# A Generic Randomization Framework Architecture for Test Execution in Automated Testing Of SoC

Subinoy Das<sup>1</sup>, Surendra Shamanna<sup>2</sup>

<sup>1</sup>(IFIN ATV PSV, Infineon Technologies India Pvt. Ltd., Bangalore, India) <sup>2</sup>(IFIN ATV PSV, Infineon Technologies India Pvt. Ltd., Bangalore, India)

**Abstract :** Conventional directed tests lack the flexibility to cover different variations of test configurations. To overcome the problem and obtain better coverage, randomization is adopted. Generic randomization framework for test execution in automated testing of SoC is effective for generating real time stimulus that covers variations in test configurations such as different functional modes, register configurations, and network data packets. It provides a way to capture the state of the DUT (Device Under Test) at randomly generated parameter configurations. It prevents the selection bias and also accidental bias. It eradicates the origin of bias and provides better bug fixation in automated testing of SoC in post-silicon validation.

**Keywords:** Automated Validation Execution Flow, Post Silicon Validation, Randomization Framework, Test Automation

## I. Introduction

Due to high complexity of modern designs and increasing pressure to reduce their time-to-market, bugs can escape the pre silicon verification environment. Therefore in order to check for extreme cases and escaped bugs post silicon environment is used effectively. In order to achieve the "zero defect SoC" (System on Chip), it is required to check whether all the possible configurations/features of the SoC are working as expected or not. This involves testing of SoC using huge number of tests manually and consumes a lot of time, and is error prone as well. Thus, automation reduces the process time for the validation of SoC. This can be done by using an automation tool to execute the test flows and writing a suitable process for controlling the test execution on the DUT (Device Under Test). Randomization framework forms a part of the automated SoC validation process. The randomization automation framework can configure test parameters and pass them to the DUT, and generate status report.

Generic randomization framework provides a way to capture the state of the DUT. It provides randomly generated test configurations for test execution. Absence of bias means more reliable tests for automated SoC validation process. Ultimate goal of randomization is to ensure that each configuration of parameters is equally likely to be assigned to the test execution, so that extreme cases are checked and better coverage is obtained.

## II. Scope Of Work

Generic randomized test framework architecture automates the generation of random test configurations for test execution. It provides a way for passing execution parameters to the DUT and save the recorded CPU state build up. This saved state buildup of the DUT is used to bring it back to the last known state just before failure and then execute the smaller set of execution parameters to the point of failure, thus requiring lesser execution time. Randomization framework enables the execution of smaller set of configurations more frequently and requires shorter reproduction time. Generic randomization framework will generate a random set of reproducible parameters and pass it to the test executable. After accepting the parameters of the framework, the test will carry out its operations for validation of SoC. Generic randomization framework is solely responsible for the passing of random reproducible parameter values to the test executable at run time.

## III. Methodology

Randomization framework consists of designing a process that can be run through the automation tool and it will start execution of the test on the DUT. "Fig. 1" shows the basic block diagram of the randomization framework. "Fig 4" shows the features/functions in the randomization framework.

Selection of parameters will be random in nature and constant for a particular seed number and seed type, so that the effect is reproducible and repeatable. The set of selected parameters/test configurations will be passed to the DUT, and then the DUT will be reconfigured repeatedly with/without performing any reset for the next iteration cycles based on the number of seed iterations.

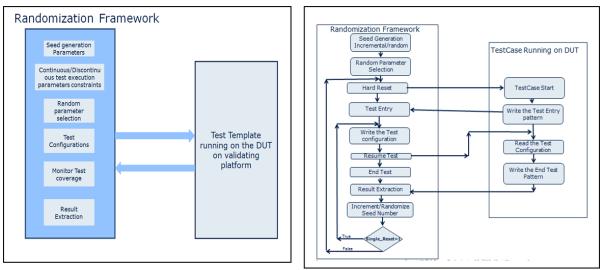


Figure 1. Randomization framework block diagram

Figure 4. Functional flow for the randomization framework

**Random value generation**- Generation of randomization parameters includes obtaining the random numbers through a random number generator based on a specific seed number [Input Value], so that the event is reproducible. The randomly generated parameter values are then constrained by user-defined continuous/discontinuous limits. And selected parameter values are passed to the test execution. This technique maintains complete randomness of the assignment of parameters to a particular test executable configuration. Seed number forms the nodes for the chain of random parameters generated. If seed number and seed type are known at any node point, the set of events can be reproduced again.

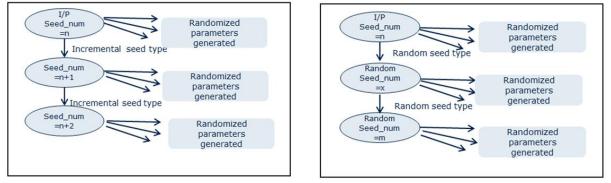


Figure 2. Random number generation using incremental seed type Figure 3. Random number generation using random seed type

**Incremental / Random Seed Type** - Incremental seed number introduces a single level of randomization. In this, seed number is incremental in nature. Based on the seed number, the random parameters are selected from user-defined parameter constraints forming single level of randomization. "Fig 2" shows the generation of random parameters using incremental seed type. Random seed number introduces two levels of randomization. In the first level, seed number is randomized and, in the second level depending on seed number the random parameters are selected from user-defined constraints. "Fig 3" shows the generation of random parameters are selected from user-defined constraints. "Fig 3" shows the generation of random parameters using random seed type.

**DUT Reset once / Reset on every test configuration** - Reset once /Reset on every test configuration can be controlled by the "single reset" flag option. If "single reset" flag is set, then DUT will reset only once, the next test configurations will be passed without resetting the device. Resetting only once ensures CPU state buildup. If 'single reset' flag is not set, then DUT will reset on every iteration before passing the test configurations. Resetting on every iteration ensures the recovery of the DUT from failure, before passing test configurations.

**Synchronization between Test execution and Automation flow** - For maintaining synchronization between test execution and automation flow, shared addresses of DUT are used. Using automation tool's built-in functions for memory reading and writing, control words can be used for establishing a semaphore mechanism

to pass configurations from automation flow to the test execution, and get results back from test execution to the automation flow.

**Test considered** - For validating the generic randomization framework, a use-case of SENT (Single Edge Nibble Transmission) Normal Frame is considered whose intent is to check for all possible frequencies and data values for the SENT Normal Frame Format by configuring the GTM (Generic Timer Module) for generating these SENT frames. "Fig. 5" shows the automation tool's GUI (Graphical User Interface) for the randomization framework, test parameters can be entered in continuous/discontinuous parameter format. Continuous/Discontinuous parameters can be specified in [min-max]/[value1, value2, value3...] format.

Table 1 shows the "Randomization framework field values" which is same for all tests. Table 2 shows the "Test execution parameters" which are specific to the SENT Normal test.

### Considered Test's Flow -

#### Step [1].Begin.

Step [2].Initialization of GTM.

**Step [3].** Test accepts selected parameters and test iteration from the Randomization framework through shared addresses, using semaphore mechanism.

**Step [4].**Then test performs frame and CRC (Cyclic Redundancy Check) calculations, and loads data into the FIFO (First In First Out). Then GTM and SENT are configured to send the data.

Step [5].It waits for 500 interrupts, for the data to be received in the SENT RD (SENT Receive Data) register.

**Step [6].** After that counter and ATOM (Advanced Routing Unit connected Timer Output Module) channel is disabled, compare and shadow registers are cleared.

**Step [7].**Then, based on the test iteration which is being passed by the Automation flow process, step[3]-step-[6] is performed again in an iterative way.

Step [8].Exit.

Table 1: Ran	domization Framework Fields Table 2				
Field Name	Field Description				
Config file	Configuration file with respect to the DUT				
Hex file	Binary file to be executed on DUT				
MC boot config	Boot configuration file for the				
file	MC(Management Control)flow				
TimeOut	Timeout time for test execution				
Program	Selection of core [0/1/2] for downloading test				
execution by	executable.				
Is Serial Prints	Enable/Disable Serial prints for the result				
Enabled?	extraction during test execution				
Seed Type?	Incremental/Random Seed type for random number generation				
Seed Number	Based on this value, random parameter values will be generated for the test				
Seed Iteration	Number of seeds used for test execution.				
Parameter[1-30]	Test execution parameters.[ min-max]				
ratameter[1-30]	/[value1, value2,]				
Single Reset	DUT Reset once/Reset on every test configuration				

Fields	Table 2: Sent Normal Frame	" Test Execution Parameters
--------	----------------------------	-----------------------------

Test Parameter Number	Test Parameter Name	Test Acceptable Range
1	Data Nibble	0x1-0xf
2	Data Nibble	0x1-0xf
3	Data Nibble	0x1-0xf
4	Data Nibble	0x1-0xf
5	Data Nibble	0x1-0xf
6	Data Nibble	0x1-0xf
7	Clock Ticks	0x1-0x5a
8	Divider mode	0x0-0x1
9	SENT Channel	0x0-0x9
10	SENT ALT.	0x1-0x2
	Channel	

## **Randomization framework Process flow -**

#### Step [1].Begin.

**Step [2].** Fetch the seed number, seed type, "single reset" and seed iteration for the randomization framework and hex-file, configuration file, MC boot configuration file for the SENT Normal Frame test from the user.

**Step [3].** Fetch the test parameters from the framework.

**Step [4].** The parameters in [min-max]/[value1,value2,value3....]format is separated and converted to hexadecimal format for transferring it to memory.

**Step [5].** If "single reset" field is set, then seed iteration becomes equal to the test iteration, and if this field is not set then test iteration becomes equal to unity.

**Step [6].** Downloading of the test executable for SENT Normal Frame test takes place in the SoC, and just after that pre-configurations like initializations for clock, and address definitions are done.

**Step [7].** Based on the "single reset" flag set or not in the framework hard reset is being performed for the SoC .If "single reset" flag is set, hard reset is performed only once otherwise reset is performed on every seed iteration.

**Step [8].** The test writes a control word [Entry pattern] in the shared memory to indicate beginning of the test execution, and reads number of test iteration which is being passed from the automation tool process through shared memory address.

**Step [9].** Test polls for the control word [resume pattern] in the same memory address which is going to be written from the automation flow, to resume the test.

**Step [10].** Based on the seed number and seed type, a random parameter value will be selected in the user specified parameter range.

**Step [11].** The random values generated, are written to the shared memory addresses based on the number of parameters from the automation process.

**Step [12].** After that, the control word for resuming test is written from the automation process .With this, the test resumes and reads all the randomized parameter data values from the shared memory addresses.

**Step [13].** The test executes for all the specified parameter configurations from the framework and the data is transmitted from one port to another

**Step [14].** The functional coverage of the test can be monitored using prints from the test directly on to automation tool's log file, which is shown in the result section

**Step [15].** After the test is executed, the test will write a control word to indicate the end of the of the test execution.

**Step [16].** Based on the number of seed iterations and "single reset" flag being set or not, process iterations are performed. If the "single reset" flag is set, the steps 9-15 are repeated. And if the flag is not set, the steps from 7-15 are performed in an iterative way.

**Step [17].** Exit.

ame:	flash download		-	Description
		114		
	0		ique ID: -102	5766708
ard Bin:	-1	5	Soft Bin: -1	
Device:	Chip			•
Unit:	Chip			-
Action:	V Test (by external Lang	uage)		•
Ext. Script:	FlashDownload.pl			•
Options				
Notes: Keys: Product(version): TC Config file: Hex file:	and the second	cfg/TriBoard_TC27xC_core0_d sktop/sram1.hex		Search Search
MC_boot_config_f	file: MC_boot_config_file			Search
Time out:	50	Program execution by:	Core0	-
Is Serial Prints E	nabled?	Seed_Type?:	Random	-
Seed_Number:	12	Seed_Iteration:	3	
Parameter1:	Ox1-Oxf	Parameter2:	Ox1-Oxf	
Parameter3:	Ox1-Oxf	Parameter4:	Ox1-Oxf	
Parameter5:	Ox1-Oxf	Parameter6:	Ox1-0xf	
Parameter7:	0x1-0x5a	Parameter8:	0x0-0x1	
Parameter9:	0x0-0x9	Parameter10:	0x1-0x1	
Parameter11:	0	Parameter12:	0	
Parameter13:	0	Parameter14:	0	
Parameter15:	0	Parameter16:	0	
Parameter17:	0	Parameter18:	0	
Parameter 19:	0	Parameter20:	0	
Parameter21:	0	Parameter22:	0	
Parameter23:	0	Parameter24:	0	
Parameter25:	0	Parameter26:	0	
Parameter27:	0	Parameter28:	0	
Parameter29:	0	Parameter30:	0	
☐ Single_Reset				

Figure 5: Automation tool's Randomization Framework GUI

## IV. Result Analysis

Randomization framework has been successfully implemented and randomized parameters are being selected between the range specified by the user, and are passed from automation tool's GUI to the test execution. See the appendix section for results in more detail.

## V. Conclusions

Randomization framework solved the problem of biasing in test execution for validation of SoC. Instead of user selecting any parameter value, it is required to provide the range within which values are required to be generated as per the data specifications. This is better than the standalone framework as it offers more coverage features in validation.

Thus, generic randomization framework for DUT would help to stress SoC components and gives user run control through automation software setup. It also helps to extract coverage and debug information from time to time through execution of test.

## VI. Future Scope

The work has a potential to be extended by adding some more features, and its application on other areas as follows:

- [1] Cases of interdependent parameters and scenario randomization where different test execution scenarios can be randomized.
- [2] Application of the randomization framework in areas of communication between two systems using different randomized communication protocol.
- [3] Application of Randomization Framework for Memory testing-where different memory blocks are considered randomly for reading/writing to test different memory accesses.

#### Acknowledgements

Thanks to Infineon Technologies India Pvt. Ltd., Bangalore for providing an opportunity to carry out this work. Thanks to Sadashivaiah Shivaprasad for his overall management for carrying out this project work. Thanks to Pammi Sesha (IFIN ATV PSV) for his support. Thanks to Mr. Rahul Thati (IFIN ATV PSV) for validation of the framework using SENT test. Our special Thanks to Ramasamy Subramanian for his valuable inputs towards the project work.

## References

 D.Ghosh, R.Subramaniam, V.Murthy, "A Randomized Methodology for Post-Silicon Validation of CAN and other Communication Modules", IN: Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on 22-25 Aug. 2013.

#### Appendix

In the "Fig. 6(a)-(b)" and "Fig. 7(a)-(b)" corresponding markings indicate-

[1] Downloading of the test executable to the DUT.

- [2] Seed number and [3] Seed type.
- [4] Selection of random parameters based on the seed type –incremental/random, and seed number.
- [5] Control word [entry pattern]- written from the test .
- [6] Writing the parameters to the shared memory locations.
- [7] Control word [resume pattern]-written from the automation process.
- [8] Prints from the test on the automation tool's result log file.
- [9] Control word [end pattern] –written from the test.
- [10] DUT reset once/reset on every iteration.

"Fig 6(a)" and "Fig 6(b)" show the result log obtained while downloading the test executable in flash using single board randomization framework with incremental seed type for three seed iterations and resetting the DUT on every test iteration, to ensure the recovery of the DUT from failure.

"Fig.7(a)" and "Fig.7(b)" shows result log generated in randomization framework for single board randomization framework with random seed type for three seed iterations without resetting the DUT more than once, to ensure a CPU state buildup. Random seed type is different from the incremental seed type, as the seed numbers generated are random in nature depending on their previous seed number.

		5		
		tmentool	00000	Test resume pattern = 0x1 found
		tmentool	00000	**************************************
		tmentool	00000	Test execution begins
	userdata/PTE Tools/Jazz/TC27xB-Aurix/tests/flashdownload_two_board.tfl e: (UTC iso8601) 2015-01-27 11:41:17		00000	Normal Anna Carlos - Carlos - Francisco
00001 Time	e: (UTC) 11:41:17 Tue 27 Jan 2015 - (local time) 17: 1:17 Tue 27 Jan 2015	tmentool	00000	ReadTestCaseMessages(50) ####################################
tmentool 00000 perl tmentool 00000 Comm	1.exe -X M:/dassubin.mcver.tc27xc.verification/mcver valix/automationL/sc mand for downloading the file: C:/Users/DasSubin/Desttop/sram_sent5.hex	tmentool	00000	######### Data read from memory ####################################
tmentool 00000 Writ	ting code to flash	tmentool	00000	########## Frane and CRC calculations DONE ####################################
	Halt mode	tmentool	00000	######################################
tmentool 00000 Swit tmentool 00000	tch to run Mode Total Parameters Entered : 10 1	tmentool	00000	########## Checking for 500 frame success interrupt ####################################
		tmentool	00000	########## Disabling ATOM channel, counter and clearing all compare and shadow :
tmentool 00000 ****	Parameters_Entered	tmentool	00000	########## Received data in the SENT RD register is 9185c4 ####################################
		tmentool	00000	########### Test Fass ##################################
***RANDOMIZATION FRAMEWORK tmemtool 00000	R	tmentool	00000	######### Test case Ends ####################################
_thencost 00000		tmentool	00000	Exiting_due_to_timecut.
Seed	d Number:12 3	tmentool	00000	************Result Extraction Done************************************
tmentool 00000 Para	ameter number 1 ::: Incremental PARAMETER SELECTED:::0x4	tmentool	00000	Test execution ends
tmentool 00000 Para	ameter_number_1 ::: Incremental PARAMETER SLLCTED:::0x6_ ameter_number_2 ::: Incremental FARAMETER_SLLCTED:::0x6_ ameter_number_3 ::: Incremental FARAMETER_SLLCTED:::0x8_	· ·		
tmentool 00000 Para tmentool 00000 Para	ameter number 3 ::: Incremental PARAMEIER SELECTED:::0x3 ameter number_4 ::: Incremental PARAMETER_SELECTED:::0x9	tmentool	00000	End_pattern_=0x10_found_
tmentool 00000 Para	ameter number 5 ::: Incremental PARAMETER SELECTED:::0x6	tmentool	00000	**************************************
tmentool 00000 Para tmentool 00000 Para	ameter number 6 ::: Incremental PARAMETER SELECTED:::0x8 > 4	÷		TestCase Passed
tmentool 00000 Para	ameter_number_7_:::_Incremental_PARAMETER_SELECTED:::0x4d ameter_number_8_:::_Incremental_PARAMETER_SELECTED:::0x1	HAAAD AND CMITZ	ATTON PRAN	EWORK*****
tmentool 00000 Para tmentool 00000 Para	ameter number 9 ::: Incremental PARAMETER SELECTED:::0x1	tmentool	00000	Emiliar
tmentool 00000 Rand	ameter_number_10_::: Incremental PARAMETER SELECTED:::0x1_ domization_FrameWork_Hard_Reset:_Connection_Established			0 000 00
tmentool 00000 ****	Testcase_Downloded_and_,Hard_Reset_Done			Seed Number:14
tmentool 00000 Entr	ry_pattern=0x1100110_found	tmentool	00000	Farameter number 1 ::: Incremental FARAMETER SELECTED:::0x5
tmentool 00000 ****		tmentool	00000	Parameter number 2 ::: Incremental PARAMETER SELECTED:::0x2
tmentool 00000 ****	TestCase Waiting	tmentool	00000	Parameter number 3 ::: Incremental PARAMETER SELECTED:::Oxf
tmentool 00000 writ	ting_at_address::0x7000f604,0x4	tmentool	00000	Parameter number 4 ::: Incremental PARAMETER SELECTED:::0x4 Parameter number 5 ::: Incremental PARAMETER SELECTED:::0x9
tmentool 00000 writ tmentool 00000 writ	ting_at_address::0x7000f608,0x6 ting_at_address::0x7000f60c,0x3	tmentool	00000	Parameter number 6 ::: Incremental PARAMETER SELECTED::: 0xe
tmentcol 00000 writ	ting_at_address::0x7000f610,0x9	tmentool tmentool	00000	Parameter number 7 ::: Incremental PARAMETER SELECTED:::0x2a Parameter number 8 ::: Incremental PARAMETER SELECTED:::0x0_
tmentool 00000 writ tmentool 00000 writ	ting_at_address::0x7000f614,0x6	tmentool	00000	Parameter number 9 ::: Incremental PARAMETER SELECTED:::0x1
tmestool 00000 writ	ting at address::0x7000f61c,0x4d	tmentool	00000	Parameter number 10 ::: Incremental PARAMETER SELECTED:::0x1
tmentool 00000 writ tmentool 00000 writ	ting_at_address::0x7000f620,0x1 ting_at_address::0x7000f624,0x1	tmemtool	00000	Randomization Framework Hard Reset: Connection Established
tmentool 00000 writ	ting at address::0x7000f628,0x1	tmentool	00000	**************************************
tmentool 00000 ****	Test_configuration_Done	tmentool	00000	Entry pattern=0x1100110 found
tmentool 00000 Test	t resume pattern = 0x1 found			
34)	7	tmentool	00000	**************************************
		tmentool	00000	writing at_address::0x7000f604,0x5
187 (F	· · · · · ·	tmentool	00000	writing_at_address::Cw7000f608,Ow2
	**************************************	tmentool	00000	writing at_address::0x7000f60c,0xf writing at_address::0x7000f610,0x4
_tmentool 00000T	Test_execution_begins		*****	natury of descent of second second
tmentool 00000 ReadT	TestCaseMessages (50)			2.22
	####### Test case begin ####################################	tmentool	00000	writing_at_address::0x7000f610,0x4
	####### Data read from memory ####################################	tmentool	00000	writing_at_address::0x7000f614,0x9
	####### Frame and CRC calculations DONE ####################################	tmemtool	00000	writing at address::0x7000f618,0xe
	ATTACK OTHER CONTRACTOR OF THE ATTACK	tmentool	00000	writing_at_address::0x7000f61c,0x2a
tmentool 00000 #####	###### Checking for 500 frame success interrupt ####################################	tmentool tmentool	00000	writing at address::0x7000f620,0x0 writing at address::0x7000f624,0x1
tmentool 00000 ##### tmentool 00000 #####	####### Disabling ATOW channel, counter and clearing fil compareow	tmentool	00000	writing at address::0x7000f628,0x1
tmentool 00000 #####	###### Test Pass ##################################	tmentool	00000	**************************************
tmentcol 00000 #####	####### End pattern return to memory ####################################			
	####### Test_case_Ends_####################################	tmentool	00000	Test resume pattern = 0x1 found
tmentool 00000 Exiti tmentool 00000 *****	ing due to timeout.	(d) -		
_usescor occo	Nesure Exclusione	tmemtool	00000	*******************************Resume Test Successful*****************************
tmentcol 00000 1	Test execution ends	tmentool	00000	Test execution begins
terretari 00000 - 1	pattern =0x10 found 9	11		
_tmemtool 00000 End_p	pattern -vxiv round	tmentool	00000	ReadTestCaseMessages (50)
tmentool 00000 *****	End pattern Successful	tmentool	00000	########## Test case begin ####################################
	Case Passed	tmentool	00000	######### Data read from memory ####################################
****DISTONTOLOTON DELLOS		tmentool	00000	########## Frame and CRC calculations DONE ####################################
tnentcol 00000		tmentool	00000	########### Data loaded into FIED ####################################
-	Verbau12 2	tmentool	00000	######################################
Seed	Number:13 Z	tmentool	00000	######################################
TRANSCO' AAAAA	manay mushaw 1 Tanyawasal Rinsuppon aproven	tmentool	00000	############# Received data in the SENT RD register is e94f25 ####################################
	meter number 1 ::: Incremental PARAMETER SELECTED:::0x4_ meter number 2 ::: Incremental PARAMETER SELECTED:::0xc_	tmentool	00000	
	meter_number_3_::: Incremental PARAMETER_SELECTED:::0x5_	tmentool	00000	
tmentool 00000 Param	meter_number 4 ::: Incremental PARAMETER_SELECTED:::0x8	tmentool	00000	
tmentool 00000 Paran	meter number 5 ::: Incremental PARAMETER SELECTED:::0x1	tmentool	00000	Exiting_due_to_timeout.
tmentcol 00000 Paran tmentcol 00000 Paran	meter number 6 ::: Incremental PARAMETER SELECTED:::0x9 meter number 7 ::: Incremental PARAMETER SELECTED:::0x4	tmentool	00000	*********Result Extraction Done************************************
tmentool 00000 Paran	meter number 8 ::: Incremental PARAMETER SELECTED:::0x0			
	meter_number_9_:::_Incremental_PARAMETER_SELECTED:::0x1_	tmentool	00000	Test_execution_ends
	meter number 10 ::: Incremental PARAMETER SELECTEDT::0x1 10			
tmentool 00000 Rando	Introview rights Baid Moter Completion 25140113060	tmentool	00000	End_pattern_=0x10_found_
	********Testcase_Downloded_and_,Hard_Reset_Done_**********************************		00000	sussessment and anteres Conservation
	y_pattern=0x1100110_found	tmentool	00000	TestCase Passed
tmemtool 00000 *****	Test Entry Successful			Everyonery England
	TestCase Waiting	tmentool	00000	P -01
[tmentool 00000 writi	ing_at_address::0x7000f604,0x4			P 001 001
	ing at address::0x7000f608,0xc			489950 ms (00:08:9.950)
	ing at_address::0x7000f60c,0x5 ing at_address::0x7000f610,0x8		0.000	
	ing at address::0x7000f614,0x1			
tmentool 00000 writi	ing_at_address::0x7000f618,0x9			
	ing_at_address::0x7000f61c,0x4	22		
	ing at address::0x7000f620,0x0 ing at address::0x7000f624,0x1			
tmentool 00000 writi	ing at address::0x7000f628,0x1	г.	<i>co</i> :	
	******Test configuration Done************************************	Figure	e.o(b)	Result log obtained for randomization

Figure.6 (a) Result log obtained for randomization framework "SENT Normal Frame Test" using Incremental seed type and "single reset" unset

## A Generic Randomization Framework Architecture for Test Execution in Automated Testing Of Soc

				ovnload	00000	########### Data read from memory ####################################
String String		00001	Time:_(UTC_iso8601)_2015-03-24_06:02:10 Time:_(UTC)_06:02:10_Tue_24_Mar_2015(local_time)_11:32:10_Tue_24_M	ownload	00000	
wnload		00000	perl.exeX_M:/dassubin.mover.tc27xc.verification/mover_valixgen2/aut	ownload		########## Data loaded into FIFO ##################################
mload		00000	Command_for_downloading_the_file:_C:/Users/DasSubin/Desktop/sraml.hex	ownload		########## SENT initialization is DONE ####################################
mload			Writing code_to_flash			
mload		00000		ownload		########### Checking for 500 frame success interrupt ####################################
mload		00000		ownload	00000	
mload		00000	Total_Parameters_Entered_:_10	ownload	00000	########## Received data in the SENT RD register is 46547b ####################################
011090		00000	iotal_fatameters_choered_:_10	ownload	00000	
aland.		00000	**************************************	ovnload	00000	
nload		00000	Palaneters_Entered	ownload	00000	**********Result_Extraction_Done************************************
	*****RANDOMIZA		MEMORIX	cwnload	00000	Test_execution_ends
load		00000		ownload	00000	End pattern +0x10 found
		2	3000 #400001112	10020101/		
nload		00000	Parameter number 1 ::: Random PARAMETER SELECTED::: 0x4	ownload	00000	TestCase Passed
mload		00000				TestLase rassed
mload		00000		20101030251000044		
mload		00000				MENTER.
nload			Parameter_number_5_::: Random_PARAMETER_SELECTED:::0x6_	cvnload	00000	
mload mload		00000				
nload		00000				Seed Number:58776
nload		00000				
nload		00000	Parameter_number_9_:::_Random_PARAMETER_SELECTED:::0x1 Parameter_number_10_:::_Random_PARAMETER_SELECTED:::0x1	ownload	00000	Parameter number 1 ::: Random PARAMETER SELECTED:::Oxa
nload		00000	Randomization Framework Hard Reset: Connection Established	ownload		Parameter number 2 ::: Random PARAMETER SELECTED:::Oxf
mogu		00000	Mandomization ridsework hard headt, connection catabilaned	cynload	00000	Parameter number 3 ::: Random PARAMETER SELECTED:::0x4
nload		00000	*************Testcase Downloded and ,Hard Reset Done *********************************	cynload		Parameter number 4 ::: Random PARAMETER SELECTED:::0x2
nload		00000	Entry pattern=0x1100110 found			
			mart handra-out of toma	ownload		Parameter_number_5_::: Random_PARAMETER_SELECTED:::Oxb_
nland		00000	ANALY ANALY TANK STANDART I ANALY ANALY	ownload	00000	Parameter_number_6_:::_Random_PARAMETER_SELECTED:::Oxc_
nload nload		00000	**************************************	ownload	00000	Parameter_number_7_:::=Random_PARAMETER_SELECTED:::0x28_
nload		00000	writing at address::0x7000f604,0x4	ownload	00000	Parameter_number_8_:::_Random_PARAMETER_SELECTED:::0x0_
nload		00000		ownload	00000	Parameter number 9 :::: Random PARAMETER SELECTED::::0x1
nload		00000		cynload	00010	Parameter number 10 ::: Random PARAMETER SELECTED:::0x1
nload		00000	writing at address::0x7000f610,0x9	ownload	00000	**************************************
nload			writing_at_address::0x7000f614,0x6	ownload		writing at address::0x7000f604,0xa
		00000				
nload nload		00000	writing_at_address::0x7000f61c,0x4d	ownload		writing_at_address::0x7000f608.0xf
nload		00000	writing at address::0x7000f620,0x1	ownload		writing_at_address::0x7000f60c,0x4
mload		00000	writing_at_address::0x7000f624,0x1	ownload	00000	writing_at_address::0x7000f610,0x2
		00000		ownload	00000	writing at address::0x7000f614,0xb
mload			writing_at_address::0x7000f628,0x1	ownload	00000	writing at_address::0x7000f618,0xc
mload		00000	**************************************	ownload		writing at address::0x7000f61c,0x28
al and		00000	Test_resume_pattern = 0x1 found	ownload		writing at address::Dx7000f620,0x0
mload		000000	rest_resume_pattern_=_ox1_round			
mload		00000	**************************************	cwnload		writing_at_address::0x7000f624,0x1
nload		00000	Test_execution_begins	cwnload cwnload	00000 00000	<pre>writing at_address::0x7000f629,0x1 ***********Test configuration Done************************************</pre>
nload		00000	ReadTestCaseMessages (50)	ownload	00000	Test resume pattern = 0x1 found
						State and the second s second second se second second sec second second sec
mload		00000	########## Data_read_from_memory_####################################	cwnload	00000	**************************************
mload			########## Frame and CRC calculations DONE ####################################	cwnload	00000	Test execution begins
mload		00000	########## Data loaded into FIFO ##################################			
mload		00000	########## SENT_initialization_is_DONE_####################################	ownload	00000	ReadTestCaseMessages (50)
wnload		00000	######### Checking_for_500_frame_success_interrupt_####################################	ownload	00000	########### Data read from memory ####################################
wnload		00000	########## Disabling_ATOM_channel,_counter_and_clearing_all_compare_and_sha	ownload	00000	
wnload wnload		00000	######################################	ownload	00000	
				ownload	00000	
vnload vnload			Exiting_due_to_timeout.	546 A 200 A 20		########## Checking_for_500_frame_success_interrupt_####################################
KIIIOBU		00000	Reading Protection Done	ownload		
wnload		00000	Test execution ends	ovnload		########## Disabling ATOM channel, counter and clearing all compare and
xuroga			1235_CAECUCICH_CHU3	ownload		######################################
wnload		00000	End_pattern_=0x10_found_	ownload	00000	########## Test_Pass_##################################
				cwnload	00000	Exiting due to timeout.
wnload		00000	***************End pattern Successful******************************	cwnload	00000	**************************************
			TestCase Passed	ownload	000	
	******RANDOMIZAT		MER.	ownload	00	000 End pattern =0x10 found
wnload		00000	Seed Number: 83178 2	ownload		000 **********************************
						TestCase Passed
wnload		00000	Parameter_number_1_:::_Random_PARAMETER_SELECTED:::Oxb_	ownload		000 P -01
wnload			Parameter_number 2 ::: Random_PARAMETER_SELECTED:::0x7	0801080	00	www.ewe
wnload			Parameter_number_3_:::_Random_PARAMETER_SELECTED:::0x4_	1		
wnload	Т		Parameter_number_4_::: Random_PARAMETER_SELECTED:::0x5_			
	J.		Parameter_number_5_:::_Random_PARAMETER_SELECTED:::0xb_ Parameter_number_6_:::_Random_PARAMETER_SELECTED:::0x4_	1		
mload		00000	Parameter number 7 ::: Random PigiMFTFR SFIFCTFD:::0x4	1		
mload mload		00000	Parameter_number_7::::Random_PARAMETER_SELECTED:::0x54_ Parameter_number_0::::Random_PARAMETER_SELECTED:::0x1_	1		
knload knload knload		00000	Parameter_number_0_:::_Random_PARAMETER_SELECTED:::0x1_			
wnload wnload wnload wnload			Parameter_number_10_::: Random_PARAMETER_SELECTED:::0x1_	55		
wnload wnload wnload wnload wnload		00000				
wnload wnload wnload wnload wnload		00000	TestCase_ Faiting			
wnload wnload wnload wnload wnload wnload wnload		00000	<pre>writing_at_address:0x7000f604,0xb</pre>	Figure	7(h) D	exult log obtained for
wnload wnload wnload wnload wnload wnload wnload wnload		00000 00000 00000	<pre>writing_at_sddress:0x7002f604,0xb writing_at_sddress:0x7002f604,0xb</pre>	Figure.	7(b) Re	esult log obtained for
wnload wnload wnload wnload wnload wnload wnload wnload		00000 00000 00000 00000	TestCase Matting writing at address: uNr000664, 0mb writing at address: UNr000664, 0mT writing at address: UNr000660, 0mT	0		e e
wnload wnload wnload wnload wnload wnload wnload wnload wnload		00000 00000 00000 00000 00000	"TeirDag Bailog viiing at address:0x7000606,0x6 viiing at address:0x7000660,0x7 writing at address:0x7000660,0x4 writing at address:0x7000660,0x4	0		e e
wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload		00000 00000 00000 00000 00000 00000	vriting ar_iddress:0x70006604,0x5 vriting ar_iddress:0x70006604,0x7 vriting ar_iddress:0x70006600,0x7 vriting ar_iddress:0x70006600,0x4 vriting ar_iddress:0x70006610,0x5 vriting ar_iddress:0x70006610,0x5	random	nization	framework "SENT Normal Frame
wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload		00000 00000 00000 00000 00000 00000 0000	<pre>viiing asddress:lor100664, ubb viiing _stddress:lor1006640, ubb viiing _stddress:lor1006640, ubf viiing _stddress:lor1006640, ubf viiing _stddress:lor1006640, ubf viiing _stddress:lor1006641, ubf viiing _st</pre>	random	nization	e e
wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload		00000 00000 00000 00000 00000 00000 0000	<pre>vriing ar_ddress:toX10006604/sbb wriing ar_ddress:toX10006604,sbb wriing ar_ddress:toX1000660,bcf wriing ar_ddress:toX1000660,bcf wriing ar_ddress:toX10006610,bd5 wriing ar_ddress:toX10006614,bdb wriing ar_ddress:toX10006614,bd4</pre>	random Test" u	nization sing rar	framework "SENT Normal Frame
wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload		00000 00000 00000 00000 00000 00000 0000	<pre>viiing as_ddiese:loT000664, bb viiing as_ddiese:loT0006640, bb viiing as_ddiese:loT0006640, bd viiing as_ddiese:loT0006640, bd viiing as_ddiese:loT0006610, bd viiing as_ddiese:loT0006610, bd viiing as_ddiese:loT0006610, bd viiing as_ddiese:loT0006610, bd viiing as_ddiese:loT0006610, bd viiing as_ddiese:loT0006610, bd</pre>	random Test" u	nization sing rar	framework "SENT Normal Frame
wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload		00000 00000 00000 00000 00000 00000 0000	<pre>viting st_detess:10/1006/06,0xb viting st_detess:10/1006/06,0xf viting st_detess:10/1006/06,0xf viting st_detess:10/1006/06,0xf viting st_detess:10/1006/06,0x5 viting st_detess:10/1006/06,0x5 viting st_detess:10/1006/06,0x4 viting st_detess:10/1006/06,0x1 viting st_detess:10/1006/06,0x1</pre>	random	nization sing rar	framework "SENT Normal Frame
wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload wnload		00000 00000 00000 00000 00000 00000 0000	<pre>versing as_ddress:lnllllllllllllllllllllllllllllllllll</pre>	random Test" u	nization sing rar	framework "SENT Normal Frame
wnload wmload wmload wmload wmload wmload wmload wmload wmload wmload wmload wmload wmload wmload wmload wmload wmload		00000 00000 00000 00000 00000 00000 0000	<pre>vriing st_address:loT000664, 0ab vriing st_address:loT0006640, 0ab vriing st_address:loT0006640, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006420, 0af vriing st_address:loT0006420, 0af vriing st_address:loT0006420, 0af vriing st_address:loT0006420, 0af vriing st_address:loT0006420, 0af</pre>	random Test" u	nization sing rar	framework "SENT Normal Frame
veniced veniced		00000 00000 00000 00000 00000 00000 0000	<pre>versing as_ddress:lnllllllllllllllllllllllllllllllllll</pre>	random Test" u	nization sing rar	framework "SENT Normal Frame
period pe		00000 00000 00000 00000 00000 00000 0000	<pre>vriing st_address:loT000664, 0ab vriing st_address:loT0006640, 0ab vriing st_address:loT0006640, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006410, 0af vriing st_address:loT0006420, 0af vriing st_address:loT0006420, 0af vriing st_address:loT0006420, 0af vriing st_address:loT0006420, 0af vriing st_address:loT0006420, 0af</pre>	random Test" u	nization sing rar	framework "SENT Normal Frame
wnload wnload		00000 00000 00000 00000 00000 00000 0000	<pre>vriting st_ddress:th7005640, thb vriting_st_ddress:th7005640, thb vriting_st_ddress:th7005640, thd vriting_st_ddress:th71006610, thd vriting_st_ddress:th7100610, thd vriting_st_ddress:th7100610, thd vriting_st_ddress:th7100610, thd vriting_st_ddress:th7100610, thd vriting_</pre>	random Test" u	nization sing rar	framework "SENT Normal Frame

Figure.7(a) Result log obtained for randomization framework "SENT Normal Frame Test" using random seed type and "single reset" flag set.