

## APUNTES PARA LA VERIFICACIÓN

### Ejercicio a realizar:

- a) Uno o varios test bench que implementan, uno a uno, cada uno de los tests definidos en la matriz de verificación (*Excel – tabla: “test verificación”*)
- b) Cada test bench tiene que generar los estímulos que implementan la descripción dada de cada test
- c) Y cada test tiene que, de manera concurrente con la generación de estímulos, comprobar que las señales de salida del DUT corresponde con los valores esperados (*Excel – tabla: “Comportamiento deseado”*), tanto sus valores como el momento en el que ocurren.
- d) Esa comprobación de los resultados de valores ha de mostrar en consola el resultado PASS/FAIL de cada test, es decir, que aparezcan todas por consola a la vez (*luego, indicarlo en Excel – tabla: “registro implementación test”*)

### Notas de ayuda para el código:

1. DUT: Device Under Test
2. Sobre la definición de tests que tienes (*Excel: “test verificación”*), codificar el test bench. Recuerda: tests autónomos que reportan el resultado PASS/FAIL porque comprueban las salidas del DUT.
3. Sobre el test bench ya terminado, tendrás que probar con DUT's con distintos bugs introducidos, para comprobar su eficacia.
4. Un test bench puede ejecutar un número de tests secuencialmente, uno tras otro.
5. La generación de estímulos, intenta utilizar bucles para hacerlo de manera más escalable, que puedas ampliar los tests con más eficacia.
6. Podemos poner un proceso para cada test
7. Para test de introducir monedas, hacer uso de secuencias (funciones) y así, poder aplicarla varias veces.
  - a) Puedo escribir un procedimiento (función) que le pasamos como parámetro el valor de la moneda.
  - b) El procedimiento lo realiza de manera automática el wait, asignación de la señal c (introducir moneda).
  - c) Añadimos un bucle para trabajar con los valores de precio:
    - i. Uso de arrays para asignar valores de precio (guarda estados de caracteres)
    - ii. Uso de contadores.

### Ayuda web:

1. Google: "example self checking test bench"( ejemplo: <https://vhdlwhiz.com/how-to-create-a-self-checking-testbench/> )
2. El ejemplo está probando un codificador y se centra en probar todas las combinaciones de entrada para ver el resultado. Mi trabajo consiste en una máquina de estados.
3. El ejemplo hace una referencia del concepto de generar estímulos en un proceso y comprobar las salidas de tu DUT en otro proceso, reportando en consola si el resultado es correcto o no.
4. Pero no puedes aplicar ese código a tu ejercicio.