**Instituto Tecnológico de Costa Rica**

**Escuela de Ingeniería Electrónica**

**Verificación Funcional de Circuitos Integrados EL-5811**

**Proyecto 1**

**Prof. Ronny García Ramírez**

**Estudiantes: Alonso Vega Badilla**

**Agustín Delgado Sancho**

El testbench está compuesto por los siguientes archivos:

* *Transactions.sv*: Se definen las transacciones y los mailboxes para la comunicación entre los módulos testbench. Además, se definen las interfaces para la comunicación con el DUT.
* *testbench.sv*: Bloque principal en el que se instancia el DUT, el test y se incluyen todos los otros archivos. Aquí se conecta el DUT al driver y al monitor a través de una interfaz.
* *test.sv*: Genera las secuencias de transacciones que conforman la prueba, las cuales se envían al agente. Este módulo permite generar ya sea casos de esquina o secuencias aleatorias.
* *ambiente.sv*: En el ambiente se instancian los bloques de agente, scoreboard, checker, driver y monitor y se unen tanto los mailbox como las interfaces.
* *agent.sv*: Cumple las funciones del generador y del agente. Se encarga de generar las instrucciones para el driver. Genera tres tipos de paquetes: trans, broadcast y reset. Los paquetes trans consisten en el envío de un paquete desde un dispositivo del bus hacia otro dispositivo específico; los paquetes de broadcast consisten en el envío de un dato desde un dispositivo hasta todos los otros dispositivos del bus mediante el código de broadcast; y los paquetes reset consisten en un reinicio del bus, ante el cual se espera que todas sus salidas pasen a ser cero.
* *scoreboard.sv*: Almacena los detalles de cada transacción realizada, incluyendo fuente del mensaje, destino, tiempos de ejecución, retardos, datos y si la transacción se completó correctamente.
* *checker.sv*: Este bloque recibe información de cada transacción generada por el agente y espera a la transacción recibida en el monitor. Luego compara ambos datos para identificar si la transacción se realizó exitosamente. El resultado de cada transacción es enviado al scoreboard para ser procesado.
* *Driver.sv*: Recibe tres tipos de transacciones del agente: trans, broadcast y reset. Envía las señales correspondientes al DUT según cada transacción. En este módulo se crean procesos independientes para manejar cada terminal del bus en paralelo. Cada proceso emula la FIFO de la terminal correspondiente.
* *Monitor.sv*: Lee las señales de salida del DUT y según estas identifica el tipo de transacción que se está ejecutando. Al detectar que se ejecutó una transacción, se envía su información al checker para ser verificada.
* *Library.sv*: Contiene la definición del dispositivo bajo prueba.

*output.csv*: Archivo generado por el scoreboard que almacena la información de todas las transacciones enviadas al DUT durante la prueba.