Verification Plan for layered testbench

#	Section	Description
	Simulation	
1	i2cmb_Registers	
1.1	Register Covergroup	Register Covergroup
1.2	Valid Register Address	Coverpoint with bins for valid address
1.3	Register Default Values	Cross to verify Default values of registers: detects enable, start, stop, set_bus, nack, write
1.4	Register field address	test for register aliasing
1.5	Register field Aliasing	Check for register aliasing
1.6	Register Access Permission	Test write to RO register (FSMR)
1.7	Bus Busy Status	Coverpoint to check status of bus
1.8	Bus Capture Status	Coverpoint to check whether bus is captured or not
1.9	Interrupt status	Coverpoint to check for intrrupt enable
2	wb_transaction	
2.1	WB data	Coverpoint for wb transaction data
2.2	WB WE	Coverpoint for wb transaction op
2.3	WB Transaction cross	cross of coverpoints for wb covergroup
2.4	WB misc case	cross of coverpoints for csr, Intrrupt enable and we
3	i2c_transaction	
3.1	I2C Covergroup	i2c transaction covergroup
3.2	I2C Address	Coverpoint for transaction address
3.3	I2C data	Coverpoint for data
3.4	I2C operation	Coverpoint for operation
3.5	i2c cross transaction	cross of coverpoints for i2c covergroup
4	FSM Coverage	
4.1	FSM Covergroup	FSM Code Coverage Covergroup
4.2	Reg_type	type ofaddress register for fsm transition check
4.3	mbyte fsm state	check if byte fsm states are all covered
4.4	mbyte level transitions	Coverpoints for checking if all states and transitions are getting covered for byte level fsm

4.5	mbit fsm state	check if bit fsm states are all covered
4.6	mbit level transitions	Coverpoints for checking if all states and transitions are getting covered for bit level fsm
5	Randomize Tests	
5.1	Random Writes	test to write random 64 numbers to I2C slave
6	Code_Coverage	
6.1	RTL Core	"Ensure that all design units have 100% statement coverage100% branch coverage"

Link	Type	Weight	Goal
		1	100
i2cmb_register_cg	CoverGroup	1	100
i2cmb_register_cg::addr_valid	Coverpoint	1	100
i2cmb_register_cg::i2cmb_register_inits	Cross	1	100
invalid_test	test	1	100
i2cmb_register_cg::register_field_aliasing	Coverpoint	1	100
default_values	test	1	100
i2cmb_register_cg::bus_status	Coverpoint	1	100
i2cmb_register_cg::bus_cap_status	Coverpoint	1	100
i2cmb_register_cg::intr_enable	Coverpoint	1	100
		1	100
i2cmb_register_cg::data	Coverpoint	1	100
i2cmb_register_cg::we	Coverpoint	1	100
i2cmb_register_cg::addr_x_we	Cross	1	100
i2cmb_register_cg::csr_x_intr_x_we	Cross	1	100
		1	100
i2c_transaction_cg	CoverGroup	1	100
i2c_transaction_cg::addr	Coverpoint	1	100
i2c_transaction_cg::data	Coverpoint	1	100
i2c_transaction_cg::op	Coverpoint	1	100
i2c_transaction_cg::addr_x_op	Cross	1	100
		1	100
fsm_cg	CoverGroup	1	100
fsm_cg::addr	Coverpoint	1	100
fsm_cg::mbyte_state_cover	Coverpoint	1	100
fsm_cg::mbyte_state_transitions	Coverpoint	1	100

fsm_cg::mbit_state_cover	Coverpoint	1	100
fsm_cg::mbit_state_transitions	Coverpoint	1	100
		1	100
random_write	test	1	100

		1	100
/top/DUT	Instance	1	100

Owner	Manager	Priority
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	1
tkulkar	tkulkar	1
tkulkar	tkulkar	1
tkulkar	tkulkar	1
tkulkar	tkulkr	1
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2

tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	2
tkulkar	tkulkar	3