A black background with red text

Description automatically generated



**DISEÑO AUTOMÁTICO DE SISTEMAS FIABLES ALUMNOS:**

**IGNACIO MURUBE CREGO**

**JUAN MANUEL VICENTE MARTINEZ**

**JESUS NAVAS MARTIN**

**PROFESOR:**

**IGNACIO AZNÁREZ RAMOS**

Práctica 3: Verificación de sistemas mediante simulación con archivos I/O

**Tabla de contenidos**

* **Introducción 3**
* **Cambios 3**
  1. **Corrección practica 1**
  2. **tb\_top**
* **Solución. 6**
* **Conclusión 7**

# Introducción

El objetivo de esta práctica manejamos la verificación de un sistema a través la lectura y escritura de ficheros, mediante la función asserts, la cual se enfocará en comprobar los resultados.

Los dos ficheros que usaremos input.txt (leerán los valores de entrada de BTNC y rst\_n, así como el valor esperado de la salida LED y el tiempo a esperar hasta la próxima instrucción) y el output (en el cual se escribirán los casos probados, incluyendo el tiempo de delay, las entradas y salidas o los errores si los hubiera).

# Cambios

Los cambios establecidos en esta práctica:

**Corrección practica 1**

Como nos fue indicado, el funcionamiento del bloque top\_practica1 era incorrecto debido a que el LED de salida no estaba respondiendo correctamente con un funcionamiento ‘toggle’ a las pulsaciones validadas de la señal ‘BTN’ de entrada al sistema. Este error resultaba en que el LED no permanecía encendido cuando se dejaba de pulsar el botón, como se puede observar en la siguiente imagen:

Interfaz de usuario gráfica, Escala de tiempo

Descripción generada automáticamente

Cómo fue indicado en los comentarios de la corrección realizada por el profesor, se presentaba un error de concepto en cuanto al funcionamiento de la FSM presente en el bloque interno ‘debouncer’. Por lo tanto, tras modificar los estímulos de prueba presentes en tb\_top y conectar la señal interna ‘timer\_debounce’ al mapa de conexiones genérico del UUT se pudo solucionar satisfactoriamente el error dando como resultado el funcionamiento tipo interruptor esperado en el LED, como se observa en la siguiente imagen:

Interfaz de usuario gráfica

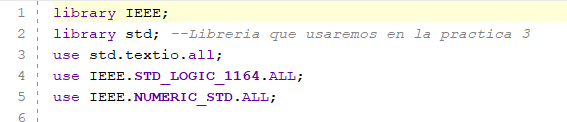
Descripción generada automáticamente

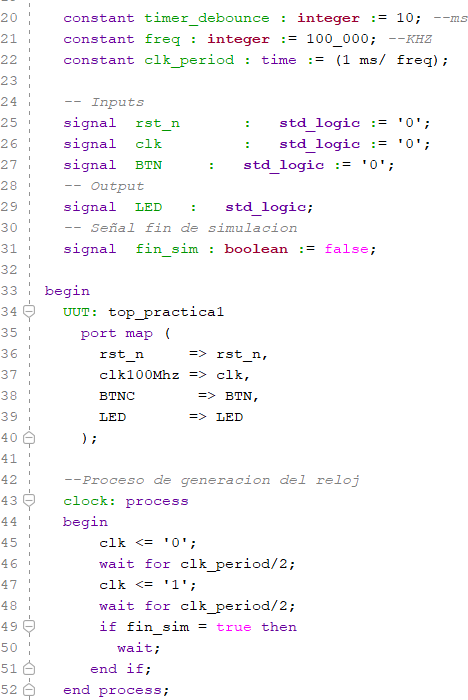
En esta simulación se ha establecido el tiempo de validación de pulsación en 1[ms], adicionalmente se han separado los casos estudiados para su fácil visualización y se ha añadido el cuarto caso de testeo de falsa pulsación de encendido al final.

Casos de prueba:

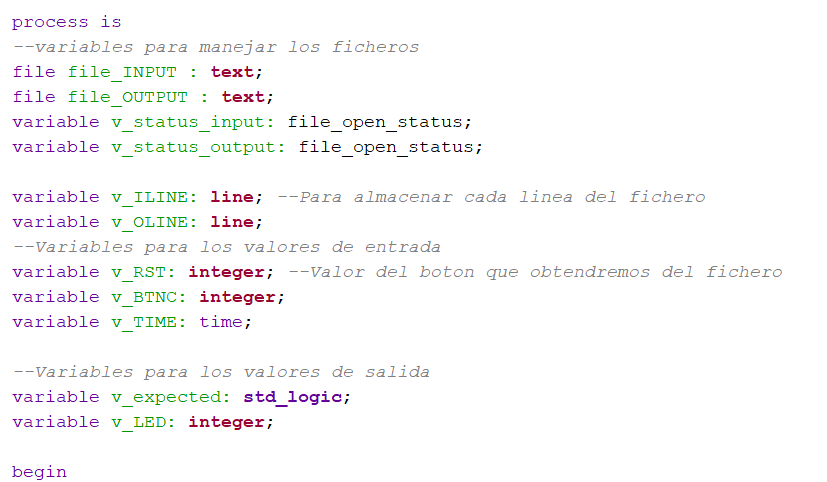
1. **“BTN on with noise”:** Se pulsa el botón y se incluye un ruido intermedio durante el transcurso del tiempo de validación de la pulsación para finalmente mantenerlo pulsado para encender el LED de salida.
2. **“False BTN off”:** Se realiza una breve pulsación del botón y se suelta antes del tiempo de validación para modelar el caso de una falsa pulsación de apagado del LED.
3. **“BTN off with noise”:** Se pulsa el botón y se incluye un ruido intermedio durante el transcurso del tiempo de validación de la pulsación para finalmente mantenerlo pulsado para apagar el LED de salida.
4. **“False BTN on”:** Se realiza una breve pulsación del botón y se suelta antes del tiempo de validación para modelar el caso de una falsa pulsación de encendido del LED.

**tb\_top**

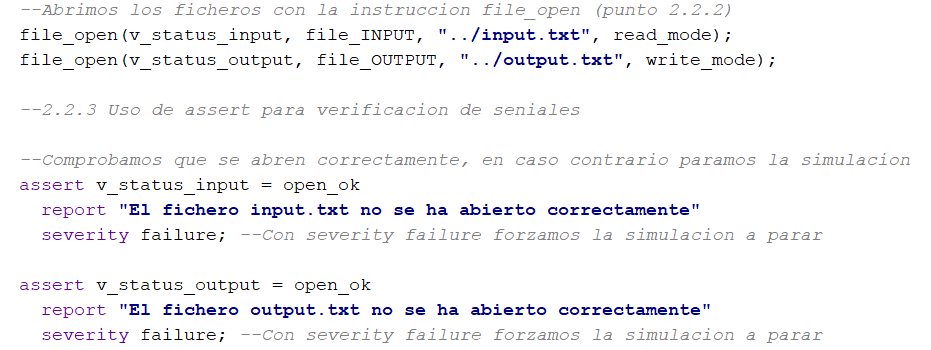
Añadiremos el paquete textio de la librería std para así poder manejar los ficheros y leer y escribir en ellos.

Continuando justo como en la práctica 1, ahora declaramos la entidad, las constantes y señales necesarias, junto con el mapa del componente top\_practica1 que hicimos en la práctica 1, y la creación de la señal de reloj:

Empezaremos un nuevo proceso en el que declararemos las variables necesarias para manejar los ficheros de input y output, además de las variables para leer y escribir los:



Empezaremos a manejar la apertura y escritura de los ficheros correspondientes. Para ello, se utiliza la función file\_open (para abrir los ficheros) y assert (para verificar si se abrieron correctamente con la señal de estado. Si no es así, enviara un reporte, y con failure se fuerza la simulación a parar). Tras esto, con los procesos write y writeline escribiremos en el output



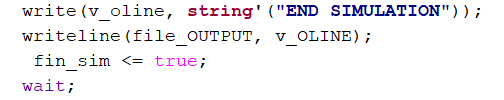
Dentro de un bucle while mientras no se llegue al final del fichero, se vayan leyendo todas las líneas del input gracias a la función readline, para luego transformar de los valores leídos a unos valores deseados.

A screenshot of a computer code

Description automatically generated

Al concluir el proceso, generamos un informe en el resultado con los datos recolectados. Dentro de este mismo proceso, realizamos una comprobación con un "assert" para determinar si el valor esperado del LED es diferente del valor actual. Si coinciden, registramos el valor del LED, si no es así, emitimos un mensaje de error.

Para terminar, escribimos un mensaje para indicar la finalización de la simulación.



# Solución

Tras ejecutar el proyecto con los cambios establecidos mostraremos los resultados obtenidos desde el fichero input.txt al output.txt

En el input:

A screenshot of a computer screen

Description automatically generated

En el output:

A screenshot of a computer

Description automatically generated

Podemos observar que en el output no hay ningún error por lo que hemos obtenido lo que se esperaba obtener.

Resultado de la simulación:

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza media

Resultados con errores:

A continuación, haremos que nos den errores (solo para ver si funciona lo de escribir los errores), para eso haremos que creyésemos que el led no se enciende nunca.

Input.txt:  
A screen shot of a computer

Description automatically generated

# Output.txt:

# A screenshot of a computer program Description automatically generated

# Como podemos observar como hemos hecho que el valor esperado siempre sea 0 cuando el LED se enciende da un error porque no es el valor que esperábamos.

# Conclusión

En conclusión, hemos aprendido a manejar ficheros de texto para el testeo de sistemas introduciendo estímulos y generando informes en estos a partir de los resultados una simulación.