

Feature	Stimulus	Checker List	Priority
Regular Write with odd number of data bytes & there are no coming configurations to be executed	<pre>i_sys_rst_tb = 1; i_engine_en_tb = 1; i_regf_toc_tb = 1; i_regf_dev_index_tb equals any value except 0;</pre>	The sequence of sending command word -> first data byte -> second data byte -> third data byte ->> Sending crc word -> EXIT pattern	High
	<pre>i_regf_short_read_tb = x; i_regf_wr_rd_bit_tb =</pre>	Flag o_engine_done set to 1 after exit pattern	High
	0; i_regf_cmd_attr_tb = 0; i_regf_dtt_tb = xxx;	The slave recognizes its address and pulls down the line to 0 means as an ack	High
	i_regf_DATA_LEN_tb = odd number	A dummy data is sent to complete the data word bits	High
		The slave doesn't abort the write operation	High



Feature	Stimulus	Checker List	Priority
Regular Write with	i_sys_rst_tb = 1;	The sequence of sending command	High
odd number of data	i_engine_en_tb =1;	word -> first data byte -> second data	
bytes & there are	<pre>i_regf_toc_tb = 0;</pre>	byte -> third data byte ->>	
<b>coming</b> configurations	i_regf_dev_index_tb	Sending crc word -> RETART	
to be executed	equals any value except	pattern	
	0;		
	i_regf_short_read_tb =	Flag o_engine_done set to 1 after exit	High
	x;	pattern	
	i_regf_wr_rd_bit_tb =		
	0;	The slave recognizes its address and	High
	i_regf_cmd_attr_tb =	pulls down the line to 0 means as an	
	0;	ack	
	i_regf_dtt_tb = xxx;		
	i_regf_DATA_LEN_tb	A dummy data is sent to complete the	High
	= odd number	data word bits	
		The slave doesn't abort the write	High
		operation	



Feature	Stimulus	Checker List	Priority
Regular Write with even number of data bytes & there are no coming configurations to be executed	<pre>i_sys_rst_tb = 1; i_engine_en_tb = 1; i_regf_toc_tb = 1; i_regf_dev_index_tb equals any value except 0;</pre>	The sequence of sending command word -> first data byte -> second data byte -> third data byte ->>  Sending crc word -> EXIT pattern	High
	<pre>i_regf_short_read_tb = x; i_regf_wr_rd_bit_tb =</pre>	Flag o_engine_done set to 1 after exit pattern	High
	0; i_regf_cmd_attr_tb = 0; i_regf_dtt_tb = xxx;	The slave recognizes its address and pulls down the line to 0 means as an ack	High
	i_regf_DATA_LEN_tb = even number	The slave doesn't abort the write operation	High



Feature	Stimulus	Checker List	Priority
Regular Write with	i_sys_rst_tb = 1;	The sequence of sending command	High
even number of data	i_engine_en_tb =1;	word -> first data byte -> second data	
bytes & there are	<pre>i_regf_toc_tb = 0;</pre>	byte -> third data byte ->>	
<b>coming</b> configurations	i_regf_dev_index_tb	Sending crc word -> EXIT pattern	
to be executed	equals any value except		
	0;		
	i_regf_short_read_tb =	Flag o_engine_done set to 1 after exit	High
	x;	pattern	
	i_regf_wr_rd_bit_tb =		
	0;	The slave recognizes its address and	High
	i_regf_cmd_attr_tb =	pulls down the line to 0 means as an	
	0;	ack	
	i_regf_dtt_tb = xxx;		
	i_regf_DATA_LEN_tb	The slave doesn't abort the write	High
	= even number	operation	



Feature	Stimulus	Checker List	Priority
Regular read with odd	i_sys_rst_tb = 1;	The sequence of sending command	High
number of data bytes &	i_engine_en_tb =1;	word -> first data byte -> second data	
there are <b>no coming</b>	i_regf_toc_tb = 1;	byte -> third data byte ->>	
configurations to be	i_regf_dev_index_tb	receiving crc word -> EXIT pattern	
executed & allowing	equals any value except		
short read	0;	Flag o_engine_done set to 1 after exit	High
	i_regf_short_read_tb =	pattern	
	1;		
	i_regf_wr_rd_bit_tb =	The slave recognizes its address and	High
	1;	pulls down the line to 0 means as an	
	i_regf_cmd_attr_tb =	ack	
	0;		
	i_regf_dtt_tb = xxx;		
	i_regf_DATA_LEN_tb	The slave sends all the required data	High
	= odd number		
	$i_rx_sda_tb = 0$ after	A dummy data is sent to complete the	High
	the last frame	data word bits	
	indicating to <b>crc</b>		



Feature	Stimulus	Checker List	Priority
Regular read with odd	i_sys_rst_tb = 1;	The sequence of sending command	High
number of data bytes &	i_engine_en_tb =1;	word -> first data byte -> second data	
there are <b>coming</b>	i_regf_toc_tb = 0;	byte -> third data byte ->>	
configurations to be	i_regf_dev_index_tb	receiving crc word -> RESTART	
executed & allowing	equals any value except	pattern	
short read	0;		
	i_regf_short_read_tb =	Flag o_engine_done set to 1 after exit	High
	1;	pattern	
	i_regf_wr_rd_bit_tb =		
	1;	The slave recognizes its address and	High
	i_regf_cmd_attr_tb =	pulls down the line to 0 means as an	
	0;	ack	
	i_regf_dtt_tb = xxx;		
	i_regf_DATA_LEN_tb	The slave sends all the required data	High
	= odd number		
	i_rx_sda_tb = 0 after	A dummy data is sent to complete the	High
	the last frame	data word bits	
	indicating to <b>crc</b>		



Feature	Stimulus	Checker List	Priority
Regular read with	i_sys_rst_tb = 1;	The sequence of sending command	High
even number of data	i_engine_en_tb =1;	word -> first data byte -> second data	
bytes & there are	<pre>i_regf_toc_tb = 0;</pre>	byte -> third data byte ->>	
<b>coming</b> configurations	i_regf_dev_index_tb	receiving crc word -> RESTART	
to be executed &	equals any value except	pattern	
allowing short read	0;		
	i_regf_short_read_tb =	Flag o_engine_done set to 1 after exit	High
	1;	pattern	
	i_regf_wr_rd_bit_tb =		
	1;	The slave sends all the required data	High
	i_regf_cmd_attr_tb =		
	0;		
	i_regf_dtt_tb = xxx;	The slave sends all the required data	High
	i_regf_DATA_LEN_tb		
	= even number		
	$i_rx_sda_tb = 0$ after		
	the last frame		
	indicating to <b>crc</b>		



Feature	Stimulus	Checker List	Priority
Regular read with	i_sys_rst_tb = 1;	The sequence of sending command	High
even number of data	i_engine_en_tb =1;	word -> first data byte -> second data	
bytes & there are <b>no</b>	$i_regf_toc_tb = 1$ ;	byte -> third data byte ->>	
<b>coming</b> configurations	i_regf_dev_index_tb	receiving crc word -> EXIT pattern	
to be executed &	equals any value except		
allowing short read	0;	Flag o_engine_done set to 1 after exit	High
	i_regf_short_read_tb =	pattern	
	1;		
	i_regf_wr_rd_bit_tb =	The slave sends all the required data	High
	1;		
	i_regf_cmd_attr_tb =	The slave recognizes its address and	High
	0;	pulls down the line to 0 means as an	
	i_regf_dtt_tb = xxx;	ack	
	i_regf_DATA_LEN_tb		
	= even number		
	$i_rx_sda_tb = 0$ after		
	the last frame		
	indicating to <b>crc</b>		



Feature	Stimulus	Checker List	Priority
Immidiate Write with	$i_sys_rst_tb = 1;$	The sequence of sending command	High
<b>one</b> data byte & there	i_engine_en_tb =1;	word -> first data byte dummy byte ->	
are <b>coming</b>	i_regf_toc_tb = 0;	> <b>Sending</b> crc word ->	
configurations to be	i_regf_dev_index_tb	RESTART pattern	
executed	equals any value except		
	0;	Flag o_engine_done set to 1 after exit	High
	i_regf_short_read_tb =	pattern	
	x;		
	i_regf_wr_rd_bit_tb =		High
	0;	The slave recognizes its address and	
	i_regf_cmd_attr_tb =	pulls down the line to 0 means as an	
	1;	ack	
	i_regf_dtt_tb = xxx;		
	i_regf_DATA_LEN_tb		
	= even number		



Feature	Stimulus	Checker List	Priority
Immidiate Write with	i_sys_rst_tb = 1;	The sequence of sending command	High
<b>one</b> data byte & there	i_engine_en_tb =1;	word -> first data byte dummy byte ->	
are <b>no coming</b>	<b>i_regf_toc_tb = 1</b> ;	> Sending crc word -> EXIT	
configurations to be	i_regf_dev_index_tb	pattern	
executed	equals any value except		
	0;	Flag o_engine_done set to 1 after exit	High
	i_regf_short_read_tb =	pattern	
	x;		
	i_regf_wr_rd_bit_tb =		High
	0;	The slave recognizes its address and	
	i_regf_cmd_attr_tb =	pulls down the line to 0 means as an	
	1;	ack	
	i_regf_dtt_tb = 1;		
	i_regf_DATA_LEN_tb		
	= even number		



Feature	Stimulus	Checker List	Priority
Immidiate Write with	i_sys_rst_tb = 1;	The sequence of sending command	High
<b>two</b> data bytes & there	i_engine_en_tb =1;	word -> first data byte dummy byte ->	
are <b>coming</b>	i_regf_toc_tb = 0;	> <b>Sending</b> crc word ->	
configurations to be	i_regf_dev_index_tb	RESTART pattern	
executed	equals any value except		
	0;	Flag o_engine_done set to 1 after exit	
	i_regf_short_read_tb =	pattern	
	x;		
	i_regf_wr_rd_bit_tb =		
	0;	The slave recognizes its address and	High
	i_regf_cmd_attr_tb =	pulls down the line to 0 means as an	
	1;	ack	
	$i_regf_dtt_tb = 2;$		
	i_regf_DATA_LEN_tb		
	= even number	Flag o_engine_done set to 1 after exit	High
		pattern	



Feature	Stimulus	Checker List	Priority
Immidiate Write with	i_sys_rst_tb = 1;	The sequence of sending command	High
<b>two</b> data bytes & there	i_engine_en_tb =1;	word -> first data byte dummy byte ->	
are <b>no coming</b>	i_regf_toc_tb = 1;	> Sending crc word -> EXIT	
configurations to be	i_regf_dev_index_tb	pattern	
executed	equals any value except		
	0;	Flag o_engine_done set to 1 after exit	High
	i_regf_short_read_tb =	pattern	
	x;		
	i_regf_wr_rd_bit_tb =		High
	0;	The slave recognizes its address and	
	i_regf_cmd_attr_tb =	pulls down the line to 0 means as an	
	1;	ack	
	<b>i_regf_dtt_tb = 2;</b>		
	i_regf_DATA_LEN_tb		
	= even number		



Feature	Stimulus	Checker List	Priority
Immidiate Write with	i_sys_rst_tb = 1;	The sequence of sending command	High
three data bytes &	i_engine_en_tb =1;	word -> first data byte dummy byte ->	
there are <b>coming</b>	i_regf_toc_tb = 0;	> <b>Sending</b> crc word ->	
configurations to be	i_regf_dev_index_tb	RESTART pattern	
executed	equals any value except		
	0;	Flag o_engine_done set to 1 after exit	High
	i_regf_short_read_tb =	pattern	
	x;		
	i_regf_wr_rd_bit_tb =		High
	0;	The slave recognizes its address and	
	i_regf_cmd_attr_tb =	pulls down the line to 0 means as an	
	1;	ack	
	<b>i_regf_dtt_tb = 3</b> ;		
	i_regf_DATA_LEN_tb		
	= even number		



Feature	Stimulus	Checker List	Priority
Immidiate Write with	i_sys_rst_tb = 1;	The sequence of sending command	High
three data bytes &	i_engine_en_tb =1;	word -> first data byte dummy byte ->	
there are <b>no coming</b>	i_regf_toc_tb = 1;	> Sending crc word -> EXIT	
configurations to be	i_regf_dev_index_tb	pattern	
executed	equals any value except		
	0;	Flag o_engine_done set to 1 after exit	High
	i_regf_short_read_tb =	pattern	
	x;		
	i_regf_wr_rd_bit_tb =		High
	0;	The slave recognizes its address and	
	i_regf_cmd_attr_tb =	pulls down the line to 0 means as an	
	1;	ack	
	<b>i_regf_dtt_tb = 3</b> ;		
	i_regf_DATA_LEN_tb		
	= even number		



Feature	Stimulus	Checker List	Priority
The Master in The Reset Mode	i_sys_rst_tb = 1; i_sys_rst_tb = 1->0;	@ negedge of i_sys_rst_tb the master enters the rst mode	High
Reset Mode	$i_sys_rst_tb = 0$ ;	enters the 1st mode	
	i_sys_rst_tb = 0->1;	o_scl_tb line is set to 1 & o_sdahnd_serial_data_tb is set to 1 (open drain) within time i_sys_rst_tb is being 0	High
		@ posedge of i_sys_rst_tb the master operates normally	High

Feature	Stimulus	Checker List	Priority
Enabling N.T Block	i_sys_rst_tb = 1;	The NT Block doesn't respond until	High
	i_engine_en_tb =1;	being enabled	
		If the enable is high the block excute	High
	i_engine_en_tb =0;	the required configuration and	
		respond with done	



Feature	Stimulus	Checker List	Priority
No Acknowledgment	i_sys_rst_tb = 1;	The sequence of sending command	High
for Write or Read	i_engine_en_tb =1;	word -> Error (no ack) -> RESTART	
operations	i_regf_wr_rd_bit_tb =	or EXIT depending on the value of	
	0 or 1;	(i_regf_toc)	
	i_sda_rx_tb = 1 @ the		
	beginning of the first data word	The type of error flag is NACK_Error	High

Feature	Stimulus	Checker List	Priority
<b>Aborting</b> Write	i_sys_rst_tb = 1;	The sequence of sending command	High
operation	i_engine_en_tb =1;	word -> first data byte -> second data	
	i_regf_wr_rd_bit_tb =	byte -> third data byte ->>	
	0;	<b>Error(aborting)</b> -> RESTART or	
		EXIT depending on the value of	
	$i_sda_rx_tb = 0$ @ the	(i_regf_toc)	
	beginning of the	The slave aborts the write operation	High
	second or third or	and the type of error flag is	
	fourth or data word	BUS_ABORTED_Error	

Feature	Stimulus	Checker List	Priority
Framing Error	i_sys_rst_tb = 1;	The sequence of sending command	High
	i_engine_en_tb =1;	word -> first data byte -> second data	
	i_regf_wr_rd_bit_tb =	byte -> third data byte ->>	
	1;	Error(unexpected bits) ->	
		RESTART or EXIT depending on the	
		value of (i_regf_toc)	
		the preamble bits of any word ,crc	High
		token values, parity and crc value are	
		not correct	
		type of error flag is (CRC_Error,	
		Parity_Error	

