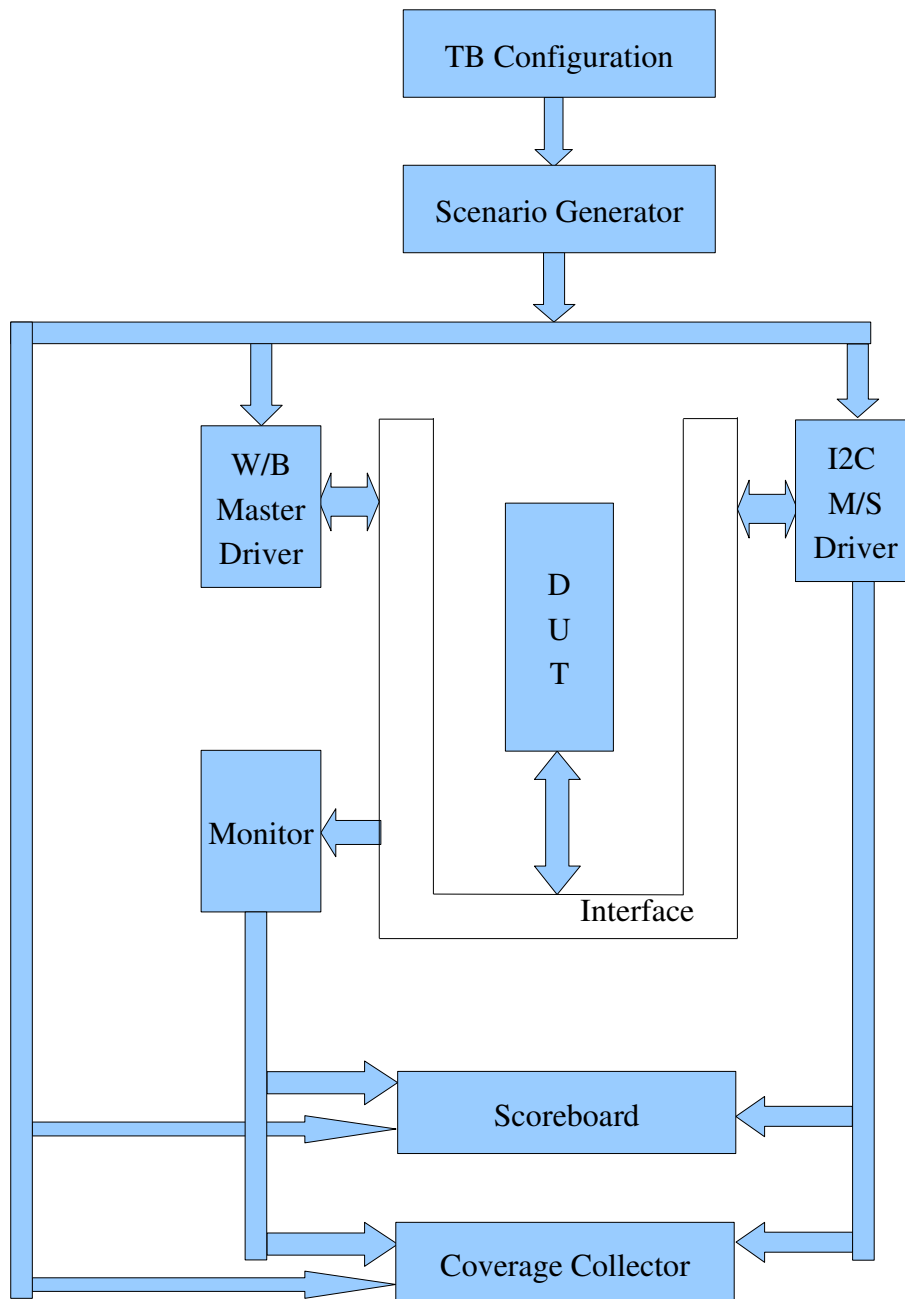


I2C Master/Slave Core Verification Plan

The block diagram shows the basic VMM methodology which will be followed to verify this I2C core.



Description:

The verification of I2C can be divided into different test-case category. Following are the categories which are gonna do exhaustive verification of I2C Core:

- Data Transmission Test case.
- Register's Writing and Reading.
- Reset Operation.

The Selection of which category of test-case to run can be done either at command line or by using configuration class. For command line selection, \$value\$plusargs system task will be used to pass value from command line.

The TB Configuration is used to configure the test case. This configuration block is commonly used in VMM to configure the test cases. In this verification environment the configuration class will be having only one field which will control no. of transaction to be done

Scenario Generator is used to generate different scenarios depending on the test case configuration's fields selection. It uses different packets to generate and pass packet to both master as well as slave driver.

Scenario Generator first randomizes a packet (scenario packet) to generate random mode (type) of test-case to be run. Scenario Packet class has following random fields:

- Master/Slave selection
- Transmit/Receive Operation
- Reset Check
- Register check.

After randomization of scenario packet, Scenario Generator randomizes stimulus packet and data packet. Stimulus packet contains following randomized fields:

- Interrupt enable/disable bit
- No. of bytes to transmit/receive
- Register data
- Register address
- Slave address

In data packet only dynamic array of data byte is used. The size of this array will be assigned after randomization of stimulus packet and then this array will be randomized.

Scenario Generator will send stimulus packet to both drivers W/B Driver (vmm_i2c_driver.v) and I2C M/S Driver or Responder(i2c_slave_driver.v). W/B driver then writer internal register for DUT configuration and I2C driver will respond accordingly.

3. Functionalities to be Covered/Checked:

Following are the functionalities which will be checked and their respective cover groups/points are mention in brackets next to them:

Data transfer (CG1)

- Both Possible mode of DUT which are given below: (m_s)
 - Master
 - Slave
- Both side of transaction which are : (tx_rx)
 - Transmitter
 - Receiver
- No. of bytes to be transfer.
This can be divided into 3 categories: (data_in)
 - low
 - mid
 - high
- Cross Coverage between master_slave , transmit /receive and data byte to make sure that all possible mode are checked for all category of data byte size. (data_mode_cross).
- Slave Address match. Again this address can be divided into 2-3 categories. (slave_addr)
 - low
 - mid
 - high

Protocol Validation (CG3)

- Generation and detection of Start signal during data transfer. (start_bit).
- Generation and detection of Stop signal during data transfer. (stopt_bit)
- Generation and detection of Slave Address Acknowledgment. (slave_ack).
- Generation and detection of Data Byte Acknowledgment. (data_ack)
- Generation and clearing the interrupt. (intr_ack)
- Cross Coverage between Start signal and all possible (four) modes of operation. (start_mode_cross)
- Cross Coverage between Stop signal and all possible (four) modes of operation. (stop_mode_cross)
- Cross Coverage for slave address acknowledgment and all possible modes of operation. (sack_mode_cross)
- Cross Coverage between data acknowledgment and all possible modes of operation. (dack_mode_cross)
- Cross Coverage between interrupt generation and all possible modes of operation. (iack_mode_cross)

Registers and Reset Check (CG2)

- Cover point with bins for all internal register's addresses.(register_addr)
- Write into a register or read from a register. (write_read).
- Cross Coverage between write/read and register address to insure all registers have been written and read. (cross_reg_addr_wr)
- Write the same register twice and see if newer value is updated in register for the transition coverage. (txn)
- Apply reset test-case randomly. Any time reset test can be selected. (reset_test)