Design requirements				Verification Requirement			Coverage Links		Traceability Matrix											
#	Section	Description	Issue	Generate	Check	Cover	Link	Type	Weight	Goal	Priority-Importance	Priority-Speed	Priority-Effort	Risk(RE)	Author	Owner	Block	Config/Mode	Updated	Effort(Days)	Status	Current-Coverage
3	Data Validity																					
3,01	Asynchronous Reset	Both SDA and SCL are free and in HIGH state during the asynchronous reset.	How many resets did we end up with and what clocks are they related to?	Random stimulus	I2C VIP Model should reset during the asynchronous reset condition.	This requirement will be covered with total no. of Asynchronous reset coverage bin.	gen_random:REQ_CMD chk_ast_SDA_1 chk_ast_SCL_1 chk_scb_rst_values cov_i2c_request_cvgp:TERM_TYPE	CoverPoint Assertion Assertion Assertion CoverPoint	1	100	5	5	4	44	Andreas	Andreas	reset	Normal	27.05.2013	2	Written	32%
3,02	SCL and SDA kept high in I2C Idle state	Section 5: Both SDA and SCL are bi-directional lines, connected to a or pull-up resistor. When the bus is free, both lines are HIGH.		Random stimulus	SCL should not toggle when the bus is Idle.	This requirement will be covered with bus idle after reset or after detecting the stop.	gen_random:REQ_CMD chk_ast_SDA_1 chk_ast_SCL_1 cov_i2c_request_cvgp:TERM_TYPE	CoverPoint Assertion Assertion CoverPoint	1	100	5	4	2	39	Andreas	Andreas	intf	Normal	27.05.2013	4	Refined	26%
3,03	Data Validity	Section 6.1: A HIGH to LOW or LOW to HIGH transition on the SDA line can only occur when the clock signal on the SCL line is LOW		Random stimulus	Any time during the transaction toggle SDA while SCL is high other than START/RE-START/STOP conditions	None	gen_random:REQ_CMD chk_ast_SDA_transition	CoverPoint Assertion	1	100	5	3	5	39	Andreas	Andreas	intf	Normal	27.05.2013	5	Reviewed	71%
3,04	Data stability	The SDA must be stable during the HIGH phase of the clock on the SCL line; otherwise a transition will be interpreted as either a START or STOP command		I2C monitor verify this condition whenever SDA changes while SCL is high, other than START/RESTART/STOP condition.	During data/address/ACK phase and not during Start or Restart or Stop.	None	chk_ast_SDA_transition	Assertion	1	100	5	4	1	38	Andreas	Andreas	cntlfsm	Normal	27.05.2013	1	Implemented	94%
3,05	Transition of SDA when SCL is High	Transition of SDA when SCL is stable is allowed during the START, RESTART and STOP Phase		Random stimulus	During Start or Restart or Stop SDA can change to low or high while SCL is High.	This requirement will be covered in START/RESTART/STOP condition detection bins.	gen_random:REQ_CMD chk_ast_start chk_ast_stop cov_i2c_request_cvgp:TERM_TYPE	CoverPoint Directive Directive CoverPoint	1	100	5	4	4	41	Andreas	Andreas	cntlsfm	Normal	27.05.2013	7	Simulated	100%
3,06	Start Condition (S)	Section 6.2: A HIGH to LOW transition on the SDA line while SCL line is HIGH defines a START condition. The START sequence is initiated by a master when the bus is free/idle (i.e. SCL = 1 & SDA = 1). The START sequence signifies the beginning of a new data transfer.		Random stimulus	Check that the bus is free/idle prior to a master initiating a start sequence.	This requirement will be covered in total no. of START condition detection bins.	gen_random:TERM_TYPE chk_ast_start cov_i2c_request_cvgp:TERM_TYPE	CoverPoint Directive CoverPoint	1	100	5	3	5	39	Andreas	Andreas	arb	Normal	27.05.2013	2	Covered	0%
3,07	Invalid Start	If data transfer starts before generating a START condition.		I2C monitor verify this condition.	After Bus Idle State	None	chk_ast_SDA_transition	Assertion	1	100	5	3	2	36	Andreas	Andreas	abc	error	27.05.2013	2	Checked Off	32%
3,08	Stop Condition (P)	Section 6.2: A LOW to HIGH transition on the SDA line while SCL line is HIGH defines a STOP condition. The STOP sequence is initiated by a master to terminate a transaction.		Random stimulus	Master can initiate the STOP sequence when it wants to drop/finish the transaction with the slave. Every Start sequence needs to be followed by a STOP sequence.	This requirement will be covered in total no. of STOP condition detection bins.	gen_random:TERM_TYPE chk_ast_stop cov_i2c_request_cvgp:TERM_TYPE	CoverPoint Directive CoverPoint	1	100	5	4	4	41	Andreas	Andreas	top	cep	27.05.2013	4	Written	26%
3,09	Invalid Stop	When bus is in idle state, if a STOP is detected it is considered as an invalid condition.		I2C monitor verify this condition.	During the idle condition	None	chk_ast_illegal_stop	Assertion	1	100	4	4	4	36	Andreas	Andreas	arb	Normal	27.05.2013	5	Refined	71%
3.10	Bus busy after START	Section 6.2: The bus should be in busy state after a START is generated by the master.		Random stimulus	Between a START and STOP sequence. User cannot generate another START condition while bus is in busy state. Restart condition is possible when bus is in busy state.	This requirement will be covered in total no. of RESTART, Valid Data Read and Write condition detection bins.	gen_random:REQ_CMD chk_ast_start_stop cov_i2c_request_cvgp:TERM_TYPE cov_i2c_request_cvgp:REQ_CMD	CoverPoint Assertion CoverPoint CoverPoint	1	100	3	3	2	26	Andreas	Andreas	cfsn	Normal	27.05.2013	1	Reviewed	94%

3,11	Repeated Start (Sr)	Section 6.2: A repeated start is functional equivalent to a start (S-bit) transaction; each subsequent start transaction must adhere to the setup and hold time define in Table 7; there can be up to N restarts as long as they are separated by a STOP sequence.		Random stimulus	Repeat START should be identified between a STOP and a START sequence for Master changing the Slave address or Read or Write transitions.	This requirement will be covered in total no. of RESTART condition detection bins.	gen_random:TERM_TYPE chk_ast_repeat_start cov_i2c_request_cvgp:TERM_TYPE	CoverPoint Assertion CoverPoint	1	100	4	2	5	31	Andreas	Andreas	xpipe	Normal	27.05.2013	7	Implemented	100%
3,12	START & STOP with out data / address byte	A START condition immediately followed by STOP condition is an illegal format		I2C monitor verify this sequence based on data or address received on the bus after start condition	Evaluated after START and a STOP is detected. Every START signal should follow at least one data phase	This requirement will be covered in total no. of START, STOP condition detection bins.	chk_ast_start_nodata cov_i2c_request_cvgp:TERM_TYPE	Assertion CoverPoint	1	100	3	5	5	35	Andreas	Andreas	intf	error	27.05.2013	2	Simulated	0%
3,13	START & RESTART with out data / address byte	A START condition immediately followed by RESTART condition is an illegal format		I2C monitor verify this sequence based on data or address received on the bus after start condition	Evaluated after a START is detected. Every START signal should follow at least one data phase	This requirement will be covered in total no. of START, RESTART condition detection bins.	chk_ast_start_nodata cov_i2c_request_cvgp:TERM_TYPE	Assertion CoverPoint	1	100	4	1	1	24	Andreas	Andreas	intf	cep	27.05.2013	2	Covered	94%
3,14	RESTART & STOP with out data / address byte	A RESTART condition immediately followed by STOP condition is an illegal format		I2C monitor verify this sequence based on data or address received on the bus after restart condition	Evaluated after a RESTART is detected. Every RESTART signal should follow at least one data phase	This requirement will be covered in total no. of RESTART, STOP condition detection bins.	chk_ast_restart_nodata cov_i2c_request_cvgp:TERM_TYPE	Assertion CoverPoint	1	100	4	5	3	38	Andreas	Andreas	cntlfsm	Normal	27.05.2013	4	Checked Off	100%
3,15	RESTART & RESTART with out data / address byte	A RESTART condition immediately followed by another RESTART condition is an illegal format		I2C monitor verify this sequence based on data or address received on the bus after restart condition	Evaluated after a RESTART is detected. Every RESTART signal should follow at least one data phase	This requirement will be covered in total no. of RESTART condition detection bins.	chk_ast_restart_nodata cov_i2c_request_cvgp:TERM_TYPE	Assertion CoverPoint	1	100	3	2	4	25	Andreas	Andreas	cntlsfm	Normal	27.05.2013	5	Implemented	0%
4	Transfer of data																arb	Normal	27.05.2013	1	Simulated	32%
4,01	Address Generation	The first byte of data transferred by the master immediately after the start sequence in the slave address. The address is comprised of a 7-bit address followed by a 1-bit read/write bit.		Random stimulus	After the Start sequence next byte has to be Slave address or I2C reserved address.	No of address transmitted in the bus will be covered in total no. of address detected coverage bin.			1	100	5	5	1	41	Bruno	Bruno	abc	error	27.05.2013	7	Covered	26%
4,02	Address Uniqueness	No two slaves in the system can have the same address.		I2C monitor verify this condition through the slave device map input.	During Slave addressing no two slaves should not give a valid ack for address matching other than General call address.	The slave address valid or not valid ack will be covered in total no. of Slave ACK for address match.			1	100	5	1	3	31	Bruno	Bruno	top	cep	27.05.2013	2	Checked Off	71%
4,03	Slave Response	Only the slave with a matching address to the one broadcasted by the master will respond by returning an acknowledge by pulling the SDA line low the cycle after the read/write bit (i.e. 9th cycle).		Random stimulus	After 8 bit data transfer from the master. The does not match his slave address should not respond with valid ACK.	The slave address valid or not valid ack will be covered in total no. of Slave ACK for address match.			1	100	5	4	4	41	Bruno	Bruno	arb	Normal	27.05.2013	5	Reviewed	94%
4,04	Data Write	Once a slave acknowledges an address request on master write, the master can write a payload of length 1 to N bytes, as long as the slave acknowledges each byte.		Random stimulus	After valid slave ACK is received. Master should stop data transfer and generate Restart or STOP condition on the bus if the slave is not providing a valid ACK for each write data transfer.	This master write operation is covered in total no. of valid master write coverage bin.			1	100	5	3	3	37	Bruno	Bruno	cfsn	Normal	27.05.2013	3	Implemented	100%
4,05	Data Read	Once a slave acknowledges an address request on master read, the master can read a payload of length 1 to N bytes, as long as the master acknowledges each byte.		Random stimulus	After the master gives a valid ACK for master read operation slave should keep on deliver data to master until master generate a no ack on the bus.	This master read operation is covered in total no. of valid master read coverage bin.			1	100	5	2	2	33	Bruno	Bruno	xpipe	Normal	27.05.2013	2	Simulated	0%

4,06	No ack for master write transaction	If while the master is writing a payload and the slave fails to acknowledge the byte during the 9th clock cycle, the master has the right to issue a STOP/RESTART sequence to abort the data transfer.		I2C monitor verify this condition	During master write master has to generate a Restart or STOP after receiving no ack from addressed slave.	This master write operation is covered in total no. of valid master write coverage bin.			1	100	5	1	5	33	Bruno	Bruno	intf	error	27.05.2013	2	Covered	94%
4,07	No ack from master during the master read transaction	If while the master is reading a payload and does not acknowledge the slave during the 9th clock cycle, the slave will release the SDA line for the master to generate a STOP or RESTART sequence.		Random stimulus	During master read mode master has to generate a Restart or STOP after sending a no ack to addressed slave.	This master read operation is covered in total no. of valid master read coverage bin.			1	100	5	5	5	45	Bruno	Bruno	intf	cep	27.05.2013	4	Checked Off	100%
4,08	Transaction Size without Restart	Section 7.1: Every transaction put on the SDA line must consist of 8 bits + ACK. Another way of looking at this is that data between START and STOP should be divisible by 9.		Random stimulus	Transaction Size is divisible by 9 provided a Restart not present between the Start and Stop.	This operation is covered in total no. of valid master write/read and address coverage bins.			1	100	5	4	4	41	Bruno	Bruno	cntlfsm	Normal	27.05.2013	5	Written	0%
4,09	Transaction Size with Restart	Section 7.1: Every transaction put on the SDA line must consist of 8 bits + ACK.		Random stimulus	Transaction Size is not divisible by 9 provided a Restart present between the Start and Stop	This operation is covered in total no. of valid master write/read, address and Restart coverage bins.			1	100	5	2	5	36	Bruno	Bruno	cntlsfm	Normal	27.05.2013	1	Refined	32%
4.10	Endianess	MSB must be transmitted first during the transaction.		Random stimulus	Applicable in both transmission and reception phase. MSB should transmit first for each byte transaction on the bus.	None			1	100	5	4	3	40	Bruno	Bruno	arb	Normal	27.05.2013	7	Reviewed	26%
4,11	CBUS Transaction Size	Section 7.1: Every transaction put on the SDA line need not be multiples of 8bits .		Generate a direct test case for master initiating a CBUS transaction on the bus.	On CBUS address detection	This CBUS operation is covered in total no. of CBUS address detection bins.			1	100	4	3	3	32	Bruno	Bruno	abc	error	27.05.2013	2	Implemented	71%
4,12	Acknowledge	Section 7.2: Each transmitted byte (8 bits) must be acknowledged, during the 9th clock cycle, by the slave by pulling the SDA line low during the HIGH phase of the clock period. A valid LOW is recognized as an ACK and High as no ACK.		Random stimulus	No ACK during START BYTE, No ACK during GCA and master read mode(during last byte to indicate the slave that master is going to generate the P or Sr)	The slave ack will be covered in total no. of Slave ACK or NACK coverage bins.			1	100	4	2	5	31	Bruno	Bruno	top	cep	27.05.2013	5	Simulated	94%
4,13	Condition 1:Bus Contention during ACK	Ensure that Master releases the SCL line after generating the ACK pulse for slave clock stretch.		Random stimulus	During Master/Slave ACK receive mode	The slave ack will be covered in total no. of Slave ACK or NACK coverage bins.			1	100	5	5	2	42	Bruno	Bruno	arb	Normal	27.05.2013	3	Covered	100%
4,14	Condition 2:Bus Contention during ACK	Ensure that the Master drives the SCL line after the slave ACK clock stretch high timeout.		Random stimulus	During Master ACK receive mode	The slave ack will be covered in total no. of Slave ACK or NACK coverage bins.			1	100	5	2	1	32	Bruno	Bruno	cfsn	Normal	27.05.2013	2	Checked Off	0%
4,15	Condition 3:Bus Contention during ACK	Ensure that the slave drives the SDA line during the slave ACK pulse		Random stimulus	During Master ACK receive mode	The slave ack will be covered in total no. of Slave ACK or NACK coverage bins.			1	100	5	5	3	43	Bruno	Bruno	xpipe	Normal	27.05.2013	4	Implemented	94%
4,16	Condition 4:Bus Contention during ACK	Ensure that master releases the SDA line during the slave ACK pulse.		Random stimulus	During Master ACK receive mode	The slave ack will be covered in total no. of Slave ACK or NACK coverage bins.			1	100	5	4	4	41	Bruno	Bruno	xctl	error	27.05.2013	5	Simulated	100%
4,17	Stop / Restart after No ACK	Section 7.2:A STOP or a Restart is required after no ACK is detected.		Random stimulus	No ACK cycle is allowed during the data byte transfer.	The slave ack will be covered in total no. of Slave ACK or NACK coverage bins.			1	100	5	4	2	39	Bruno	Bruno	rstart	cep	27.05.2013	1	Covered	0%