SystemVerilog for Verification

Final Project

Transaction

-

Can I run predefined sequences? (e.g. reset sequence, random write sequence, random read/write bursts, a directed test). Can I debug easily if my test fails? Do I need one or multiple transactions for bursts?

-

Yes I can run pre-defined sequences. These sequences are defined in the pre_randomize() function in the extended transaction class.
Yes I can debug easily if my test fails. I have added a print_trans() function in my transaction class that prints the values of all signals in the interface. This function can be used in generator, driver, monitor and scoreboard by creating an object for the transaction class inside these classes.

Generator

-

How to control how many transactions get generated? Sometimes random transactions are not needed. How do I generate non-random transactions when required?

-

There is a variable repeat_count in the generator that controls the number of transactions being generated.
To generate non random transactions, we turn the random mode off in the pre_randomized() function using the rand_mode() function. Their random mode is turned off and are assigned some manual value.

Driver

-

Are the interface signals driven according to the spec? Does the transaction have proper address/data Phases? Do I need to sample inputs to decide whether to drive outputs or not on the next clocking event?

-

Yes the interface signals are driven according to the specifications. Yes, the transaction have proper address and data phases.
Yes, we need to sample inputs to decide whether to drive output or not on the next clocking event.

Monitor
-
Are the interface signals sampled according to the spec? Does the transaction have proper address/data Phases?
-
Yes, the interface signals are sampled according to the specifications and each transaction have proper address and data phases.

Scoreboard
-
Does the scoreboard implement proper endianness? How to change endianness if required? How to not compare reset values and to compare only those memory locations which have already been written? Should scoreboard memory be static or dynamic?
-
In the scoreboard, only little endianness is taken into consideration. The endianness cannot be changed as it is not supported in my scoreboard.
It can be done if we create an indication for that address on which data is written. And when we read from it, we first check whether the indication is high or not. If it is high it means it is compared correctly and stored data will be displayed at output.
The memory should be according to the specifications of the memory of the DUT.

Verification Plan

Test Id	**Test Name**	**Test Status**	**Test Description**	**Stimulus Generation Procedure**	**Test Passes When**	**Test Fails When**	**Checking Procedure**	**Comments**
1	test_rand_wr_rd_fixed_addr	PASS	Checks for random write and read on fixed address	Randomization was turned off for these variables and they were assigned values as HADDR = 4, with HREADY = 1, HBURST = 0, HSEL = 1, HTRANS = 2, HSIZE = 2	Test passess on both HWRITE = 0 and HWRITE = 1. Which means that read and write on a certain address is correctly performed	Never	checks if HWRITE is high then it must be written in memory and when HWRITE is low then it must be displayed on HRDATA	HWRITE, along with high HREADY value ensures that on high HWRITE data coming from master is stored in local mem and on low HWRITE it is transferred back to master

2	test_wr_not_ready_in_addr_phase	PASS	Checks for write on a certain address when ready signal is turned off in the address phase	Randomization was turned off for these variables and they were assigned values as HADDR = 4, with HREADY = 0 or 1, HBURST = 0, HSEL = 1, HTRANS = 2, HSIZE = 2, HWRITE = 1	All the transactions are passed.	Never	checks; if the HREADY signal is off in the address phase of write, nothing should be written in memory for write.	It ensures that the write operation occurs only when HREADY signal is high.
3	test_rd_not_ready_in_addr_phase	FAIL	Checks for write and then read on a certain address when ready signal is turned off in the address phase	Randomization was turned off for these variables and they were assigned values as HADDR = 4, with HREADY = 0 or 1, HBURST = 0, HSEL = 1, HTRANS = 2, HSIZE = 2, HWRITE = 1 and then 0	Transactions during ready = 1, pass but when ready = 0, they fail	Read case with HREADY = 0	checks; if the HREADY signal is off in the address phase of read, nothing should be displayed on HRDATA for read.	It ensures that the read operation occurs only when HREADY signal is high.
4	test_wr_not_ready_in_data_phase	PASS	Checks for write on a certain address when ready signal is turned off in the data phase	Randomization was turned off for these variables and they were assigned values as HADDR = 4, with HREADY = 0 or 1, HBURST = 0, HSEL = 1, HTRANS = 2, HSIZE = 2, HWRITE = 1	All the transactions are passed.	Never	checks; if the HREADY signal is off in the data phase of write, nothing should be written in memory for write	It ensures that the write operation occurs only when HREADY signal is high.
5	test_rd_not_ready_in_data_phase	FAIL	Checks for read on a certain address when ready signal is turned off in the data phase	Randomization was turned off for these variables and they were assigned values as HADDR = 4, with HREADY = 0 or 1, HBURST = 0, HSEL = 1, HTRANS = 2, HSIZE = 2, HWRITE = 1 and then 0	Transactions during ready = 1, pass but when ready = 0, they fail	Read case with HREADY = 0	checks; if the HREADY signal is off in the data phase of read, nothing should be displayed on HRDATA for read.	It ensures that the read operation occurs only when HREADY signal is high.
6	test_sel	PASS	Checks for the output when selection signal is turned off	Randomization was turned off for these variables and they were assigned values as HSEL = 0, with HADDR = multiple of 2, HREADY = 1, HBURST = 0, HTRANS =	when HSEL = 0, nothing is drivin. Thus, the test passes with any set of random	Never	Checks when the selection signal is turned off, nothing is driven.	It ensures that the data transfer takes place only when the slave is set active

				2, HSIZE = 2	values with HSEL = 0			
7	test_not_ready	PASS	Checks for output when ready is turned off throughout the test	Randomization was turned off for these variables and they were assigned values as HREADY = 0, with HADDR = multiple of 2, HSEL = 1, HBURST = 0, HTRANS = 2, HSIZE = 2 and HWRITE = 0 or 1	when HREADY = 0, slave is not ready for any read or write. Thus, the test passes with any set of random values with HREADY = 0	Never	Checks when the ready signal is turned off, nothing is read nor written.	It ensures that the data transfer occurs only when HREADY signal is high.
8	test_nonseq_wr_rd	PASS	Checks for Non Sequential transfer for read and write.	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HSEL = 1, HBURST = 0, HTRANS = 2, HSIZE = 2, HREADY = 1 and HWRITE = 0 or 1	test passes in all cases	Never	Check whether is Non Sequential transfer of data is working correctly or not by writing Non sequentially at a certain address and then reading from it	It ensures the correct Non Sequential transfer of data by slave
9	test_seq_wr_rd	PASS	Checks for Sequential transfer for read and write.	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HSEL = 1, HBURST = 2, HTRANS = 3, HSIZE = 2, HREADY = 1 and HWRITE = 0 or 1	test passes in both cases of read and write	Never	Check whether is Non Sequential transfer of data is working correctly or not by writing Sequentially at a certain addresses and then reading from them	It ensures the correct Sequential transfer of data by slave
10	test_idle_wr	PASS	Checks for Idle transfer for write.	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HSEL = 1, HBURST = 0, HTRANS = 0 or 2, HSIZE =	test passes in all cases	Never	Checks whether there is any write operation when slave is in idle state of transfer	It ensures that there should be no data tranfer when slave is idle.

				2, HREADY = 0 and HWRITE = 1				
11	test_idle_rd	FAIL	Checks for Idle transfer for read	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HSEL = 1, HBURST = 0, HTRANS = 0 or 2, HSIZE = 2, HREADY = 0 and HWRITE = 1 and then 0	test passess when HTRANS = 2	test fails for read when HTRANS = 0	Checks whether there is any read operation when slave is in idle state of transfer	It ensures that there should be no data tranfer when slave is idle.
12	test_busy_wr	PASS	Checks for Busy transfer for write.	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HSEL = 1, HBURST = 0, HTRANS = 1 or 2, HSIZE = 2, HREADY = 0 or 1 and HWRITE = 1	test passes in all cases	Never	Checks whether there is any write operation when slave is in busy state of transfer	It ensures that there should be no data tranfer when slave is busy.
13	test_busy_rd	FAIL	Checks for Busy transfer for read	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HSEL = 1, HBURST = 0, HTRANS = 1 or 2, HSIZE = 2, HREADY = 0 or 1 and HWRITE = 0 or 1	test passess when HTRANS = 2	test fails for read when HTRANS = 1	Checks whether there is any read operation when slave is in busy state of transfer	It ensures that there should be no data tranfer when slave is busy.
14	test_WORD_wr_rd	PASS	Checks for WORD write and read	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HSEL = 1, HBURST = 0, HTRANS = 2, HSIZE = 2, HREADY = 1 and HWRITE = 0 or 1	test passes in all cases	Never	Check whether a WORD is written in memory by first writing and then reading it	It ensures that WORD is written in memory if the given size is word
15	test_HALFWORD_wr_rd	PASS	Checks for HALFWORD write and read	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HSEL = 1, HBURST = 0, HTRANS = 2, HSIZE = 1, HREADY = 1 and HWRITE = 0 or 1	test passes in all cases	Never	Check whether a HALFWORD is written in memory by first writing and then reading it	It ensures that HALFWORD is written in memory if the given size is word

16	test_BYTE_wr_rd	PASS	Checks for BYTE write and read	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HSEL = 1, HBURST = 0, HTRANS = 2, HSIZE = 0, HREADY = 1 and HWRITE = 0 or 1	test passes in all cases	Never	Check whether a BYTE is written in memory by first writing and then reading it	It ensures that BYTE is written in memory if the given size is word
17	test_RESETn	FAIL	Checks for values whenever the reset is low at any time	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HSEL = 1, HBURST = 0, HTRANS = 2, HSIZE = 2, HREADY = 1 and HWRITE = 1. From interface, HERESTn was set 0 at start, then set to 1 after #10 then again set to 0 after #50 and then to 1 after #20	test passes when HRESETn is set to 0 at start	test fails whenever HRESETn is set to 0 in between simulation, it has no affect on any of the signals	Checks whether all the signals are reset when it is low	It ensure that all signals should be turn to zero whenever there is reset in code
18	test_Idle_to_NonSeq_wr_rd	FAIL	Waited Transfer, from IDLE to NON SEQ, checks for wait transfers on IDLE state	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HDATA = manual, HSEL = 1, HBURST = 0 or 3, HTRANS = 0 or 2, HSIZE = 2, HREADY = 0 or 1 and HWRITE = 0 or 1.	test passes in case of HWRITE = 1, HREADY = 0 or 1 and HWRITE = 0, HREADY = 1, HTRANS = 2	tests fails when HWRITE = 0, HREADY = 0, HTRANS = 0 or 2	checks the whether waited transfers are inserted on idle state or not	It ensures that waited transfers should be inserted on idle state of slave
19	test_Busy_to_Seq_wr_rd	FAIL	Waited Transfer, from Busy to SEQ, checks for wait transfers on BUSY state	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HDATA = manual, HSEL = 1, HBURST = 0 or 3, HTRANS = 1 or 2 or 3, HSIZE = 2, HREADY = 0 or 1 and HWRITE = 0 or 1.	test passes in case of HWRITE = 1, HREADY = 0 or 1 and HWRITE = 0, HREADY = 1, HTRANS = 2 or 3	tests fails when HWRITE = 0, HREADY = 0, HTRANS = 1 or 2 or 3	checks the whether waited transfers are inserted on busy state or not	It ensures that waited transfers should be inserted on busy state of slave

20	test_addr_change_in_idle	FAIL	Waited Transfer, from IDLE to NON SEQ, checks for wait transfers on IDLE state. During IDLE state, address is changed	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HDATA = manual, HSEL = 1, HBURST = 0 or 3, HTRANS = 0 or 2, HSIZE = 2, HREADY = 0 or 1 and HWRITE = 0 or 1.	test passes in case of HWRITE = 1, HREADY = 0 or 1 and HWRITE = 0, HREADY = 1, HTRANS = 2	tests fails when HWRITE = 0, HREADY = 0, HTRANS = 0 or 2	checks the whether waited transfers are inserted on idle state or not. and if the address changes in the idle state, the data is written or read at the address which was before the idle state or not	It ensures that waited transfers should be inserted on idle state of slave and if the address changes during that time, then when the idle state finishes, the data should be written or read from the address before the idle state
21	test_error_resp	FAIL	checks for error response	Randomization was turned off for these variables and they were assigned values as with HADDR = manual, HDATA = manual, HSEL = 1, HBURST = 0, HTRANS = 2, HSIZE = 2, HREADY = 0 or 1 and HWRITE = 0 or 1.	test passes in socreboard	test fails if we see waveform, no error resp is indicated by HRESP	checks whether the error response is generated or not	Ensures that slave generates the error response in HRESP when there is some error condition