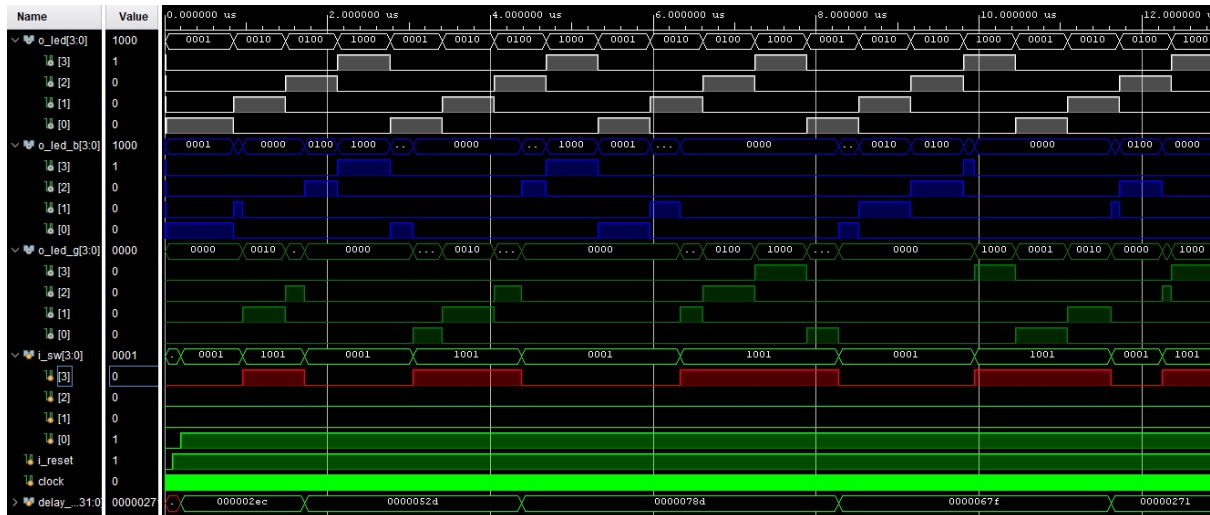
Al finalizar la simulación, se imprime un mensaje por consola que indica el resultado del test, con el fin de evitar una verificación manual del comportamiento de la señal.

```
Corriendo test...
Corriendo test...
Corriendo test...
Corriendo test...

=====
TEST 1 PASSED: Reset correcto, o_led = 0001
=====
```

Test 2

En este test se verifica el correcto funcionamiento del selector de color de salida. Cuando se activan los LEDs azules (`o_led_b`), los LEDs verdes (`o_led_g`) deben permanecer apagados, y viceversa. Para comprobarlo, se alterna entre ambos colores en instantes de tiempo aleatorios, controlando que se encienda únicamente el color seleccionado y que el otro permanezca inactivo durante toda la transición.



Como se aprecia en la imagen, cuando $i_sw[3]$ esta en bajo, la secuencia se refleja unicamente en los LEDs azules. Por otro lado, si $i_sw[3]$ esta en alto, son los LEDs verdes son los que cambian, permaneciendo los azules inactivos.

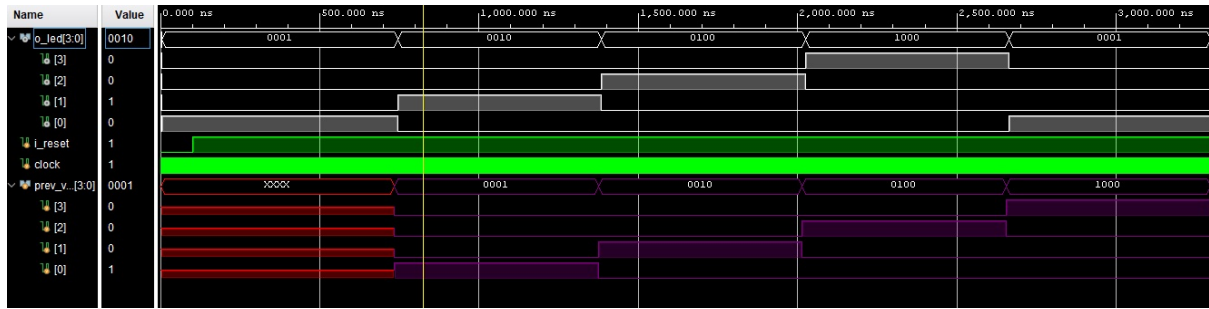
Al igual que en el test anterior, para independizarse de la verificación manual de las señales, al finalizar la simulación se muestra un mensaje automático con el resultado obtenido.

```
Corriendo test...
Corriendo test...
Corriendo test...
Corriendo test...
Corriendo test...
Corriendo test...
Corriendo test...
Corriendo test...
Corriendo test...

=====
TEST 2 PASSED: Cambio de color correcto
=====
```

Test 3

El objetivo de este test es verificar que los LEDs se desplacen correctamente siguiendo la secuencia: $0001 \rightarrow 0010 \rightarrow 0100 \rightarrow 1000$, y que esta se repita cíclicamente. Para ello, se utiliza una variable auxiliar (también visible en la simulación) que almacena el valor anterior de la señal de salida. Esta variable permite comparar cada nuevo valor con el anterior, asegurando que el desplazamiento ocurra en la dirección esperada y respetando el orden definido por la secuencia.



Una vez que el contador alcanza la cantidad de ciclos correspondiente a la velocidad configurada (en este caso R0), se realiza la verificación del comportamiento esperado. Si en cualquier paso la transición no coincide con el valor esperado, el test falla inmediatamente. Si todas las transiciones son correctas, el test finaliza con éxito.

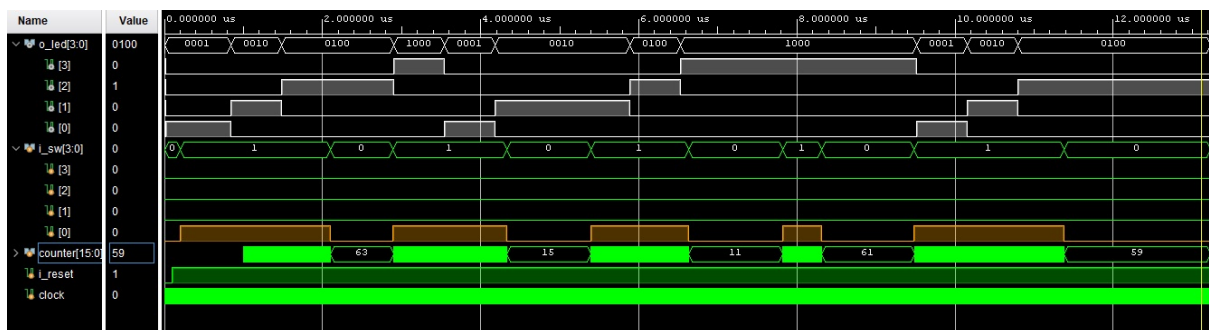
```
Paso 0 OK: 0010
Paso 1 OK: 0100
Paso 2 OK: 1000
Paso 3 OK: 0001

=====
TEST 3 PASSED: Corrimiento correcto a velocidad R0
=====
```

Test 4

Este test tiene como objetivo verificar el correcto funcionamiento de la señal i_sw[0], encargada de habilitar o deshabilitar el contador. Durante la prueba, se alterna de forma aleatoria entre habilitar y deshabilitar el contador, utilizando i_sw[0], mientras se monitorea el estado de la salida o_led.

Tal como se observa en la imagen, o_led solo avanza cuando i_sw[0] se encuentra en alto. Si se detectara una variación sin que el contador esté habilitado, el test falla.



Al finalizar la simulación, se muestra un mensaje con el resultado obtenido.

```

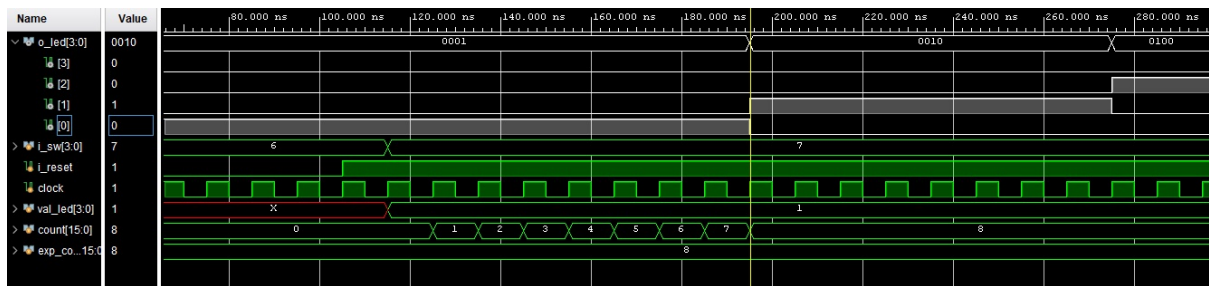
Corriendo test...
Corriendo test...
Corriendo test...
Corriendo test...
Corriendo test...

=====
TEST 4 PASSED: i_sw[0] controla el contador
=====

```

Test 5

Este test consiste en validar que el contador genere un cambio en la secuencia de salida (o_led) exactamente después del número de ciclos determinado por la velocidad seleccionada (R0 a R3). Se inicializa el sistema, se habilita el contador con i_sw[0], y se cuenta cuántos flancos de reloj se necesitan hasta que se produce un corrimiento de un bit a la izquierda en o_led. El test compara este valor con el esperado y muestra por consola si el resultado es correcto o no.



```

=====
TEST 5 PASSED: Corrimiento tras 8 ciclos
=====

```