TEC | Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica

Verificación Funcional de Circuitos Integrados

Proyecto 3

Profesor:

Ronny García Ramírez

Integrantes:

Susana Astorga Rodriguez

Mac Alfred Pinnock Chacón

II Semestre 2021

Cartago, Costa Rica

Índice

Plan de pruebas (Test plan)

GitHub: https://github.com/astorga02/Proyecto3_VFCI

La prueba se corre en EDA Playground o en VCS

En la misma ruta de archivo en donde abre el documento “LEERME.txt”, encontrará 2 carpetas: "Para_EDA" y "Para_VCS" estas carpetas difieren en lo siguiente para poder ejecutarlos.

Para_EDA:

- Solo se puede ejecutar en EDA Playground.
- Para ejecutar cada escenario (2 en total) debe hacerlo manualmente, cambiando la línea 42 por el número de escenario a ejecutar :
  "run_test("test_escenario2"); //se coloca el nombre del test a probar (test_escenario1 o test_escenario2)"

Para_VCS:

- Solo se puede ejecutar en VCS
- Para ejecutar la prueba solo es necesario abrir la terminal en la raíz de los archivos de la carpeta y escribir ./correr.sh y dar enter, la prueba finalizara mostrando la cobertura de la prueba y también puede acceder al reporte.csv. Además de poder consultar la salida de la terminal para verificar cualquier problema con el DUT.

Módulos a utilizar:

- monitor.sv
- design.sv
- agent.sv
- scoreboard.sv
- environment.sv
- driver.sv
- interface.sv
- sequence.sv
- sequence_item.sv
- test.sv
- testbench.sv

Caso de uso común:

Aleatorizando:

- Eventos de datos de entrada.
- Eventos de modo de redondeo.

Casos de esquina:

- Datos de entrada que cubran todos los bits.
- Overflow.
- Underflow.
- No es un número (NaN)
- Caso infinito (inf).

Estructura del ambiente

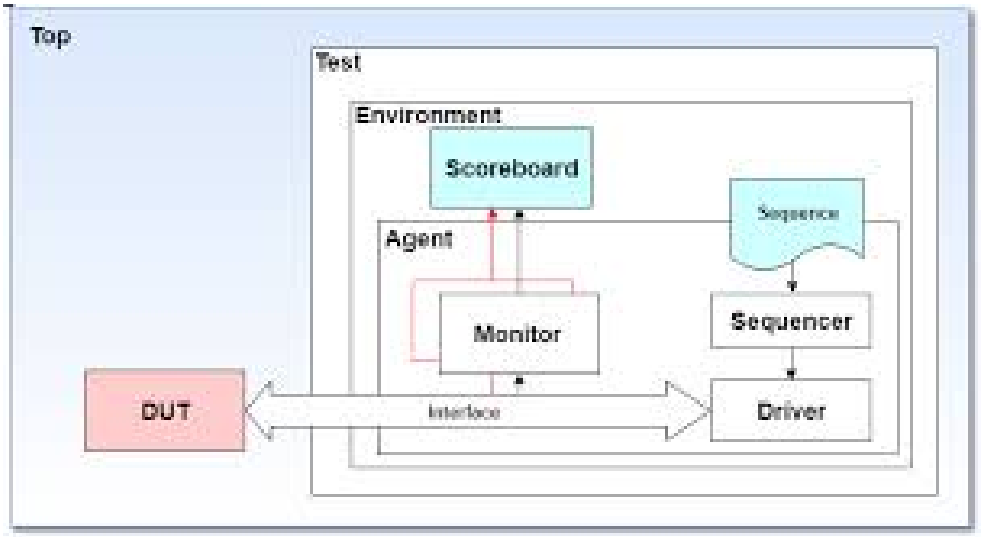


Figura 1. Estructura del ambiente.

Pruebas para los diferentes escenarios

Escenario 1: Envío de mensaje con instrucción del tipo aleatorio
Descripción del test
Test 1: Envío de mensajes tipo aleatorio **Objetivo:** Verificar el correcto funcionamiento de envío de mensajes aleatorios en el DUT y que su la multiplicación de los valores de entrada sea correcta. **Descripción:** En este test se envía un número de 100 transacciones con valores de entrada aleatorios que no sobrepasen las capacidades del DUT en cuanto a overflow, underflow y no ser un número, esto debido a que se dejan estos casos como objetivos para tests de caso de esquina. Las transacciones se enviarán con retardos aleatorios independientes para cada uno de los dispositivos. **Criterio de aprobación:** Se considera que el test es exitoso si la multiplicación de los valores de entrada es correcta comparándolo con un dispositivo simulado que simula el correcto funcionamiento del DUT. Se espera una línea indicando “Todo correcto” como salida inmediata para cada verificación. **Reportes:** Al final de la prueba se imprimirá un reporte con el contenido de cada una de las operaciones (entradas y salida)
Escenario 2: Comportamiento del DUT en casos de esquina
Descripción del test
Test 2.1: Datos de entrada que cubran todos los bits

Objetivo: Cubrir todos los bits que es capaz de operar el DUT

Descripción:
En este test se enviarán valores para una máxima alternancia de bits con una cantidad de 100 transacciones de estos valores.
Las transacciones se enviarán con retardos aleatorios.

Criterio de aprobación: Se considera que el test es exitoso si la multiplicación de los valores de entrada es correcta comparándolo con un dispositivo simulado que simula el correcto funcionamiento del DUT. Se espera una línea indicando "Todo correcto" como salida inmediata para cada verificación.

Reportes: Al final de la prueba se imprimirá un reporte con el contenido de cada una de las operaciones (entradas y salida)

Test 2.2: Overflow

Objetivo: Verificar el funcionamiento de envío de mensajes con el caso de overflow para los valores multiplicados por el DUT.

Descripción:
En este test el DUT tendrá valores de entrada con valores que vayan a causar un overflow en el resultado de la operación. De este caso se envía un número de 100 transacciones.
El contenido de los datos de transacción será totalmente aleatorio pero que se encuentren fuera del rango de operación del DUT. Las transacciones se enviarán con retardos aleatorios independientes.

Criterio de aprobación: Se considera que el test es exitoso si la multiplicación de los valores de entrada da como resultado la activación de la bandera de overflow y un valor infinito a la salida del DUT. Se espera una línea indicando "Todo correcto" como salida inmediata para cada verificación.

Reportes: Al final de la prueba se imprimirá un reporte con el contenido de cada una de las operaciones (entradas y salida)

Test 3.3: Underflow

Objetivo: Verificar el funcionamiento de envío de mensajes con el caso de underflow para los valores multiplicados por el DUT.

Descripción:

En este test el DUT tendrá valores de entrada con valores que vayan a causar un underflow en el resultado de la operación. De este caso se envía un número de 100 transacciones.
El contenido de los datos de transacción será totalmente aleatorio pero que se encuentren fuera del rango de operación del DUT. Las transacciones se enviarán con retardos aleatorios independientes.

Criterio de aprobación: Se considera que el test es exitoso si la multiplicación de los valores de entrada da como resultado la activación de la bandera de underflow y un valor de 0 a la salida del DUT. Se espera una línea indicando "Todo correcto" como salida inmediata para cada verificación.

Reportes: Al final de la prueba se imprimirá un reporte con el contenido de cada una de las operaciones (entradas y salida)

Test 3.4: No es un número

Objetivo: Verificar el funcionamiento de envío de mensajes con salida "NaN".

Descripción:

En este test se envía un número de 100 transacciones de las cuales se hacen asignando los valores de entrada para que sean cero para la entrada 1 e infinito para la entrada 2 y viceversa
El orden de envío será totalmente aleatorio en donde cero e infinito se asignan aleatoriamente a las entradas con la condición de que las entradas no pueden ser a la vez

cero o a la vez infinitas.

Criterio de aprobación: No existe problema con la verificación de que todo sea NaN en ese caso de esquina. Se espera una línea indicando “Todo correcto” como salida inmediata para cada verificación.

Reportes: Al final de la prueba se imprimirá un reporte con el contenido de cada una de las operaciones (entradas y salida).

Test 3.5: Infinito

Objetivo: Verificar el funcionamiento de envío de mensajes con salida “inf”.

Descripción:
En este test se envía un número de 100 transacciones de las cuales se hacen asignando los valores de entrada para representar el escenario de un resultado infinito
El orden de envío será totalmente aleatorio en donde cero e infinito se asignan aleatoriamente a las entradas con la condición de que las entradas no pueden ser a la vez cero o a la vez infinitas.

Criterio de aprobación: No existe problema con la verificación de que todo sea inf en ese caso de esquina. Se espera una línea indicando “Todo correcto” como salida inmediata para cada verificación.

Reportes: Al final de la prueba se imprimirá un reporte con el contenido de cada una de las operaciones (entradas y salida)

Solución del error del DUT

Observando los operandos multiplicando y multiplicador se pudo notar que los errores se daban en dos circunstancias específicas: Cuando uno de los operando presentaba underflow y el otro no y cuando uno de los operando presentaba overflow y el otro no. En el primer caso se obtiene un resultado con underflow pero sin generación de bandera de underflow, mientras que en el segundo de manera similar se obtiene un resultado con overflow pero sin la debida generación de la bandera de overflow.

Para solucionarlo se agrega en el DUT en la sección de código de la figura 1, donde se daba la generación de las señales de overflow y underflow, la funcionalidad de poder generar estas señales cuando alguno de los dos operandos presenta esta condición.

```
module EXP(              //module con cambios agregados
  input norm,
  input [7:0]exp_X, exp_Y,
  output [7:0]exp_Z,
  output ovrf, udrf);

  wire [8:0]buffer;
  wire over_X, over_Y, under_X, under_Y;

  assign buffer = exp_X + exp_Y;

  wire [7:0]bias;

  assign over_X = &exp_X;
  assign over_Y = &exp_Y;
  assign under_X = ~|exp_X;
  assign under_Y = ~|exp_Y;
  assign bias = {7'b0111111, !norm};
  assign ovrf = {{buffer >= {255 + bias}}|{over_X}|{over_Y}};
  assign udrf = {{buffer <= bias}|{under_X}|{under_Y}};


  assign exp_Z = exp_X + exp_Y - bias;

endmodule
```

Figura 2. Secciones de código agregadas usando Visual Studio Code.

Reportes de los escenarios (usando EDA Playground)

Solo se muestran ciertas secciones debido a la gran cantidad de datos obtenidos en el test y para el test aleatorio sí se muestra el encabezado de la simulación de donde se inicializan los componentes del ambiente, esto con el objetivo de evitar redundar en cosas que son siempre iguales para los demás tests.

Escenario 1: Envío de mensaje con instrucción del tipo aleatorio
Test 1. Aleatorización

```
UVM_INFO scoreboard.sv(167) @ 1790: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 11100100100000000000000001111111 Entrada y = 11100010100000000000000000001011
Bandera overflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 1810: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 01110111000000000000000000000110 Entrada y = 10101110100000000000000001001110
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 1830: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 00101110000000000000000001001010 Entrada y = 11101011100000000000000001010101
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 1850: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 01010100100000000000000000101001 Entrada y = 11001111000000000000000001100000
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 1870: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 00000010100000000000000001111001 Entrada y = 11000101100000000000000000000111
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 1890: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 11011011100000000000000000010100 Entrada y = 00111101000000000000000000011110
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 1910: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 01000011100000000000000001001101 Entrada y = 10010001100000000000000000111010
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 1930: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 00101100100000000000000001100101 Entrada y = 10000001000000000000000000111010
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 1950: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 00010111000000000000000001011000 Entrada y = 11001111000000000000000001011110
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 1970: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 01010111000000000000000001001011 Entrada y = 10001110100000000000000000011010
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 1990: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 10110011100000000000000001101011 Entrada y = 00111011100000000000000001010011
UVM_INFO sequence.sv(43) @ 1990: uvm_test_top.ambiente_instancia.agente_instancia.secuenciador_instancia@@seq.Secuencia_aleatorio [SEQ] Se generaron 100 items del tipo aleatorio
UVM_INFO /apps/vcsmx/vcs/Q-2020.03-SP1-1//etc/uvm-1.2/src/base/uvm_objection.svh(1276) @ 1990: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
UVM_INFO /apps/vcsmx/vcs/Q-2020.03-SP1-1//etc/uvm-1.2/src/base/uvm_report_server.svh(894) @ 1990: reporter [UVM/REPORT/SERVER]
--- UVM Report Summary ---

** Report counts by severity
UVM_INFO :  105
UVM_WARNING :    0
UVM_ERROR :    0
UVM_FATAL :    0
** Report counts by id
[RNTST]     1
[SCBD]   100
[SEQ]     1
[TEST_DONE]     1
[TOP]     1
[UVM/RELNOTES]     1
Resultado del DUT = 01111111100000000000000000000000 Resultado esperado = 01111111100000000000000000000000 Overflow = 1 Underflow = 0


Resultado del DUT = 11100110000000000000000001010101 Resultado esperado = 11100110000000000000000001010101 Overflow = 0 Underflow = 0

Resultado del DUT = 11011010000000000000000010011111 Resultado esperado = 11011010000000000000000010011111 Overflow = 0 Underflow = 0

Resultado del DUT = 11100100000000000000000010001001 Resultado esperado = 11100100000000000000000010001001 Overflow = 0 Underflow = 0

Resultado del DUT = 10001000100000000000000010000000 Resultado esperado = 10001000100000000000000010000000 Overflow = 0 Underflow = 0

Resultado del DUT = 11011001000000000000000000110011 Resultado esperado = 11011001000000000000000000110011 Overflow = 0 Underflow = 0

Resultado del DUT = 10010101100000000000000010000111 Resultado esperado = 10010101100000000000000010000111 Overflow = 0 Underflow = 0

Resultado del DUT = 10000000000000000000000000000000 Resultado esperado = 10000000000000000000000000000000 Overflow = 0 Underflow = 1


Resultado del DUT = 10100110100000000000000010110110 Resultado esperado = 10100110100000000000000010110110 Overflow = 0 Underflow = 0

Resultado del DUT = 10100110000000000000000001100101 Resultado esperado = 10100110000000000000000001100101 Overflow = 0 Underflow = 0

Resultado del DUT = 10101111100000000000000010111110 Resultado esperado = 10101111100000000000000010111110 Overflow = 0 Underflow = 0
```

Escenario 2: Envío de mensaje con instrucción para casos de esquina

Test 2.1. Alternancia de bits en los datos de entrada

```
UVM_INFO scoreboard.sv(167) @ 50: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 00000000000000000000000000000000 Entrada y = 11111111111111111111111111111111
UVM_INFO scoreboard.sv(167) @ 70: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 00000000000000000000000000000000 Entrada y = 01010101010101010101010101010101
Bandera underflow activada
UVM_INFO scoreboard.sv(167) @ 90: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 00000000000000000000000000000000 Entrada y = 10101010101010101010101010101010
Bandera underflow activada
UVM_INFO scoreboard.sv(167) @ 110: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 11111111111111111111111111111111 Entrada y = 00000000000000000000000000000000
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 130: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 11111111111111111111111111111111 Entrada y = 11111111111111111111111111111111
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 150: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 11111111111111111111111111111111 Entrada y = 01010101010101010101010101010101
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 170: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 11111111111111111111111111111111 Entrada y = 10101010101010101010101010101010
UVM_INFO scoreboard.sv(167) @ 190: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 01010101010101010101010101010101 Entrada y = 00000000000000000000000000000000
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 210: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 01010101010101010101010101010101 Entrada y = 11111111111111111111111111111111
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 230: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 01010101010101010101010101010101 Entrada y = 01010101010101010101010101010101
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 250: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 01010101010101010101010101010101 Entrada y = 10101010101010101010101010101010
UVM_INFO scoreboard.sv(167) @ 270: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 10101010101010101010101010101010 Entrada y = 00000000000000000000000000000000
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 290: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 10101010101010101010101010101010 Entrada y = 11111111111111111111111111111111
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 310: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 10101010101010101010101010101010 Entrada y = 01010101010101010101010101010101
UVM_INFO sequence.sv(93) @ 310: uvm_test_top.ambiente_instancia.agente_instancia.secuenciador_instancia@@seq.Secuencia_alternancia [SEQ] Se generaron 16 items del tipo alternancia

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 1
Resultado del DUT = 00000000000000000000000000000000 Resultado esperado = 00000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 10000000000000000000000000000000 Resultado esperado = 10000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 1

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0
Resultado del DUT = 00000000000000000000000000000000 Resultado esperado = 00000000000000000000000000000000 Overflow = 0 Underflow = 1


Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01101011001100011100011100011100 Resultado esperado = 01101011001100011100011100011100 Overflow = 0 Underflow = 0

Resultado del DUT = 11000000100011100011100011100011 Resultado esperado = 11000000100011100011100011100011 Overflow = 0 Underflow = 0
Resultado del DUT = 10000000000000000000000000000000 Resultado esperado = 10000000000000000000000000000000 Overflow = 0 Underflow = 1


Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 11000000100011100011100011100010 Resultado esperado = 11000000100011100011100011100010 Overflow = 0 Underflow = 0


Resultado del DUT = 00010101111000111000111000110111 Resultado esperado = 00010101111000111000111000110111 Overflow = 0 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0
```

Test 2.2. Overflow

```
UVM_INFO scoreboard.sv(167) @ 2150: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 01101010011000101000110000001111 Entrada y = 01010101000111101110110101010000
Bandera overflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 2170: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 11000101100101111010001001011010 Entrada y = 01111001110001010001110000000100
Bandera overflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 2190: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 01100000111001011101000101100100 Entrada y = 11011110100001101001110100001110
Bandera overflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 2210: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 01001110111010101011100111101101 Entrada y = 01110000101100001000000101010001
Bandera overflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 2230: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 11111010101110110101100101010110 Entrada y = 11000100110101011000001000101000
Bandera overflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 2250: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 01001000000010011111001101111001 Entrada y = 11110111011100100011101101000111
Bandera overflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 2270: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 11100011101110100101111000011101 Entrada y = 11011011111101011011010011001101
Bandera overflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 2290: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 11011011100000101011001100100010 Entrada y = 11100011101001111011000000000011
Bandera overflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 2310: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 01111110011000110111001000000000 Entrada y = 11000001011100001010110110001110
Bandera overflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 2330: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 01100010100110010110101010000100 Entrada y = 01011100110111100101010010001011
UVM_INFO sequence.sv(131) @ 2330: uvm_test_top.ambiente_instancia.agente_instancia.secuenciador_instancia@@seq.Secuencia_overflow [SEQ] Se generaron 100 items del tipo overflow
Resultado del DUT = 01111111100000000000000000000000 Resultado esperado = 01111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 11111111100000000000000000000000 Resultado esperado = 11111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 11111111100000000000000000000000 Resultado esperado = 11111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111100000000000000000000000 Resultado esperado = 01111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111100000000000000000000000 Resultado esperado = 01111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 11111111100000000000000000000000 Resultado esperado = 11111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111100000000000000000000000 Resultado esperado = 01111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111100000000000000000000000 Resultado esperado = 01111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 11111111100000000000000000000000 Resultado esperado = 11111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111100000000000000000000000 Resultado esperado = 01111111100000000000000000000000 Overflow = 1 Underflow = 0
```

Test 2.3: Underflow

```
UVM_INFO scoreboard.sv(167) @ 4130: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 10000000011100111011010100000011 Entrada y = 11001111001110111000011011001011
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 4150: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 00000000100100100010110000010110 Entrada y = 11001011101111100010000111010110
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 4170: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 11010110001111010010101111111101 Entrada y = 00000000001110001101010001000101
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 4190: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 00011111100000011001000000101110 Entrada y = 00000000000100001110101111111110
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 4210: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 11110100001101101111101111010110 Entrada y = 00000000100101111011011011100110
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 4230: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 00010100100110011101110011111101 Entrada y = 10000000111111101011111100111111
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 4250: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 10000000101101000110000010000100 Entrada y = 11110100010011101000101011000000
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 4270: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 00000000001100001011000101000100 Entrada y = 11110110010111000110010111000001
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 4290: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 10000000110010101111011000010000 Entrada y = 11001011100011111010000011110000
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 4310: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 11011101010110101111110000000100 Entrada y = 00000000111111000111000110010100
Bandera underflow activada
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 4330: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 10100010000110001001000011001011 Entrada y = 10000000000000000101011111100001
UVM_INFO sequence.sv(169) @ 4330: uvm_test_top.ambiente_instancia.agente_instancia.secuenciador_instancia@@seq.Secuencia_underflow [SEQ] Se generaron 100 items del tipo underflow
```

```
Resultado del DUT = 00000000000000000000000000000000 Resultado esperado = 00000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 10000000000000000000000000000000 Resultado esperado = 10000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 10000000000000000000000000000000 Resultado esperado = 10000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 00000000000000000000000000000000 Resultado esperado = 00000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 10000000000000000000000000000000 Resultado esperado = 10000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 10000000000000000000000000000000 Resultado esperado = 10000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 00000000000000000000000000000000 Resultado esperado = 00000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 10000000000000000000000000000000 Resultado esperado = 10000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 00000000000000000000000000000000 Resultado esperado = 00000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 10000000000000000000000000000000 Resultado esperado = 10000000000000000000000000000000 Overflow = 0 Underflow = 1

Resultado del DUT = 00000000000000000000000000000000 Resultado esperado = 00000000000000000000000000000000 Overflow = 0 Underflow = 1
```

Test 2.4: No es un número (NaN)

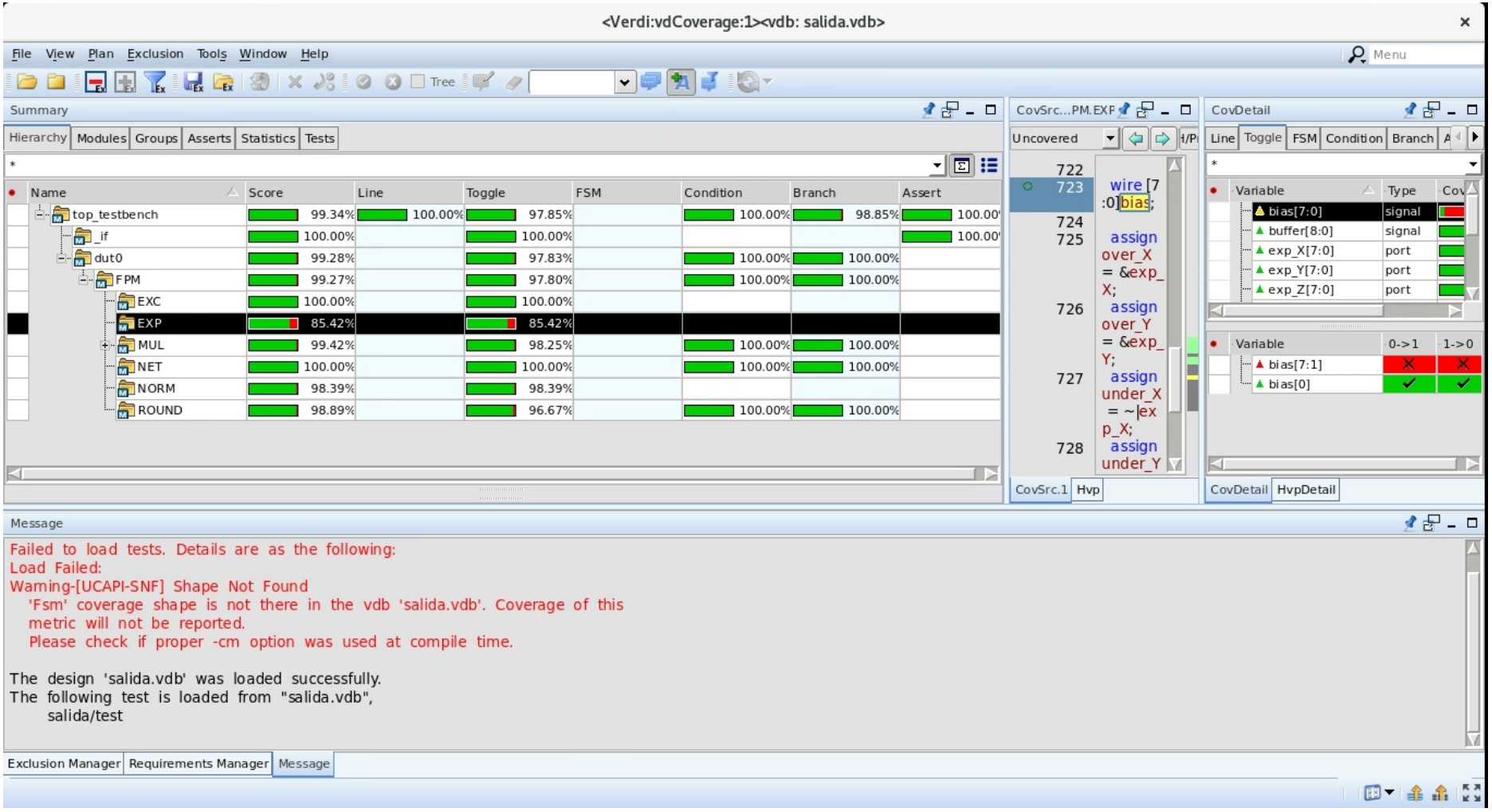
```
UVM_INFO scoreboard.sv(167) @ 6030: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 01111111100100101111101000011000 Entrada y = 10100100111110101011100101100001
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6050: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 11011000111011111101100001110010 Entrada y = 01111111101011011010010110011110
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6070: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 01111111101010011101000000011111 Entrada y = 11001111100001111110001010101001
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6090: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 11111111101011010000111001100000 Entrada y = 11010001001000101000111100100101
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6110: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 11111111011110010100001001001010 Entrada y = 11000110100001100000110110111110
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6130: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 01111111101011111101011000001011 Entrada y = 00001110010110101001110101111011
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6150: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 10111010011101000000011001100011 Entrada y = 01111111111111101110001001111110
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6170: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 11100001000010011111101011001100 Entrada y = 01111111111100110000101110010010
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6190: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 01011010000110000101100110110011 Entrada y = 01111111100101000001100111101011
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6210: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 10010101000100001011110000100010 Entrada y = 01111111101010010100001110100000
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6230: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 10100100010000111110011000101000 Entrada y = 01111111110010001111100101011011
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6250: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 10101011000100001011010010111001 Entrada y = 11111111110011000100010011000010
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6270: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 100 Entrada x = 11110110111011000101100110100101 Entrada y = 11111111101110100001001111110011
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6290: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 01111000110100001011111011000001 Entrada y = 11111111110110000100100101100110
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6310: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 010 Entrada x = 11111111100100001000000101111110 Entrada y = 010101011111111110011000001110111
Resultado redondeado
UVM_INFO scoreboard.sv(167) @ 6330: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 10101010001110110000110111001100 Entrada y = 11111111100111010100001110100111
UVM_INFO sequence.sv(207) @ 6330: uvm_test_top.ambiente_instancia.agente_instancia.secuenciador_instancia@@seq.Secuencia_NaN [SEQ] Se generaron 100 items del tipo NaN
```

```
Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111110000000000000000000000 Resultado esperado = 01111111110000000000000000000000 Overflow = 1 Underflow = 0
```

Test 2.4: Infinito (inf)

```
UVM_INFO scoreboard.sv(167) @ 8210: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 11111111100000000000000000000000 Entrada y = 00011110110110011010010101000000
Bandera overflow activada
UVM_INFO scoreboard.sv(167) @ 8230: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 01000111101110010111001100011100 Entrada y = 01111111100000000000000000000000
Bandera overflow activada
UVM_INFO scoreboard.sv(167) @ 8250: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 01111111100000000000000000000000 Entrada y = 00111110111011110110000101100011
Bandera overflow activada
UVM_INFO scoreboard.sv(167) @ 8270: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 11111111100000000000000000000000 Entrada y = 01000100000001100000011011101011
Bandera overflow activada
UVM_INFO scoreboard.sv(167) @ 8290: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 011 Entrada x = 00001100011010100001001000000001 Entrada y = 01111111100000000000000000000000
Bandera overflow activada
UVM_INFO scoreboard.sv(167) @ 8310: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 11111111100000000000000000000000 Entrada y = 01111010110110110100111000010001
Bandera overflow activada
UVM_INFO scoreboard.sv(167) @ 8330: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 000 Entrada x = 11100010111010000101110100011011 Entrada y = 01111111100000000000000000000000
Bandera overflow activada
UVM_INFO scoreboard.sv(167) @ 8350: uvm_test_top.ambiente_instancia.scoreboard_instancia [SCBD] Mode = 001 Entrada x = 01111111100000000000000000000000 Entrada y = 10100010001111100110100000000100
UVM_INFO sequence.sv(246) @ 8350: uvm_test_top.ambiente_instancia.agente_instancia.secuenciador_instancia@@seq.Secuencia_inf [SEQ] Se generaron 100 items del tipo inf
UVM_INFO /apps/vcsmx/vcs/Q-2020.03-SP1-1//etc/uvm-1.2/src/base/uvm_objection.svh(1276) @ 8350: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
UVM_INFO /apps/vcsmx/vcs/Q-2020.03-SP1-1//etc/uvm-1.2/src/base/uvm_report_server.svh(894) @ 8350: reporter [UVM/REPORT/SERVER]
--- UVM Report Summary ---

** Report counts by severity
UVM_INFO :  427
UVM_WARNING :    0
UVM_ERROR :    0
UVM_FATAL :    0
** Report counts by id
[RNTST]     1
[SCBD]   418
[SEQ]     5
[TEST_DONE]     1
[TOP]     1
[UVM/RELNOTES]     1

Resultado del DUT = 11111111100000000000000000000000 Resultado esperado = 11111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111100000000000000000000000 Resultado esperado = 01111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111100000000000000000000000 Resultado esperado = 01111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 11111111100000000000000000000000 Resultado esperado = 11111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 01111111100000000000000000000000 Resultado esperado = 01111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 11111111100000000000000000000000 Resultado esperado = 11111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 11111111100000000000000000000000 Resultado esperado = 11111111100000000000000000000000 Overflow = 1 Underflow = 0

Resultado del DUT = 11111111100000000000000000000000 Resultado esperado = 11111111100000000000000000000000 Overflow = 1 Underflow = 0
```

Cobertura de la prueba

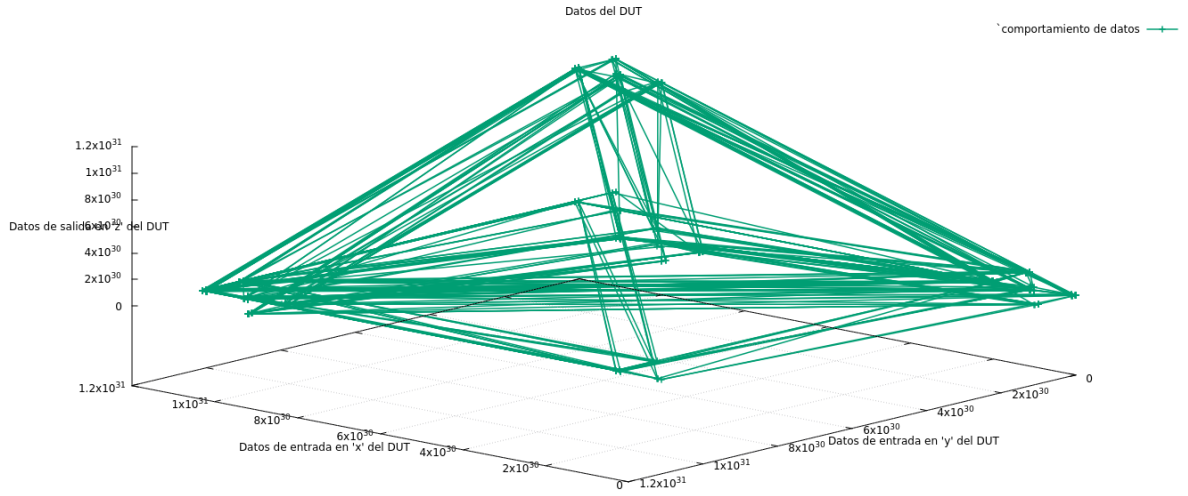


En la imagen anterior se muestra la cobertura de la prueba en la que se cumple la condición de al menos un 95% de cobertura para el toggle y para los casos.

Gráficos de reportes con GNUplot

- **Gráfica reporte de las entradas en “x” y “y” con respecto a la salida “z”**

En el siguiente apartado se grafican los datos obtenidos en el reporte obtenido del test, por lo que la siguiente figura muestra la gráfica de los datos de entrada en “y” del DUT, los datos de entrada en “x” del DUT y los datos de salida en “z” del DUT, mediante una gráfica de datos de las entradas en “x” y “y” con respecto a la salida “z”.



Conclusiones

- Se cumplen las condiciones de aprobación en la verificación para todos los escenarios expuestos en el test plan.
- En el momento en que el DUT sea implementado en la vida real, se debe tener mucho cuidado sobre la forma en que se usan las profundidades de las entradas de datos para evitar la pérdida de datos por overflow o por underflow.
- El fin mismo de la verificación de este DUT fue el poder lograr reparar el error que tenía el dispositivo, que al final fue satisfactoriamente solucionado
- Las gráficas obtenidas mediante el software GNUplot permiten de una manera visual conocer el comportamiento del DUT ante diferentes escenarios.
- Los datos de cobertura de la prueba facilitan en gran medida poder contemplar escenarios de esquina que no son pensados de primera mano a la hora de montar el test plan.