

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica

Verificación Funcional de Circuitos Integrados

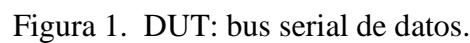
Proyecto 1: Diseño e implementación de pruebas para un bus serial de
datos

Estudiante: Marcelo Cardona Aguilar

Carné: 2013032712

II semestre, 2021

El objetivo de este proyecto es el diseño e implementación de una prueba de verificación funcional mediante el uso de System Verilog para un bus de datos serial. El diagrama de dicho sistema puede observarse en la figura 1.



- Testbench.sv
- Test.sv
- Ambiente.sv
- Agente.sv
- Driver.sv
- Checker.sv
- Monitor.sv
- Scoreboard.sv

En la figura 2 se puede observar el diagrama que describe la relación entre cada uno de estos módulos.

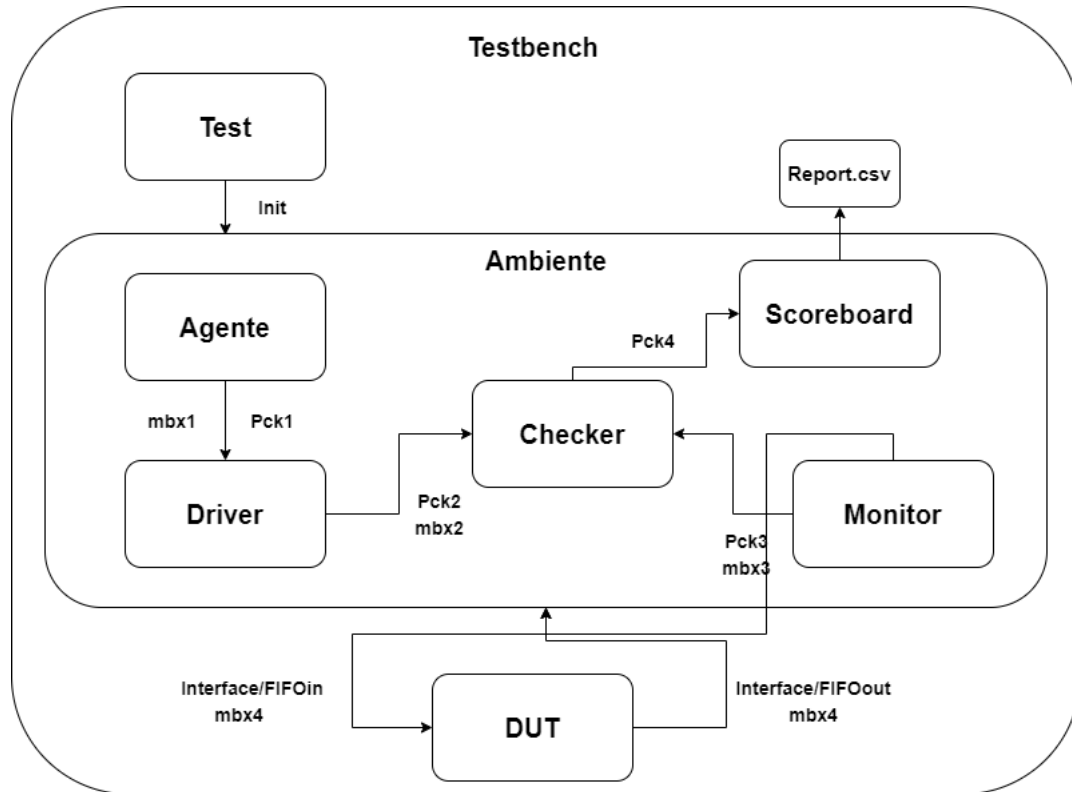


Figura 2. Diagrama de capas de la prueba diseñada.

Además, se cuenta con el módulo de descripción de hardware (design.sv) que se encarga de estructurar el DUT y otros tres módulos para generar la interfaz que debe tener el DUT para interactuar con la prueba

- Design.sv (código del DUT proporcionado)
- Transactions.sv
- Interface.sv
- Fifoem.sv

Definición de paquetes y mailboxes:

Init: Inicializa el ambiente y le pasa todos los parámetros necesarios a la hora de instanciarlo. Parámetros: tamaño del paquete, cantidad de drivers, tamaño de la cola o profundidad, broadcast.

Pck1: transmite del agente al driver el destino [8 bits], datos [tamaño aleatorizable que depende del tamaño del paquete], origen [tamaño aleatorizable que depende de la cantidad de drivers o dispositivos], retardo aleatorio, retardo máximo, dirección máxima a la que se puede enviar un paquete y dirección mínima.

Pck2: transmite del driver al checker el destino del paquete, los datos del paquete, el origen del paquete, el indicador de broadcast y el tiempo de envío.

Pck3: transmite del monitor al checker los datos del paquete, el receptor del paquete (para compararlo con el destino) y el tiempo de recepción del paquete.

Pck3: transmite del monitor al checker los datos del paquete, el receptor del paquete (para compararlo con el destino) y el tiempo de recepción del paquete.

Mbx1: agent_drvr_mbx es el mailbox que se encarga de conectar el dispositivo agente y el dispositivo driver.

Mbx2: drvr_chkr_mbx es el mailbox que se encarga de conectar el dispositivo driver y el dispositivo checker.

Mbx3: mntr_chkr_mbx es el mailbox que se encarga de conectar el dispositivo monitor y el dispositivo checker.

Mbx4: fifo_mbx es un mailbox auxiliar utilizado para iterar dentro de la cola de la fifo emulada.

Variables parametrizables: tamaño del paquete, cantidad de drivers o dispositivos, profundidad de la cola de la fifo.

Variables aleatorizables: las variables que se logran aleatorizar son el # de dispositivo, el tiempo de retardo y los datos del mensaje que contiene el paquete.

Cabe destacar también, que el módulo de Scoreboard se encarga de tomar los datos del checker y los introduce en un archivo .csv que muestra los resultados de las transacciones. Estos resultados incluyen el dispositivo fuente y destino, el dato enviado (por el driver), el dato recibido (por el monitor), el dispositivo receptor del dato, el tiempo en el que se envía y se recibe el paquete y el tiempo de retardo o delay.