Evaluations of CAN

A CAN bus that stands for Controller Area Network is tested for its functionality. CAN bus was designed by the BOSCH industry to allow any ECU(Engine control unit) in the car to be able to communicate with each other. It is a serial communication bus protocol. Engine Control Units are just like nodes of electrical circuits with branches having various components of CAR environment like and audio system, an Airbags unit, Electronic gearbox control or even the temperature sensor within the car. A modern car can have up to 70 plus ECUs. This is where CAN bus comes into picture allowing no complex wiring for communication between any ECU unit.

Introduction

A CAN bus that stands for Controller Area Network is tested for its functionality. CAN bus was designed by the BOSCH industry to allow any ECU(Engine control unit) in the car to be able to communicate with each other. It is a serial communication bus protocol. Engine Control Units are just like nodes of electrical circuits with branches having various components of CAR environment like and audio system, an Airbags unit, Electronic gearbox control or even the temperature sensor within the car. A modern car can have up to 70 plus ECUs. This is where CAN bus comes into picture allowing no complex wiring for communication between any ECU unit.

Methodology

Sequence Item

As we have seen before that sequence item is nothing but the uvm object where we declare all the fields. For CAN protocol verification, the fields that were necessary as per the design specification are listed in the sequence item class and is shown as below.

Each field has been explained by using the comment line. For example, [3:0] datalen has a comment line saying that 0-8 are valid entries. Similarly, [3:0] xmitdata says that it is data in and big-endian byte order is used. Since we have included cantdef package, it is necessary that we see what does the package contains in it.

Sequence

As discussed earlier each UVM component shall have a constructor function. A constructor function is the one which takes string name as input which is usually the class name in which the constructor is being used. Constructors are functions. Now let us use the body task function that sequence uses to generate randomized sequences. As the design needs various parameters to be checked, therefore we need to have a set of randomizations that need to be performed in order to generate the appropriate stimulus.

The DUT and interface for the CAN is also instantiated here. Interfaces are considered as bundle of signals that are used for connectivity to the DUT and Test bench. And it is used for simplicity and decreases the effort for hardcoding of signals, in case of there are signals additions or removal of signals

Analysis and Results

The Result shows the output of the verification performed on four different DUTs. The simulation is performed using Synopsys VCS simulator and ran on the Linux-based machine. Below figure shows the inputs and outputs of the CAN design and performed verification using universal verification methodology.

Scoreboard

Scoreboard is a verification component that contains checkers and verifies the overall DUT’s functionality at transactional level. A reference model can be used that depicts the DUT and Reference model’s output can be compared with DUT’s output and it can be part of checker. Reference model is the one that calculates the expected value in form of golden vectors that is compared with the actual values from DUT.