

**A REPORT**

**ON**

**Test Program Development for Post Silicon Validation of Devices using  
Automatic Test Equipment (ATE)**

**BY**

**Name of the Student:**  
A Vipula

**ID No:**  
2021HT80530

**AT**

**Infineon Technologies, Bangalore**



**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**  
**(November 2023)**

**A REPORT**

**ON**

**Test Program Development for Post Silicon Validation of Devices using  
Automatic Test Equipment (ATE)**

**BY**

**Name of the Student:** A Vipula      **ID No:** 2021HT80530      **Discipline** Microelectronics

Prepared in partial fulfilment of the  
WILP Dissertation/Project/Project Work Course

**AT**

**Infineon Technologies, Bangalore**



**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**(November 2023)**

## **Acknowledgments**

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders, and friends. Several personalities, in their own capacities have helped us in carrying out this project work. I would like to take this opportunity to thank them all.

I express my profound gratitude to Dean of the Institution, for providing an opportunity to pursue my M. Tech Degree at BITS PILANI through Work Integrated Learning Program.

My sincere thanks to **Anudeep Singh**-Senior Staff Engineer and **Manibharath N**-Principal Engineer at **Infineon Technologies Bangalore** for their wholehearted support, suggestions, unmatched services and invaluable advice throughout my project work and I am profoundly grateful to them for rendering guidance in completion and reviewing of this report.

I would also like to thank the Advantest Applications team for their invaluable assistance and support during the project.

Last but not the least I would like to thank my family and friends, for their unwavering support and encouragement throughout the journey.

**A Vipula**

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**  
**(RAJASTHAN)**  
**WILP Division**

**Organization:** Infineon Technologies

**Location:** Bangalore

**Duration:** 2 years

**Date of Start:** January 2022

**Date of Submission:** 13<sup>th</sup> November 2023

**Title of the Project:** Test Program Development for Post Silicon Validation of Devices using Automatic Test Equipment (ATE)

**ID No:** 2021HT80530

**Name of the student:** A Vipula

	<b>Supervisor</b>	<b>Additional Examiner</b>
<b>Name:</b>	Anudeep Singh	Manibharath N
<b>Designation:</b>	Senior Staff Engineer	Principal Engineer

**Name of the Faculty mentor:** Anudeep Singh

**Key Words:** Automatic Test Equipment (ATE), Device under Test (DUT), Advantest Verigy 93K, Validation, Pattern, Mode Transition (MT), Power on Reset (POR)

**Project Areas:** Testability for VLSI

## **Abstract:**

Semiconductor Testing plays a crucial role in the electronics industries as it leverages the quality and reliability of design. Traditional test methods require time and effort which is a cumbersome process hence, the need states the use of the AUTOMATIC TEST EQUIPMENT in short ATE for post silicon validation of the Device Under Test (DUT's). An ATE is capable of automatically testing and diagnosing or detecting faults in sophisticated electronic packaged parts or on wafer testing, including System on Chips, Digital Circuits, Analog Circuits, RF Circuits and integrated circuits as well. Testing not only makes it possible to select the good parts from the lot but also the data collected from the test results help in improving the design, the technology involved, the fabrication process and various other yield parameters.

The dissertation is based on developing a complete Test Program using Automatic Test Equipment (Advantest V93K Tester) for Post Silicon Validation of Devices which mainly focuses on verifying the Mode Transition and Power on Reset behavior of the Device. Mode transition validation is done to verify the functionality of the chip when the chip transitions between different operating modes. The different operating modes are Active Mode, Sleep Mode, Reset Mode and Deepsleep Mode, where each of these modes characterizes the power consumption of the chip. The Test cases for Mode Transitions that will be validated are Next State Tests and Interruptible Transitions where Devices operating modes are switched by providing/triggering various types of inputs through ATE (which will be referred to as Interrupts also). Power on Reset involved various types of ramps generations and inputting to device supplies through ATE and verify the Device POR architecture. In both the Mode Transition and Power on Reset Validation, it involves a custom Firmware programmed to Device based on its Register configurations. Upon the analyzing the Device's response; then qualify the Device behavior's if it meets the specifications guidelines.



**Signature of Student**  
**Date: - 13<sup>th</sup> November 2023**



**Signature of your Supervisor**  
**Date: -13<sup>th</sup> November 2023**

**Thank You Team**  
**EVM**  
**BITS Pilani WILP**

## Table of Contents

1. Introduction	1
2. The fundamentals of Semiconductor Testing	2
3. Basics of Pin Electronics Architecture	4
4. Power on Reset (Por) Trim and Screen Development and Verification at Wafer Sort	5
4.1 Introduction	5
4.2 Test Case1: POR Circuitry Comparator Offset Trimming at Wafer Sort	5
4.2.1 Test Program Development	5
4.2.2 Validated Results	8
4.3 Test Case2: POR Circuitry Hysteresis Screening at Wafer Sort	9
4.3.1 Test Program Development	9
4.3.2 Validated Results	11
4.3.2.1 Lower Trip Point Results	11
4.3.2.2 Upper Trip Point Results	15
5. Firmware Programming Through Automatic Test Equipment (ATE)	18
5.1 Introduction	18
5.2 Test Program Development	18
5.3 Validated Results	19
6. Power on Reset Extensive Functional Characterization Through Automatic Test Equipment (ATE) On Packaged Units	21
5.1 Introduction	21
5.2 Test Program Development	21
5.3 Validated Results	23
7. Mode Transition (MT) Functional Verification on Packaged Units	27
5.1 Introduction	27
5.2 Test Program Development	27
5.3 Validated Results	29
8. Conclusion	38
9. Appendices	39
10. References	40
11. Glossary	41

## **1.INTRODUCTION**

The complexity of IC is progressively ever increasing, the complexity of testing them has also increased significantly. The largest portion of device manufacturing cost is testing. VLSI devices require large sets of voltages, current and timing tests. Millions of functional and parametric steps may be required to ensure complete functionality of semiconductor device. To perform such complex testing, an Automated Test Equipment (ATE) is used. ATE is a collection of high-performance computer-controlled test instruments. A test system is the result of this merging of many test instruments or equipment's with a computer. The computer controls the test hardware by executing a set of instructions called the test program. Consistent test results are obtained through test system which can be relied upon and repeated easily. To keep the results, correct and consistent, test systems are periodically undergo calibration to verify the accuracy of forcing and measuring instruments/tester channels. The overview of the testing methodology involved right from the inception of the project till completion has been enumerated with highlights on features of automatic testing, the design considerations and other automatic tools used in the process for easier analysis of the results/ log obtained have been put forth.

The process of bringing up an integrated circuit (IC) from design to testing is a complex and time-consuming task that involves several stages, such as:

- ✓ Design for Test (DFT): Techniques from design that guarantee the chip is created in a way that it can be efficiently evaluated within internal to itself or with minimal inputs from ATE.
- ✓ Bench: Examining the device in a controlled lab to confirm that the design is accurate according to datasheet specifications and meets the necessary standards
- ✓ Characterization: The production/process variations and refining performance, levels, timing, temperatures for quality large scale production release.
- ✓ Development, Debug and Bringup and Production Release: The creation of test program for verifying the Device at Wafer Sort and Backend and bringing up of all test cases pertaining to device functionality as specified in data sheets. Performing data analysis on test results for Standard Deviations, Process Parameter's like Cpk, Mean Shifts, Gauge Repeatability, Distribution and Quantile plots so as to sample the devices to customers with proper sigma quality.

This dissertation is based on verifying the device's **POWER on RESET architecture and MODE TRANSITION functionality** for Wafer Sort and Packaged program.

## 2.THE FUNDAMENTALS OF SEMICONDUCTOR TESTING

Wafer, Die and Packages

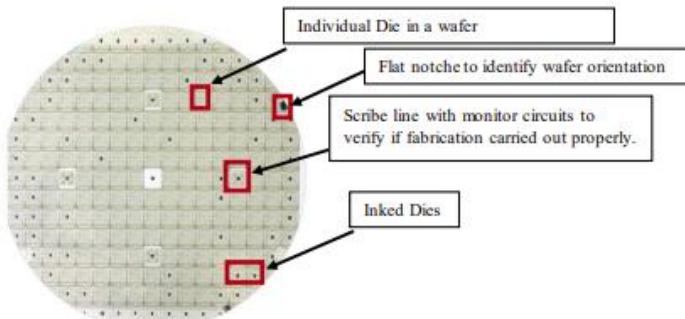


Fig2a: Tested wafer

- ✓ Semiconductor circuits are initially fabricated in the form of wafer-**Fig2a**. A wafer is a circular slice of silicon upon which individual circuits are built upon. These individual circuits are referred to as dies. Each die is isolated from and is completely independent of others.
- ✓ Testing of wafer is referred as wafer probing or die sorting. Each die must be thoroughly tested to ensure it meets the device specifications thoroughly. At some cases if die failure occurs they are inked so as to mark them as bad dies and process is call Inking.
- ✓ After probing is done dies are cut from wafer, this is referred to as wafer sawing. After sawing defective die are scrapped off and the proper ones are sent for packaging. Some example of packages is Dual Inline Package (DIP), Single Inline Package (SIP), Thin Small outline package (TSOP), Ball Grid Array (BGA), Quad Flat pack (QFP) etc.
- ✓ The basic components requisite for testing are probe card, performance board/load board, device handler, tester etc.

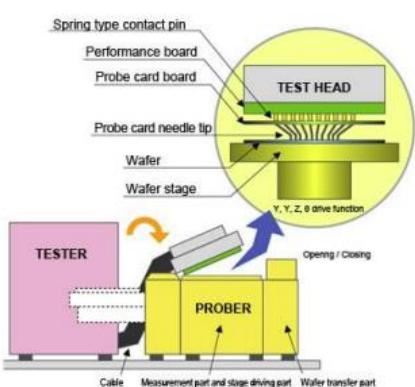
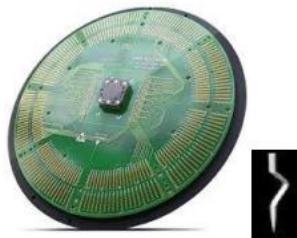


Fig2b: Model Tester System



Fig2c: Advantest V93K Tester



**Fig2d: ProbeCard**

- ✓ A Probe card is used to verify proper operation of a die within a wafer when a wafer level testing (probing) is carried out. A performance board or load is an interface circuit board that connects the instruments in the test system to the probe card. The load board and probe card work together and enable electrical signals to pass to and fro between the test system and the die.



**Fig2e: Device Handler docked into the Tester**

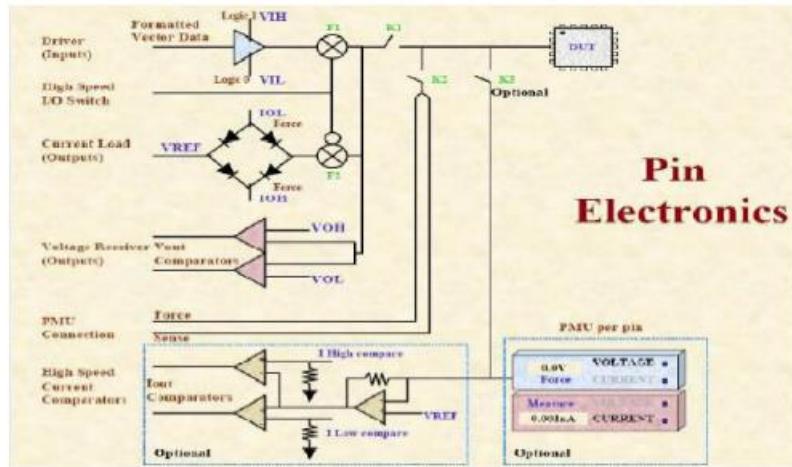
- ✓ The Device handler in a test system (send information to device handler) is crucial for bin assignment. A device handler usually has a minimum of three bins – an untested bin, a good device bin and a reject bin. Other bins such as Functional rejects bin, AC rejects bin and DC rejects bin are also useful. Dies which fail a test flow might prove useful in some applications, therefore, instead of rejecting all dies in the end, which leads to wastage of tester resources and time, it is better to sort the devices as per test flow failures. -**Fig2e**



**Fig2f: Temperature Forcing Unit**

- ✓ To check for packaged DUT under extremes temperatures to withstand harsh environmental conditions a temperature forcing unit is used similarly, pressure forcing units are deployed-**Fig2f**

### 3.BASICS OF PIN ELECTRONICS ARCHITECTURE



**Fig3a: Basic Architecture of Pin Electronics**

Pin Electronics System form the basic backbone of the tester architecture. Every ATE system might have different architecture but the fundamentals remain the same.

It provides the following capabilities:

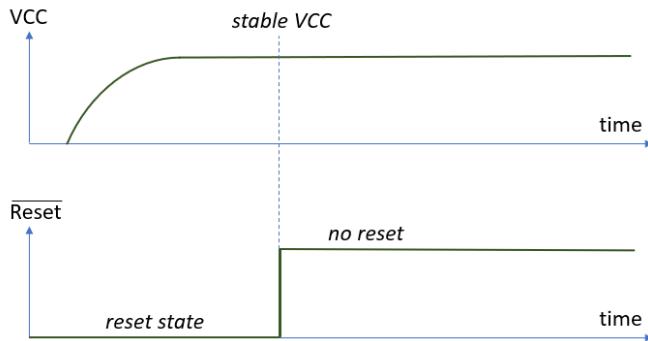
- ✓ Driver circuitry for input signals where VIL and VIH signals are applied to formatted data.
- ✓ I/O driving/switching circuitry to turn drivers and current loads on/off.
- ✓ Voltage comparator circuitry for detecting/comparing output levels where VOL and VOH levels are used.
- ✓ A connection point to the PMU (Precision Measurement unit) – force and measure voltage and current capabilities.
- ✓ Programmable Current loads- acts as loads for DUT for sinking or sourcing positive and negative currents from test system to and from DUT respectively.
- ✓ High speed current comparators are used for high leakage currents measurements carried out during both static and dynamic current testing.
- ✓ PMU per pin has capabilities of Force/Measure Currents and Voltage and clamp settings

**Note: All the test programming is done in JAVA which is Language for the ATE (Advantest V93K tester Smartest8 version).**

## **4.POWER ON RESET (POR) TRIM AND SCREEN DEVELOPMENT AND VERIFICATION AT WAFER SORT**

### **4.1 Introduction**

- ✓ POR circuitry is mainly built, such that the Devices are held in reset state until a stable voltage on the POWER supply/supplies is reached. This is mainly required so that the Device starts or boots up from a known state. -**Fig4a**

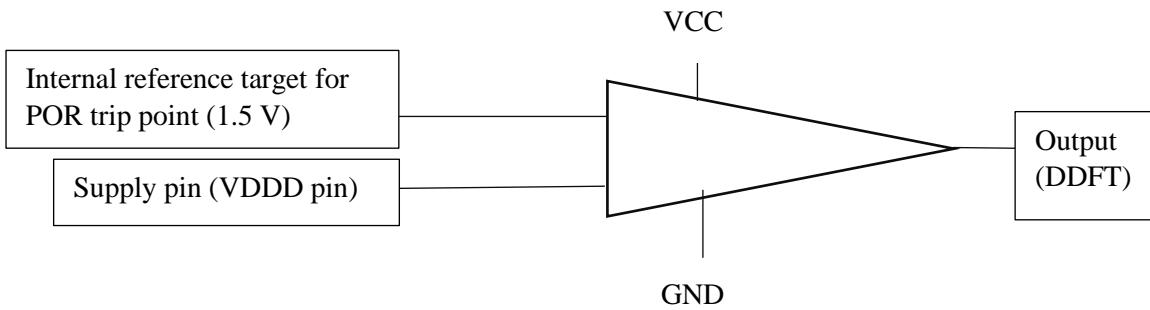


**Fig4.1a: Reference Example waveform for POR Circuitry.**

### **4.2 Test Case1: POR Circuitry Comparator Offset Trimming at Wafer Sort**

#### **4.2.1 Test Program Development**

- ✓ POR circuitry has a comparator which has to be trimmed to find out the comparator offset which is unique for each device owing to its process variations. In practical Op amp's have an offset which is known as a small differential voltage to be applied to the inputs of comparators to force the output to zero. This is the offset computed through this Test Case. Here the target trip point/POR Point is input as internal reference for one end of the comparator and another end of comparator is the supply pin for which the POR circuitry is built- **Fig4.2a**



**Fig4.2a: Comparator Sample Schematic**

✓ **Fig4.2b** shows the test suite which performs the comparator offset trimming. The algorithm is written in a file called Test Method: here it is **IP\_S40SRSSA\_BOD\_TRIM3**. Here the major parameter details given as input to the Test Method are as follows

- **InputPinlist**- it defines the device power supply pin for which the POR circuitry is built and the offset of that comparator of POR circuitry is to be found. Here it is “pl\_VDDD”.
- **OutputPinlist**-this is pin which is the output of the comparator routed through a GPIO pin any. Here it is named “pl\_DDFT0”.
- **TrimPattern**- this is a pattern which has the details for device Bandgaps, Vref and Iref trims needed for basic device functionality.
- **VohLevel**- this is the output voltage high level in Volt to be recognized as output 1. This is for IO pins.
- **VolLevel**- this is the output voltage low level in Volt to be recognized as output 0. This is for IO pins.
- **TargetTripPoint**- 1.5V it is target at which the the VDDD pin is set to for the POR circuitry to start working as per its functionality.
- **TrimSliceName**- “PWR\_TRIM\_BODOVP\_CTL\_HVPORBOD\_OFSTRIM” this is the 3bit wide register which has finer adjustments to the comparator offset- Refer **Fig4.2c**.
- **Settling Time**- this is parameter for tester when values are forced to settle upon which further measurements are made.
- **Force Current**- 0 means we are doing a high impedance voltage measurement

✓ Here the Test Method: **IP\_S40SRSSA\_BOD\_TRIM3**- take the input parameters as in **Fig4.2b**, here initially the patterns to put the device for POR trim functionality(Instruction File parameter) is executed, then the Target of 1.5 V is forced on the VDDD pin, then the Register

“PWR\_TRIM\_BODOVP\_CTL\_HVPORBOD\_OFSTRIM” is swept for the entire trim code set-  
Fig10, the point at which the pl\_DDFT0 pin goes high that is its VOH level is measured, then the offset is added/subtracted to the target 1.5V to get the device POR trip point. The final computed trim code is stored in the trim register.

```

suite S40SRSSA_HVPORBOD_OFSTRIM_temp calls tml.IP_S40SRSSA.IP_S40SRSSA_BOD_TRIM3
{
    TN = 797;
    HardBinOnFail = 7;
    ExecOn = #[ "SORT1", "SORT2", "CRI" ];
    baseSpecs = #[ setupRef(common.level.VCCD_MIN_VDDA_MIN2), setupRef(common.timings.STANDARD_TIME_250NS) ];
    InstructionFile = "InsFile/IP_S40SRSSA/FX3G2/S40SRSSA_HVPORBOD_SCREEN.ins";
    POW[YES] = {};
    InputPinlist = "pl_VDDD";
    OutputPinlist = "pl_DDFT0";
    TrimPattern = "InsFile/Support/Register_Load_MIX.ins";
    VohLevel = 1.0;
    VolLevel = 0.6;
    TargetTripPoint = 1.5;
    TrimSliceName = "PWR_TRIM_BODOVP_CTL_HVPORBOD_OFSTRIM";
    SettlingTime = 2;
    ForceCurrent = 0;
    rbm_mode = REGBYPASS;
    acquire_mode_pattern = DFT_ENABLE_NOREG_SWD_ACQ;
    trim_pattern = TRIM;
}

```



**Fig4.2b: Input parameters to the Test Method**

hvporbod_ofstrim_hv<2:0>		Adjust level [mV]
Bin	Hex	
000	0h	0
001	1h	+30
010	2h	+20
011	3h	+10
100	4h	0
101	5h	-10
110	6h	-20
111	7h	-30

Default: 3'b000

**Fig4.2c: Comparator Finer Adjustments for the offset.**

#### 4.2.2 Validated Results

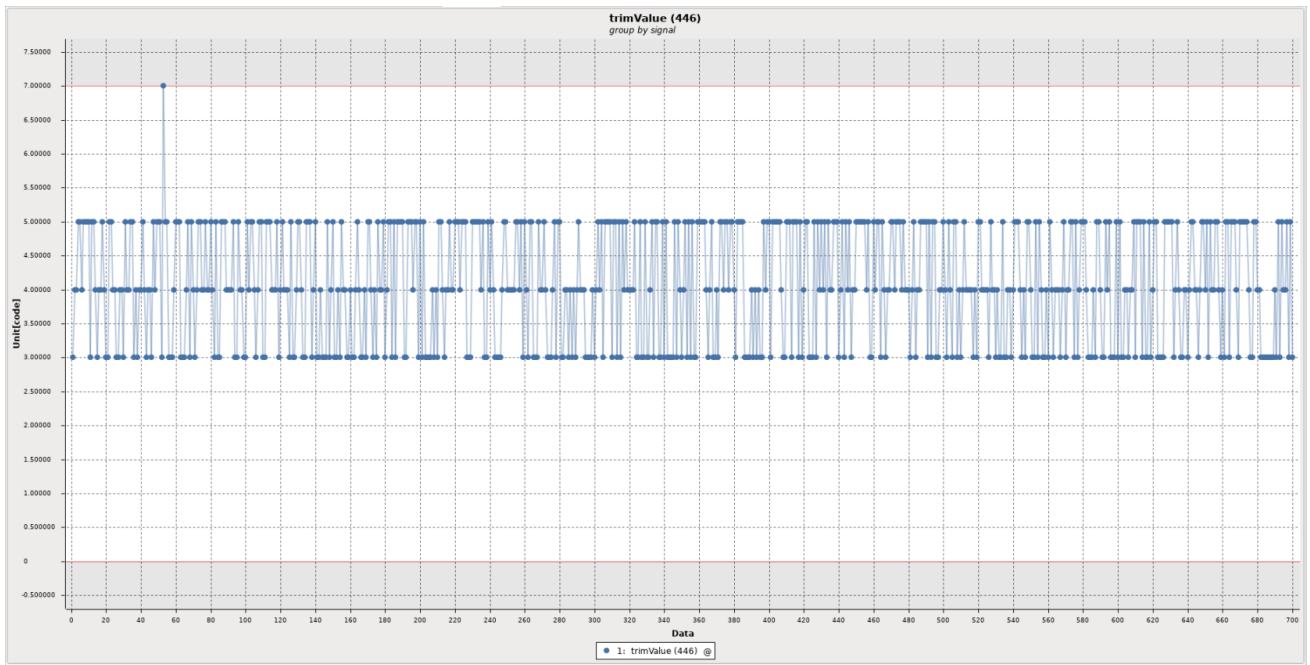
```

OpSeq_1.seq ✎
1 //This operating sequence file was generated with the following prefix: CRI
2 @sequence OpSeq_1 uses dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Spec_1
3 {
4     sequential _seq_1
5     [
6         //XRES Power Down
7         patternCall dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Pattern_1 ;
8
9         wait 1ms;
10        patternCall dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Pattern_2 ;
11
12        wait 21.000000 us;
13        patternCall dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Pattern_3 ;
14
15        wait 5.000000 us;
16        patternCall dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Pattern_4 ;
17        wait 13.000 ms;
18        patternCall dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Pattern_5 ;
19
20        wait 50.00000 us;
21
22        patternCall dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Pattern_6 ;
23        patternCall dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Pattern_7 ;
24
25        wait 1.000000 us;
26
27        patternCall dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Pattern_8 ;
28
29        patternCall dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Pattern_9 ;
30        transactSeqCall TransSeqDef_S40SRSS_DFT_ENABLE_NOREG_SWD_ACQ1;
31
32        wait 200.0000 us;
33
34        patternCall dsa_gen.CRI.Main.S40SRSSA_PORBOD_TRIM_FLOW.S40SRSSA_HVPORBOD_OFSTRIM_temp.Pattern_10 ;
35
36        transactSeqCall TransSeqDef_Register_Load_MIX_CALLS_Register_Load_MIX_SUB1;
37
38        transactSeqCall TransSeqDef_S40SRSSA_HVPORBOD_SCREEN1;
39        transactSeqCall TransSeqDef_S40SRSSA_HVPORBOD_SCREEN_CALLS_ADFTO_DRIVE_HIGH_SUB1;
40        wait 200.0000 us;
41    }
42 }
43 }
```

```

OpSeq_1.seq ✎ OpSeq_2.seq ✎ Spec_2.spec ✎
137
138@    setup digInOut pl_DDFTO
139{
140@    action iforceVmeas vMeas
141{
142    forceValue = 0 A;
143    waitTime = 2.000000000 ns;
144    averages = 1;
145    vclampLow = -2 V;
146    vclampHigh = 4 V;
147    irange = 10.00000 uA;
148    limits.low = 1 V;
149}
150}
151action vMeas;
```

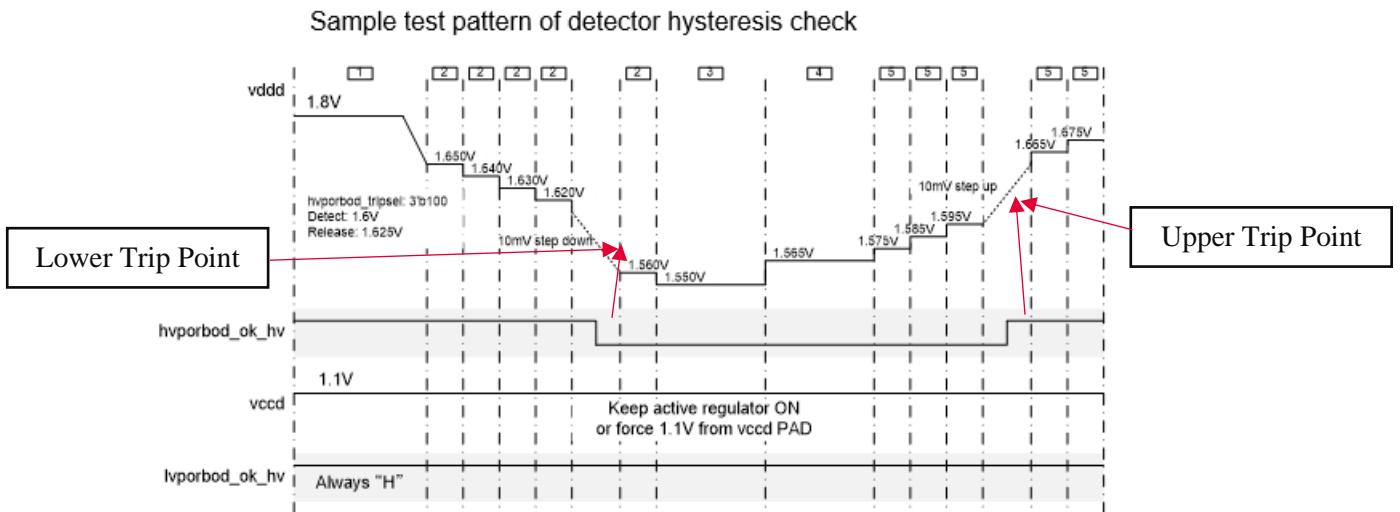
**Fig4.2d: Test Case1: Test Program Compiled with no errors and successfully generated Operating Sequences**



**Fig4.2e: Test Case1: Trim Codes Results Distribution done on Wafer Sort (70 dies tested) in loop of 10x run.**

### 4.3 Test Case2: POR Circuitry Hysteresis Screening at Wafer Sort

#### 4.3.1 Test Program Development



**Fig4.3a: POR screen for finding the Upper and Lower Trip points**

- ✓ The trimmed device for the comparator offset from Test Case1 will be screened(device is loaded with its corresponding trim code) to find out the Upper and Lower Trip points-**Fig4.3a** to get the Hysteresis (this is built so to have immunity from signal noises) of the POR circuitry.
- ✓ The Lower and Upper Trip points of comparator is found as follows:
  - **Fig4.3b** shows the computation of *Lower Trip point* of the comparator when the VDDD pin is swept from 1.7V(RampStart) to 1.4V(RampEnd) in steps of 0.002V(RampStep). Here the output of comparator that is OutputPinlist-“pl\_DDFT0” is checked for High value- through a pattern. The voltage of VDDD when the pl\_DDFT goes from high to low forms the Lower Trip point/Vtrip Fall point.
  - **Fig4.3c** shows the computation of *Upper Trip point* of the comparator when the VDDD pin is swept from 1.4V(RampStart) to 1.7V(RampEnd) in steps of -0.002V(RampStep). Here the output of comparator that is OutputPinlist-“pl\_DDFT0” is checked for Low value- through a pattern. The voltage of VDDD when the pl\_DDFT goes from Low to High forms the Upper Trip point/Vtrip Rise point.

```

suite S40SRSSA_HVPORBOD_TRIP_FALLING calls tml.IP_S40SRSSA.IP_S40SRSSA_Vmon_Trip
{
    TN = 813;
    HardBinOnFail = 7;
    ExecOn = #[ "SORT1", "SORT2", "SORT3", "CRI", "CHI" ];
    baseSpecs = #[ setupRef(common.level.VDDD_NOM_VDDA_NOM), setupRef(common.timings.STANDARD_TIME_250NS) ];
    InstructionFile = "InsFile/IP_S40SRSSA/FX3G2/S40SRSSA_HVPORBOD_SCREEN.ins";
    POW[YES] = {};
    InputPinlist = "pl_VDDD";
    OutputPinlist = "pl_DDFT0";
    ForceCurrent = 0;
    SettlingTime = 0.000002;
    RampStart = 1.7;
    RampEnd = 1.4 ;
    RampStep = 0.002;
    MinLimit = 1.461;
    MaxLimit = 1.589;
    Inverse = true;
    MeasureOutput = true;
    PartestOrFuntest = FUNTEST;
    Pattern = setupRef(vectors.Ensemble.CHECK_DDFT0_HIGH_Capture);
    NewTrimValue1 = 0;
    NewTrimValue2 = 0;
    ContainerLabelSaveMeas = "HVPORBOD_TRIP_FALLING";
    rbm_mode = REGULATED;
    acquire_mode_pattern = DFT_SAFEREG_SWD_ACQ;
    trim_pattern = TRIM;
    PatExpect=0;
    measurement.specification=setupRef(common.specs.Groups_Common);
}

```

**Fig4.3b: Lower trip point computation**

```

suite S40SRSSA_HVPORBOD_TRIP_RISING calls tml.IP_S40SRSSA.IP_S40SRSSA_Vmon_Trip
{
    TN = 815;
    ExecOn = #[ "SORT1", "SORT2", "SORT3", "CRI", "CHI" ];
    baseSpecs = #[ setupRef(common.level.VDDD_NOM_VDDA_NOM), setupRef(common.timings.STANDARD_TIME_250NS) ];
    InstructionFile = "InsFile/IP_S40SRSSA/FX3G2/S40SRSSA_HVPORBOD_SCREEN.ins";
    POW[YES] = {};
    InputPinlist = "pl_VDDD";
    OutputPinlist = "pl_DDFT0";
    ForceCurrent = 0;
    SettlingTime = 0.000002;
    RampStart = 1.4;
    RampEnd = 1.7;
    RampStep = -0.002;
    MinLimit = 1.461;
    MaxLimit = 1.589;
    Inverse = false;
    MeasureOutput = true;
    PartestOrFuntest = FUNTEST;
    Pattern = setupRef(vectors.Ensemble.CHECK_DDFT0_LOW_Capture);
    NewTrimValue1 = 0;
    NewTrimValue2 = 0;
    ContainerLabelSaveMeas = "HVPORBOD_TRIP_RISING";
    rbm_mode = REGULATED;
    PatExpect=0;
    acquire_mode_pattern = DFT_SAFEREG_SWD_ACQ;
    trim_pattern = TRIM;
    measurement.specification=setupRef(common.specs.Groups_Common);
}

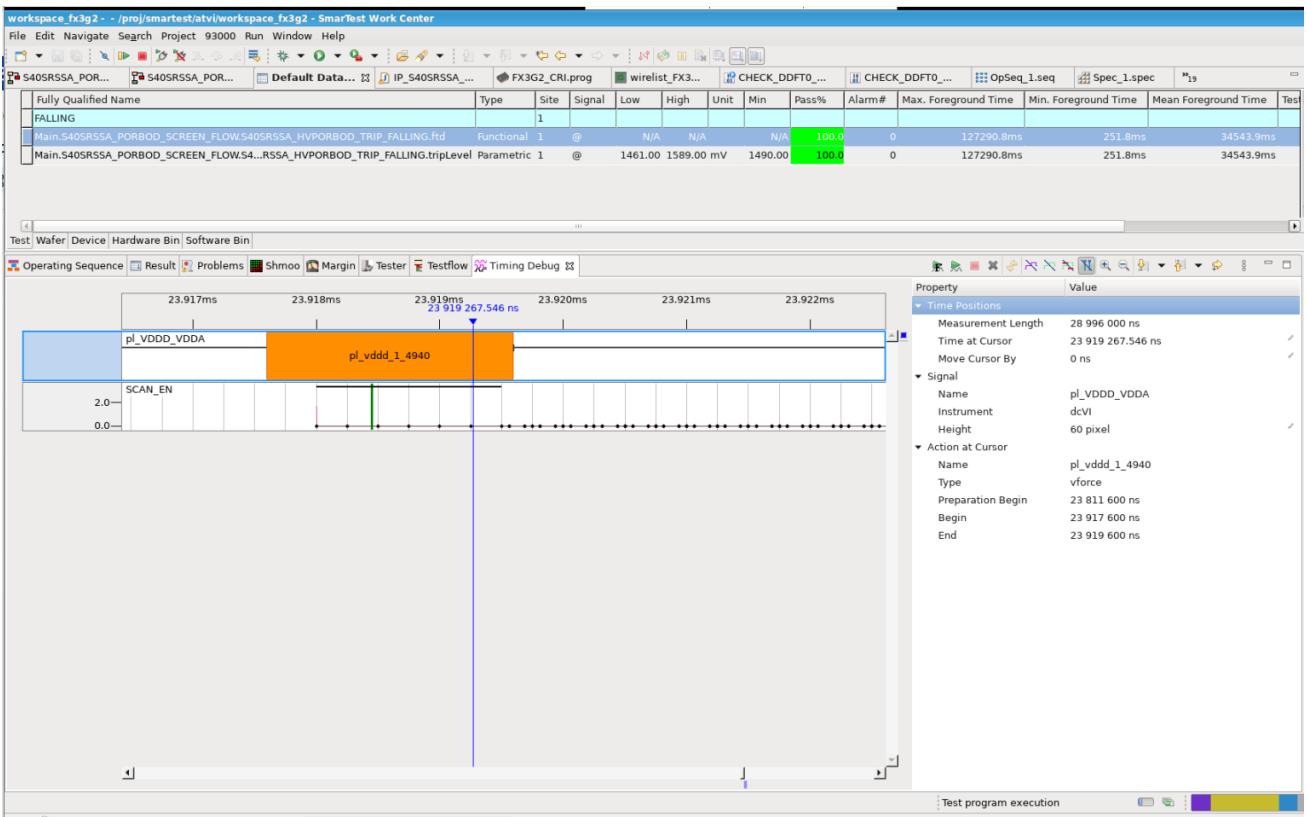
```

**Fig4.3c: Upper Trip Point computation**

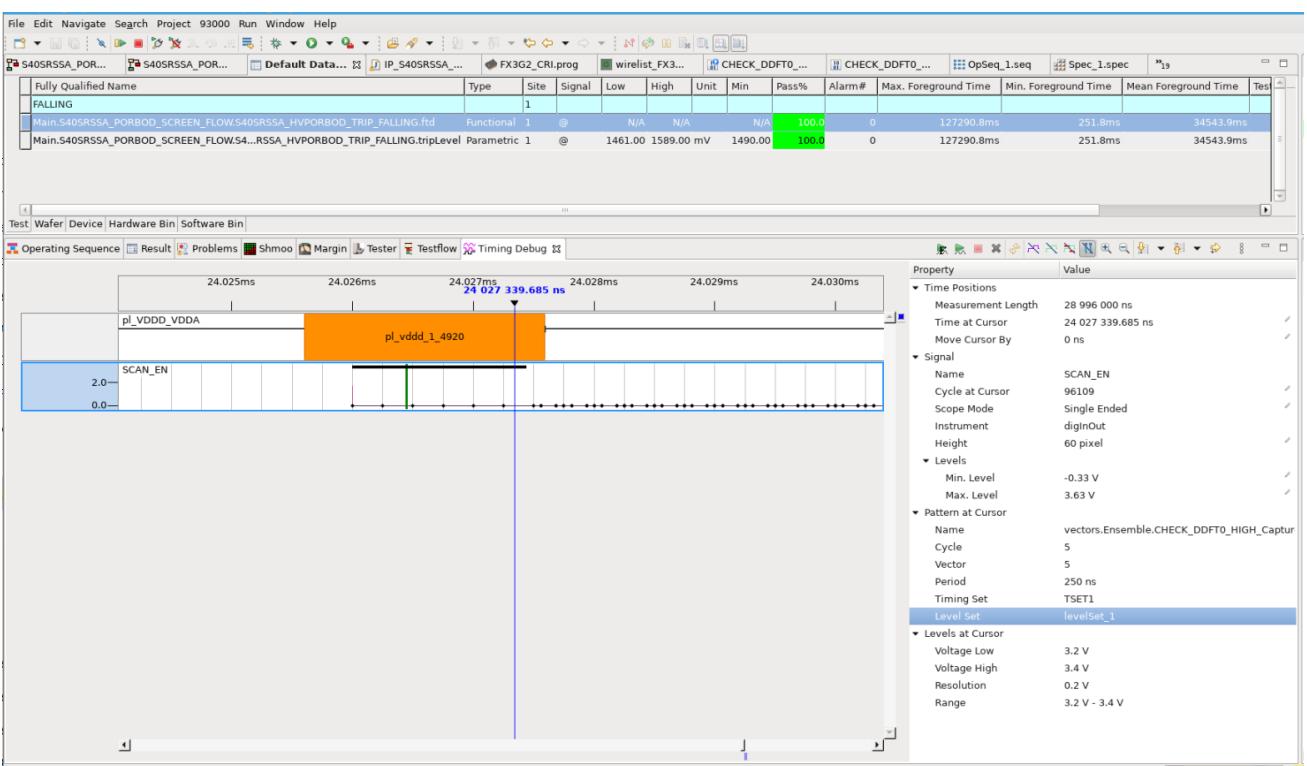
### 4.3.2 Validated Results

#### 4.3.2.1 LOWER TRIP POINT RESULTS (HVPORBOD FALLING)

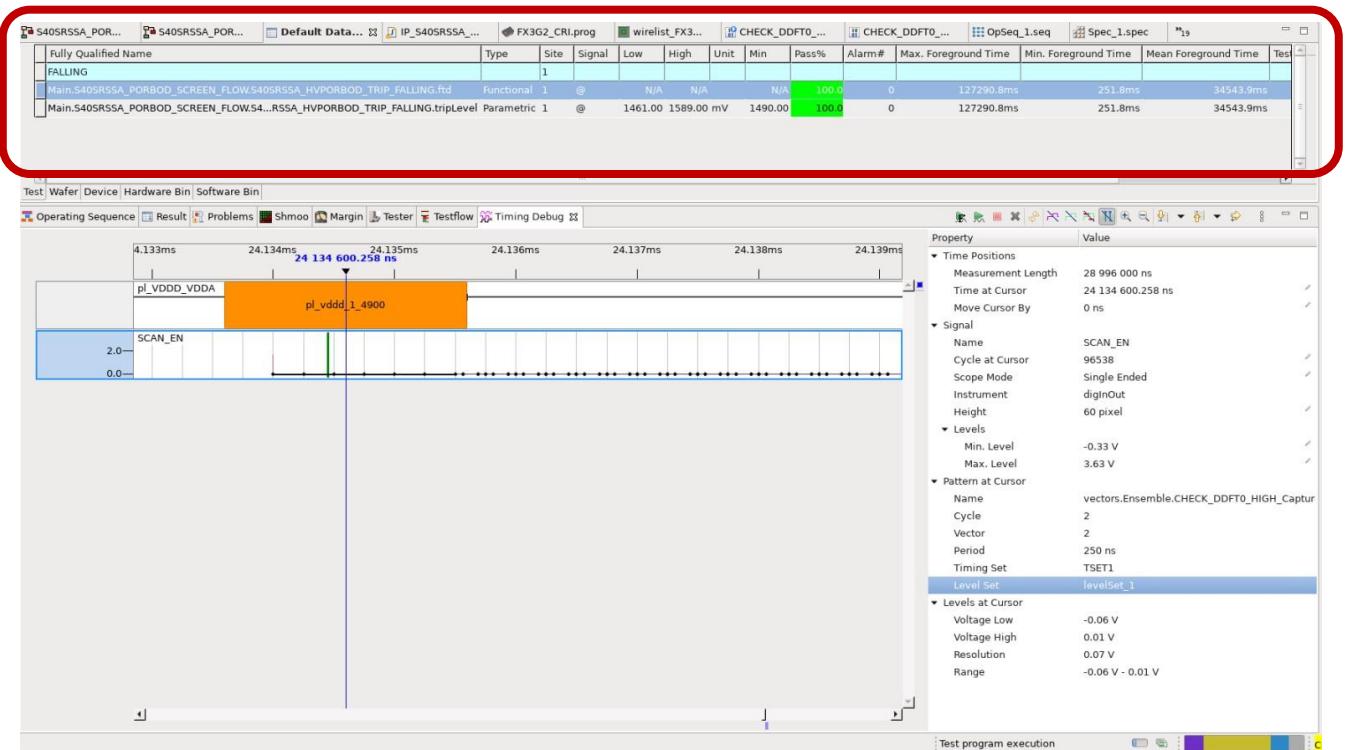
- ✓ As in **Fig4.3d** to **Fig4.3e** the DDFT0 pin is High for VDDD voltage 1.4940V and 1.4920V respectively and the DDFT0 pin transits to Low for VDDD voltage 1.4900V-**Fig4.3f** and remains low further-**Fig4.3g**. Hence the VDDD value of 1.4900V is the final “*Lower Trip Point*” value which is also highlighted in red box in the datalog.



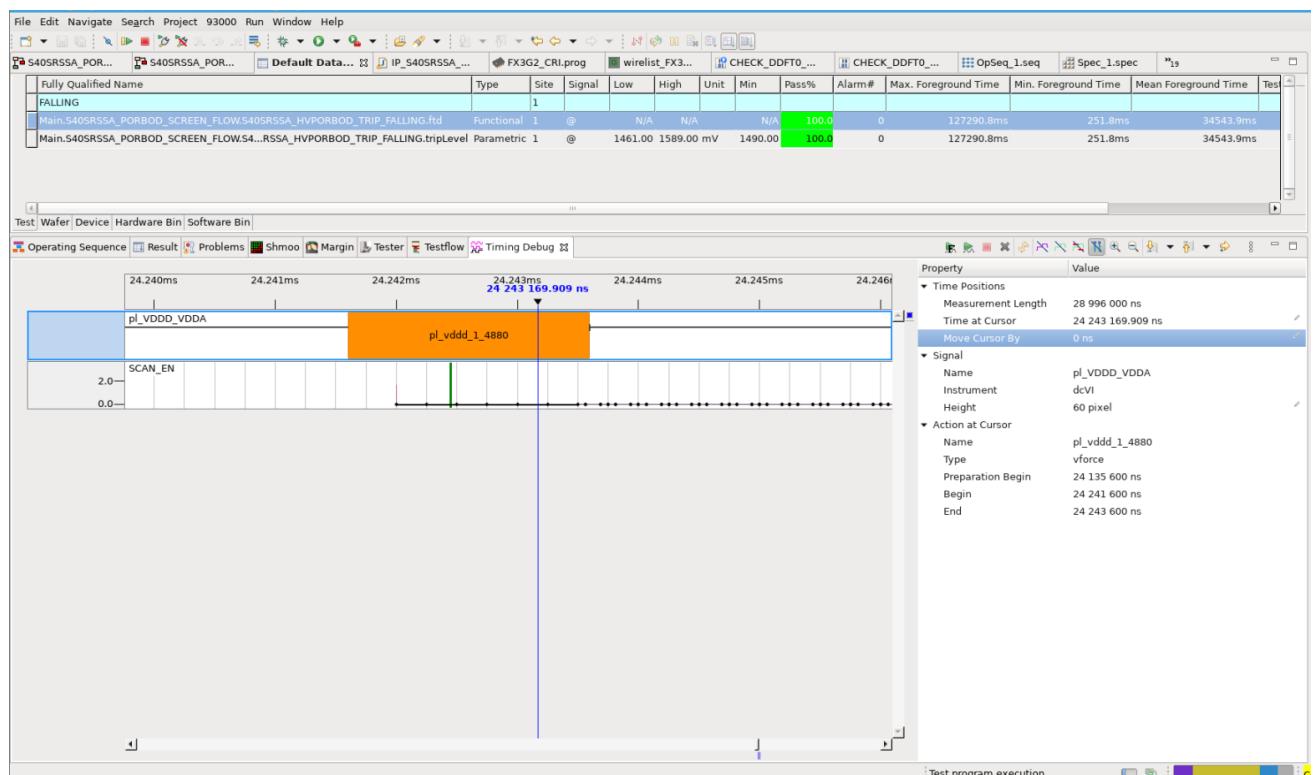
**Fig4.3d: Part1: DDFT0 Capture (Comparator Output) High for VDDD input of 1.4940V**



**Fig4.3e: Part2: DDFT0 Capture (Comparator Output) High for VDDD input of 1.4920V**



**Fig4.3f: Part2: DDFT0 Capture (Comparator Output) Low for VDDD input of 1.4900V**



**Fig4.3g: Part4: DDFT0 Capture (Comparator Output) Low for VDDD input of 1.4880V**

- ✓ Here in **Fig4.3i** the Comparator Output:DDFT0 pin bit sequence capture from the test pattern-  
**Fig4.3h** translated through the test program code here where resolution of each bit is 20 mV and  
the value at which the pin goes from 1 to 0 as pointed in arrow is the VDDD value- here it is  
1.4900V at “0” DDFT Reading. This forms the “***Lower Trip Point.***”

Signal					
X-Mode: x1					
Site Memory Sharing: OFF					
Actions	XRES	SWD_CLK	SWD_DATA	DDFT0	
Vector#	Instruction	Comment			
0			1 0 0	X	
1			1 0 0	C	
2			1 0 0	X	
3			1 0 0	X	
4			1 0 0	X	
5			1 0 0	X	
STOP					

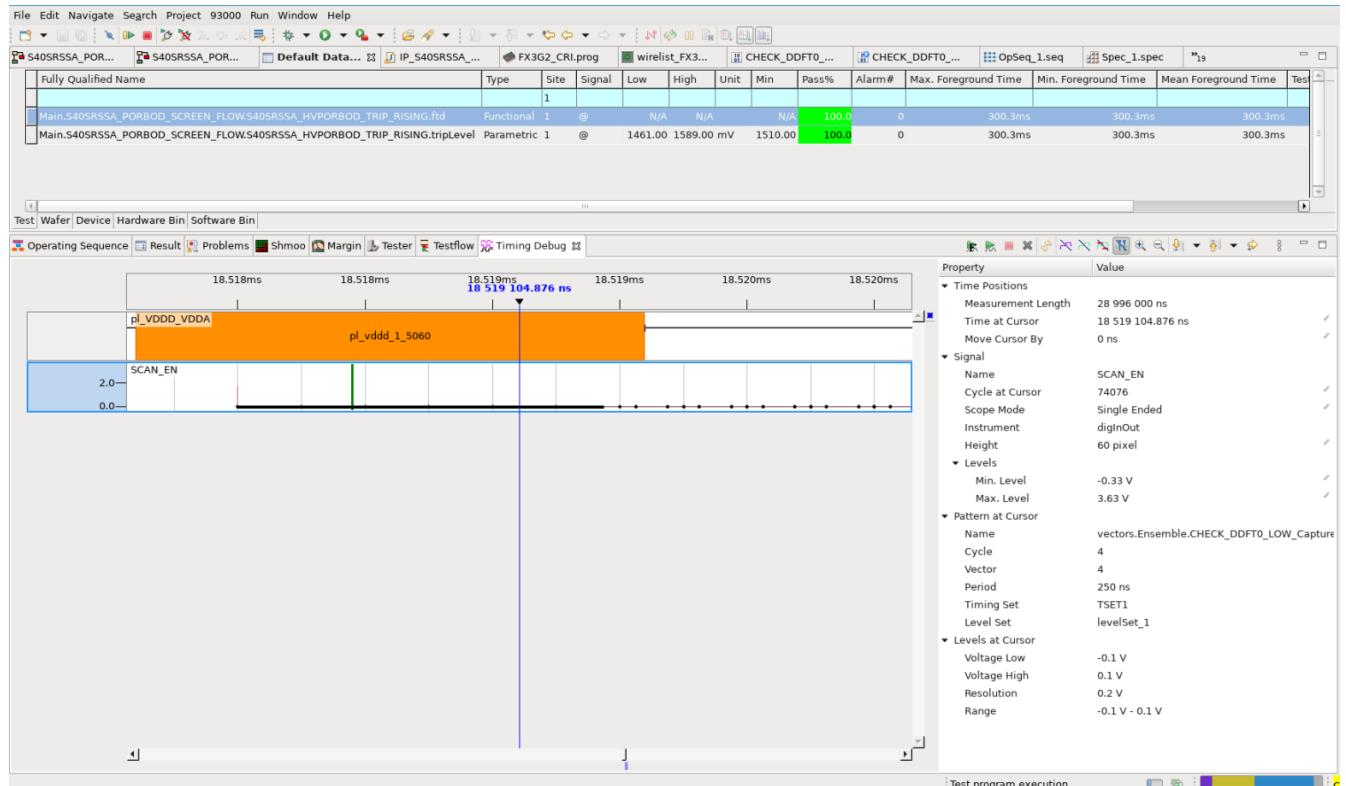
**Fig4.3h: Test Program Pattern Capture on DDFT0**

The screenshot shows the FX3G2\_SMT8\_Program\_Char\_Ht software interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, 93000, Run, Window, Help. The toolbar has icons for Open, Save, Print, Copy, Paste, Find, Replace, Cut, Copy, Paste, Delete, Undo, Redo, and others. The main window displays a Java code editor with the file IP\_S405RSSA/I/P\_S405RSSA/Vmon\_Trip.java. A red arrow points from the code editor down to a memory dump window. The memory dump window shows a 151-bit sequence starting with '1' followed by many zeros. Below the code editor is a navigation bar with tabs: Operating Sequence, Result, Problem. At the bottom, there's a search bar labeled 'Search FQN' and a tree view of project components under 'S405RSSA PORBOD SCREEN\_FLC'. The status bar at the bottom right shows 'Writable', 'Smart Insert', '317 : 18 : 13056', 'Test program execution', and a date/time stamp.

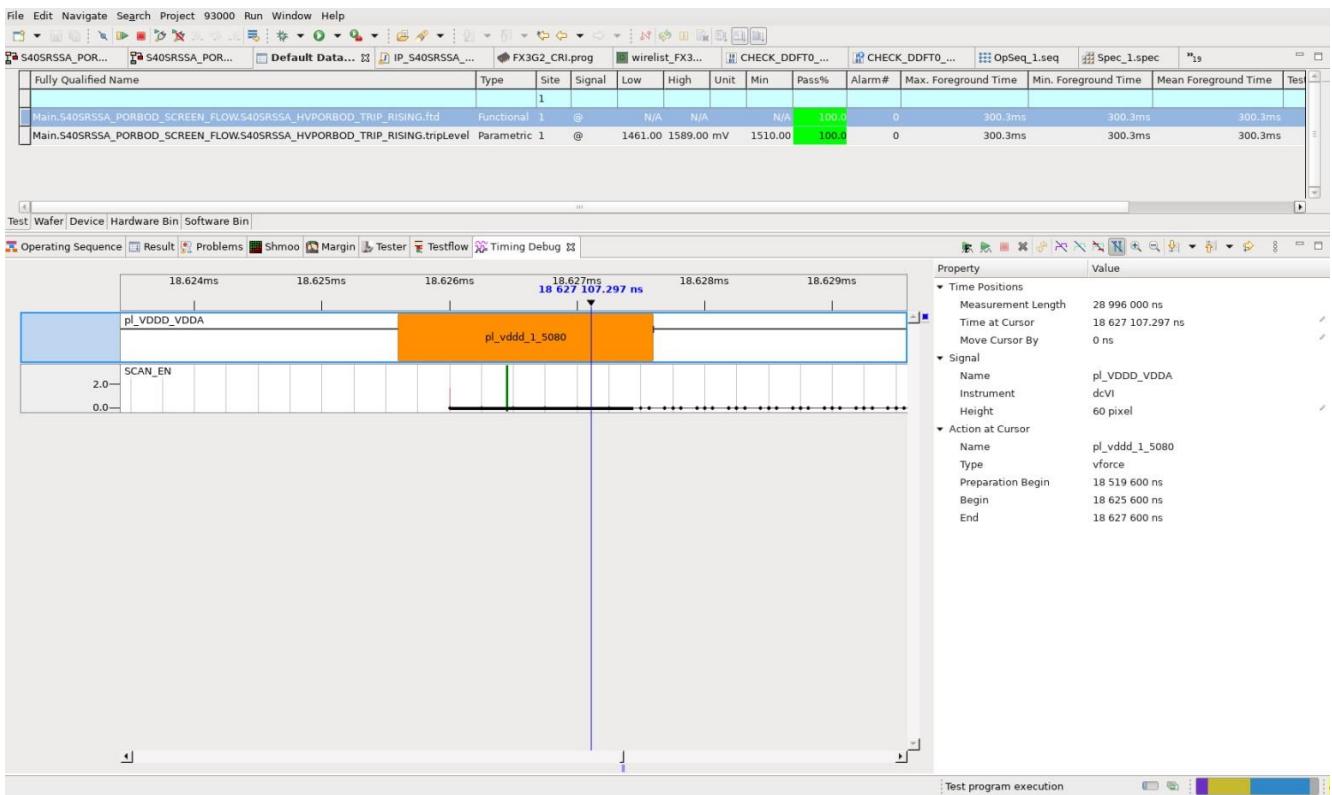
**Fig4.3i: Part5: Here the bit sequence of the DDFT 0 capture through Test Program Code**

#### 4.3.2.2 UPPER TRIP POINT RESULTS (HVPORBOD RISING)

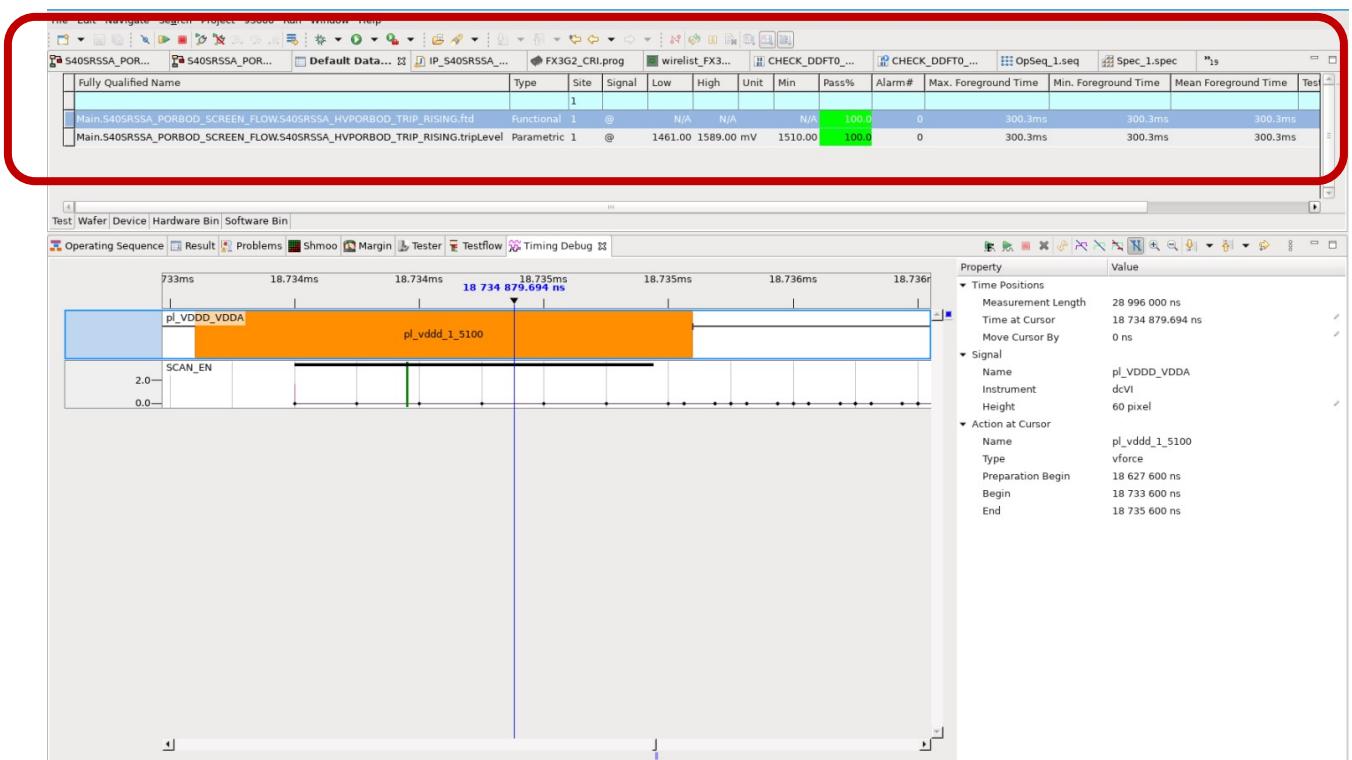
- ✓ As in **Fig4.3j** to **Fig4.3k** the DDFT0 pin is Low for VDDD voltage 1.5060V and 1.5080V respectively and the DDFT0 pin transits to High for VDDD voltage 1.5100V-**Fig4.3l** and remains high further-**Fig4.3gm**. Hence the VDDD value of 1.5100V is the final “*Upper Trip Point*” value which is also highlighted in red box in the datalog.



**Fig4.3j: Part1: DDFT0 Capture (Comparator Output) Low for VDDD input of 1.5060V**



**Fig4.3k: Part1: DDFT0 Capture (Comparator Output) Low for VDDD input of 1.5080V**



**Fig4.3l: Part4: DDFT0 Capture (Comparator Output) High for VDDD input of 1.5100V**

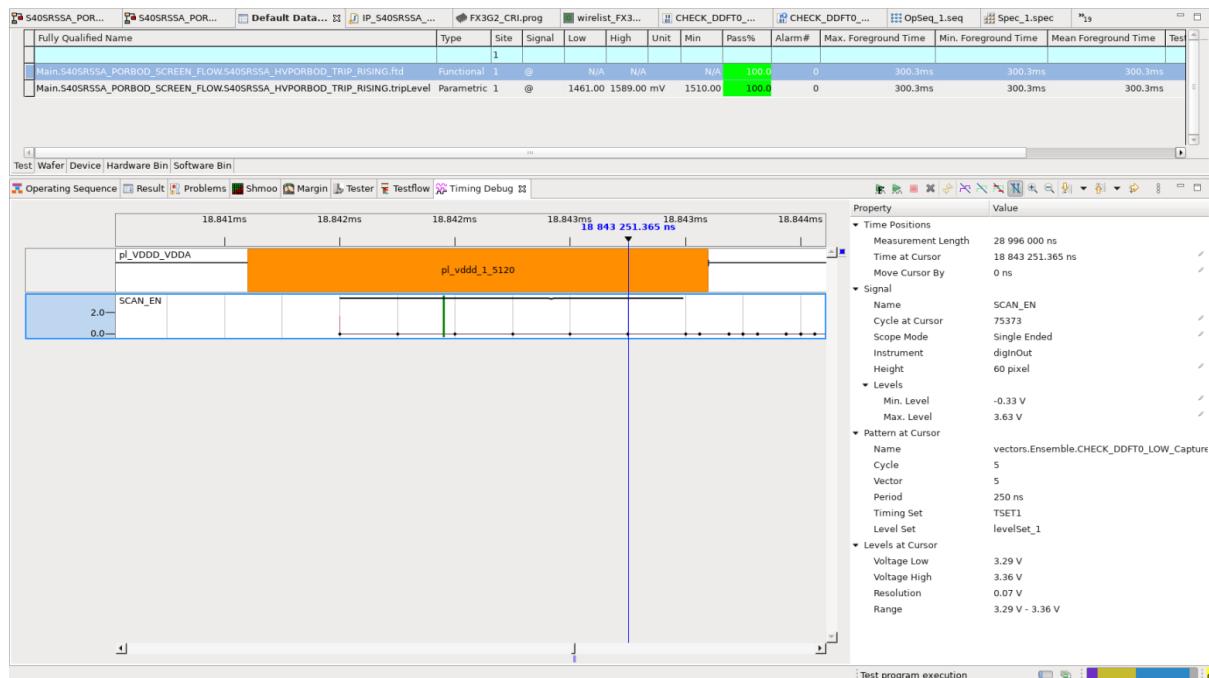


Fig4.3m: Part4: DDFT0 Capture (Comparator Output) High for VDDD input of 1.5120V

- ✓ Here in **Fig4.3n** the Comparator Output:DDFT0 pin bit sequence capture from the test pattern-  
**Fig4.3h** translated through the test program code here where resolution of each bit is 20 mV and  
the value at which the pin goes from 0 to 1 as pointed in arrow is the VDDD value- here it is  
1.5100V at “1” DDFT Reading. This forms the **“Upper Trip Point.”**

The screenshot shows the SmartTest Work Center interface with the following components:

- Top Bar:** workspace\_fx3g2 - FX3G2\_SMT8\_PROGRAM\_CHAR\_H7 / src/tm1/IP\_S40SRSSA/IP\_S40SRSSA\_Vmon\_Trip.java - /pro/smartert/atv/workspace\_fx3g2 - SmartTest Work Center
- File Menu:** File Edit Source Refactor Navigate Search Project 93000 Run Window Help
- Package Explorer:** Shows files like IP\_S40RSSA\_VH, IP\_S40RSSA\_VREF, IP\_Voltage\_Trim\_0.j, IP\_Voltage\_Trim.j, IP\_Voltage\_Trim2.j, IP\_Voltage\_TrimLDC, IP\_USB\_LVDS, MDUT, parser.ins, protocols, SCAN, and utils.
- Code Editor:** Displays Java code for IP\_S40RSSA\_VH, specifically handling bit sequences and trip values.
- Site Result:** Shows 2 Sites, 1 Available, 1 Enabled, 0 Passed, 0 Failed, 0% Total Yield.
- Search FDN:** Shows a list of flow definitions, including MDUT\_UPDATE\_FLOW, LEAKAGE\_TESTS\_FLOW, SCAN\_TESTS\_FLOW, S405RSSA\_BANDGAP\_TRIMS\_FLOW, and S405RSSA\_PORBOD\_TRIM\_FLOW.
- Bottom Status Bar:** Writable, Smart Insert, 324 : 12 : 13294, Test program execution, and a connected status icon.

**Fig4.3n: Part5:** Here the bit sequence of the DDFT 0 capture through Test Program Code

## **5.FIRMWARE PROGRAMMING THROUGH AUTOMATIC TEST EQUIPMENT(ATE)**

### **5.1 Introduction**

- ✓ Firmware is a piece of code residing in the Application Flash of the device which gets executed upon boot up of the device. This section provided details on the programming Firmware to the DUT through ATE. Firmware is made for custom specific applications- here custom firmware's are targeted for verify the Power on Reset and Mode Transition Functionalities for device characterization. To program the Firmware through ATE. the Firmware is coded in any IDE and is converted into a standard format known as .hex format- which is a machine interpretable file. This .hex file is then parsed as input to ATE which further process it and programs it into the Application Flash Area of the Device.

### **5.2 Test Program Development**

- ✓ Here the steps to Firmware Programming thorough ATE are as follows as in **Fig5.2a**
  - 1) Bulk\_Erase test erases the entire Application Flash
  - 2) ProgramFirmware test parses the .hex file and program the .hex data into the Application Flash.
  - 3) The VerifyFirmware test compares the .hex file with the actual contents in the Application Flash and verifies if device has the Firmware programmed properly with right contents.

```
1) suite Bulk_Erase_Final_Final calls tml.IP_BLOCK.IP_Functional{
baseSpecs = #[ setupRef(common.level.VDDD_NOM), setupRef(common.timings.STANDARD_TIME_FAST) ];
TN=1545;
HardBinOnFail=14;
rbm_mode=REGULATED;
acquire_mode_pattern=DFT_ENABLE_SWD_ACQ;
trim_pattern=SKIPTTRIM;
POW[YES] = {};
InstructionFile = "InsFile/IP_FLASH/FX3G2/P50C6_BULK_ERASE_USER_FLASH_SROM.ins";
}

2) suite ProgramFirmwareNonContig_Secure calls tml.IP_BLOCK.IP_FuncTestFlashDBM{
baseSpecs = #[ setupRef(common.level.VDDD_NOM), setupRef(common.timings.STANDARD_TIME_FAST) ];
TN=840;
HardBinOnFail=14;
acquire_mode_pattern=DFT_ENABLE_SWD_ACQ;
trim_pattern=SKIPTTRIM;
POW[YES] = {};
InstructionFile = "S40FIRMWARE_WRITE_FIRMWARE_NON_CONTIG_SECURE_MIX.ins";
hexFile
pinlist
columns
rows
verifySecurity
}
suite VerifyFirmware calls tml.IP_BLOCK.IP_VerifyFirmware{
baseSpecs = #[ setupRef(common.level.VDDD_NOM), setupRef(common.timings.STANDARD_TIME_FAST) ];
TN=906;
HardBinOnFail=11;
acquire_mode_pattern=DFT_ENABLE_SWD_ACQ;
POW[YES] = {};
hexFile
pinlist
numRows =1024;
bytesInRow =512;
mode = UPLOAD_CMP;
//mode = DIRECT_CMP;
}

3) = "InsFile/IP_S40FIRMWARE/fx3g2_por_es10.hex";
= "SDATA";
=1024;//BLE2:2048 as it is 1MB
=512;
=false;
```

**Fig5.2a: Firmware Programming Steps**

### 5.3 Validated Results

**Fig5.3a: Sample .hex file- INTEL Standard**

```
21 import common.protocols.Swd;
22
23 transactSeqDefinitions TrSeq_1
24 {
25- transactSeqDef TransAct_S405RSS_DFT_ENABLE_SWD_ACQ1
26 {
27     SWDR(DPIDCODE, 0x6BA02477);
28     SWDW(DPCTRL, 0x10000000);
29     SWDW(DPSELECT, 0x00000000);
30     SWDR(DPCTRL, 0x30000000);
31     SWDW(APCSW, 0x03000002);
32     IOW(0x40260100, 0x90000000);
33     IOR(0x40201410, 0x00000003);
34     IOR(0x40260100, 0xD0000000);
35 }
36
37- transactSeqDef iow(
38             UnsignedLong IN addr,
39             UnsignedLong IN data)
40 {
41     IOW(addr, data);
42 }
43
44- transactSeqDef start_sram_download(
45             UnsignedLong IN sram_addr)
46 {
47     SWDW(APCSW, 0x00000012);
48     SWDW(APTAR, sram_addr);
49 }
50
51- transactSeqDef tsd_row_0()
52 {
53     SWDW(APDRW, 0x00004000);
54     SWDW(APDRW, 0x10000093);
55     SWDW(APDRW, 0x00000000);
56     SWDW(APDRW, 0x10000a35);
57     SWDW(APDRW, 0x00000000);
58     SWDW(APDRW, 0x00000000);
59     SWDW(APDRW, 0x00000000);
60     SWDW(APDRW, 0x00000000);
61     SWDW(APDRW, 0x00000000);
62     SWDW(APDRW, 0x00000000);
63     SWDW(APDRW, 0x00000000);
64     SWDW(APDRW, 0x10000a11);
65     SWDW(APDRW, 0x00000000);
66     SWDW(APDRW, 0x00000000);
```

**Fig5.3b:** Generated transaction sequence to be executed by ATE in terms of SWD protocol after the test program parses the hex file.

- ✓ As in **Fig5.3c**, the ADR item shows the Application flash address and the HEX item show the .hex file data and the DEV items shows the data read back from the device from that address.

```
[DBG] : +-----+
[DBG] : | Main.VerifyFirmware (tml.IP_BLOCK.IP_VerifyFirmware)
[DBG] : +-----+
INFO: Blob#0 : addr=0x10000000 length=0x00005ae4
ADR=0x10000000 CO=      0 : HEX=0x08004000 DEV=> [S#1: =0x08004000]
ADR=0x10000004 CO=      1 : HEX=0x100009d3 DEV=> [S#1: =0x100009d3]
ADR=0x10000008 CO=      2 : HEX=0x0000000d DEV=> [S#1: =0x0000000d]
ADR=0x1000000c CO=      3 : HEX=0x10000a35 DEV=> [S#1: =0x10000a35]
ADR=0x10000010 CO=      4 : HEX=0x00000000 DEV=> [S#1: =0x00000000]
ADR=0x10000014 CO=      5 : HEX=0x00000000 DEV=> [S#1: =0x00000000]
ADR=0x10000018 CO=      6 : HEX=0x00000000 DEV=> [S#1: =0x00000000]
ADR=0x1000001c CO=      7 : HEX=0x00000000 DEV=> [S#1: =0x00000000]
ADR=0x10000020 CO=      8 : HEX=0x00000000 DEV=> [S#1: =0x00000000]
ADR=0x10000024 CO=      9 : HEX=0x00000000 DEV=> [S#1: =0x00000000]
ADR=0x10000028 CO=     10 : HEX=0x00000000 DEV=> [S#1: =0x00000000]
ADR=0x1000002c CO=     11 : HEX=0x10000a31 DEV=> [S#1: =0x10000a31]
```

**Fig5.3c: The Verification of Device with properly programmed .hex file contents**

Fully Qualified Name	Test#	Type	Signal	Low	High	Unit	Min	Max	Mean	Device#	Pass%	Fail#	Alarm#	Time
Main.MAIN_Firmware.WriteFirmware_POR.Bulk_Erase_Final.F	1545	Func...	@	N/A	N/A		N/A	N/A	N/A	1	100.0	0	0	
Main.MAIN_Firmware.WriteFirmware_POR.Bulk_Erase_Final.FUSE.P	1545	Func...	@	N/A	N/A		N/A	N/A	N/A	1	100.0	0	0	
Main.MAIN_Firmware.WriteFirmware_POR.ProgramFirmwareNonContig_Secure.ftd	840	Func...	@	N/A	N/A		N/A	N/A	N/A	1	100.0	0	0	
Main.MAIN_Firmware.WriteFirmware_POR.VerifyFirmware.ftd	906	Func...	@	N/A	N/A		N/A	N/A	N/A	1	100.0	0	0	

**Fig5.3d: Passing datalog for Successful Firmware Programming through ATE**

## **6.POWER ON RESET EXTENSIVE FUNCTIONAL CHARACTERISATION THROUGH AUTOMATIC TEST EQUIPMENT(ATE) ON PACKAGED UNITS**

### **6.1 Introduction**

- ✓ Once the parts are trimmed and screened in wafer sort, they are obtained as packaged units. As a part of extensive characterization, devices are subject to stresses across temperatures. Here the devices power supplies are subject to Linear monotonic, Focused and Random Ramps which test the functionality of the POR circuitry. **“During POR testing the decoupling capacitors on the device Load Board which limit the variations in voltage or decouple the supply noise/transients are desoldered so as to get proper slew rate applied to the DUT supply pins.”**

### **6.2 Test Program Development**

- ✓ Here as in **Fig6.2a**, the tests cases are that the all device pins are ramped together at the same time as in first picture and in second case the main supply VDDD is ramped first followed by other supplies.

```
suite GANG_POR_TestCase002 calls tml.IP_BLOCK.IP_CHAR_por_ramp_gang
{
    TN = 102;
    HardBinOnFail = 12;
    ExecOn = #[ "SORT1", "SORT2", "SORT3", "CRI", "CHI", "CLI" ];
    baseSpecs = #[ setupRef(common.level.VDDD_MIN), setupRef(common.timings.STANDARD_TIME) ];
    InstructionFile = "InsFile/Support/NO_APG.ins";
    POW_NO = true;
    PowerSupplyPin = "pl_POWER_PINS_NO_VCCD";
    InitialLevelSet = "common.level.POWERDOWN_POR";
    InitialTimingSet = "common.timings.STANDARD_TIME";
    BootStatusPin = "pl_POR_Status";
    BootStatusPattern = "InsFile/POR/POR_Reset_Status_Check.ins";
    TestCase = "ramp2";
    TweakCyclekhz = 100;
    PowerOffDelayTime = 0.5;
    FWDelayTime = 0.5;
    RegByPassPin = "pl_VCCD";
    TriggerPattern = "InsFile/POR/POR_Start_Trigger.ins";
    XRESPattern = "InsFile/POR/XRES_DeAssert.ins";
    acquire_mode_pattern = NOACQUIRE;
}

suite SPLIT_OTHERS_VDDD_POR_TestCase001 calls tml.IP_BLOCK.IP_CHAR_por_ramp_split
{
    TN = 301;
    HardBinOnFail = 12;
    ExecOn = #[ "SORT1", "SORT2", "SORT3", "CRI", "CHI", "CLI" ];
    baseSpecs = #[ setupRef(common.level.VDDD_MIN), setupRef(common.timings.STANDARD_TIME) ];
    InstructionFile = "InsFile/Support/NO_APG.ins";
    POW_NO = true;
    PowerSupplyPin = "pl_POWER_PINS_NO_VDDD|pl_VDDD";
    InitiallevelSet = "common.level.POWERDOWN_POR";
    InitialTimingSet = "common.timings.STANDARD_TIME";
    BootStatusPin = "pl_POR_Status";
    BootStatusPattern = "InsFile/POR/POR_Reset_Status_Check.ins";
    TestCase = "ramp1";
    TweakCyclekhz = 100;
    PowerOffDelayTime = 0.5;
    FWDelayTime = 0.5;
    RegByPassPin = "pl_VCCD";
    TriggerPattern = "InsFile/POR/POR_Start_Trigger.ins";
    XRESPattern = "InsFile/POR/XRES_DeAssert.ins";
    acquire_mode_pattern = NOACQUIRE;
}
```

**Fig6.2a: POR test cases**

- ✓ Here the ramps are modelled as custom waveform files standard format of ATE as in **Fig6.2b**, where each data point form the voltage applied to the device and the time duration between the data point forms the sampling rate. The maximum slew rate applied to the device depends on the data sheet spec. The total data points is 5000 voltages. A total of 100 ramps across temperatures of 125C, -45C and 25C the device will be subjected to.

```

ramp1.cwf
1 RampTest1
2 0.0
3 0.0
4 0.0
5 0.002
6 0.002
7 0.002
8 0.004
9 0.004
10 0.004
11 0.006
12 0.006
13 0.006
14 0.006
15 0.008
16 0.008
17 0.008
18 0.01
19 0.01
20 0.01
21 0.012
22 0.012
23 0.012
24 0.012
25 0.014
26 0.014
27 0.014
28 0.016
29 0.016
30 0.016

```

**Fig6.2b: POR ramp file**

- ✓ Prior to running the POR test cases the POR Firmware is programmed to device, once the XRES toggle is enabled through test program and the POR test case starts the Firmware starts running the code in device, the device is held in reset state until the device supplies are stable upon successful POR execution, the Firmware compares the device registers with its expected/default values upon successful completion of checks the Firmware is written such a way that device drives/toggles some of the GPIO pins to HIGH state, to ensure successful boot completion. **Fig6.3c** shows the pattern having the H compares from ATE upon successful boot.

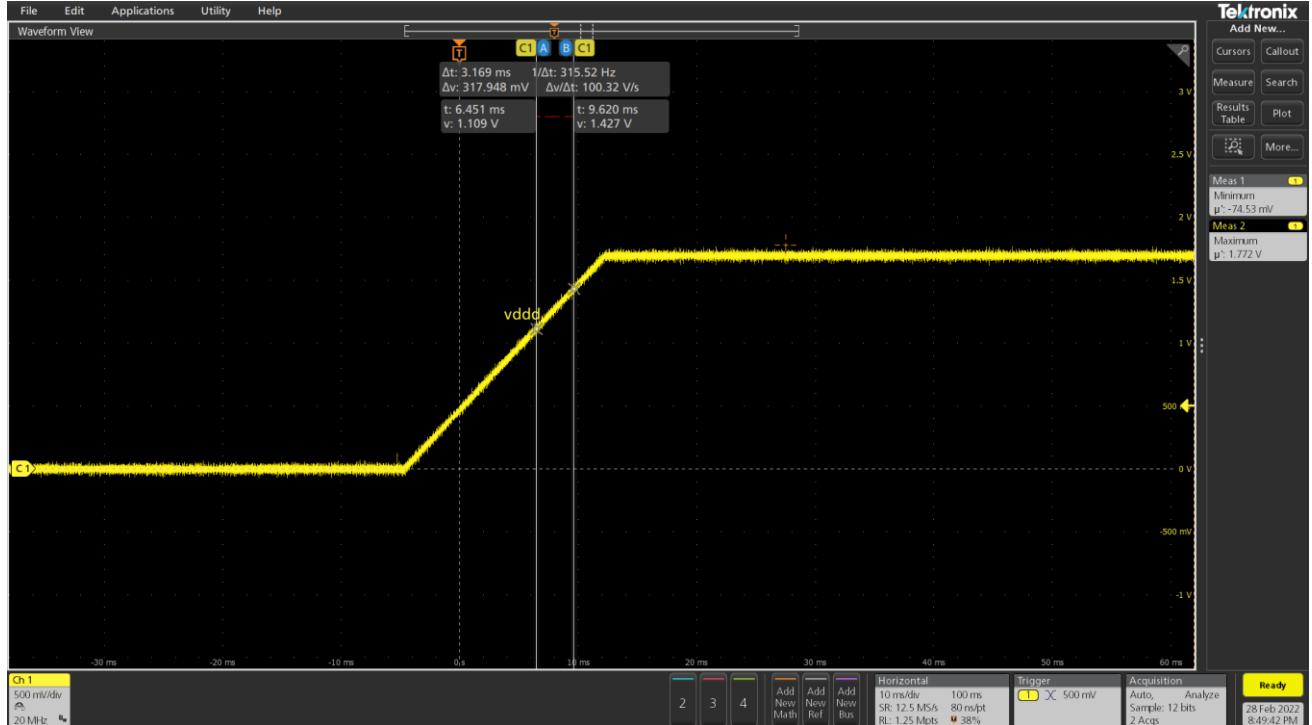
```

ramp1.cwf POR_GANGED.flow POR_Reset_Status_Check.ins
1
2 ADD_VECTOR( XRES=1, SWD_CLK=0, SWD_DATA=0, STATE3_POR=H, STATE4_POR=H, STATE1_POR=H, STATE2_POR=H, TEST_PASS_CM4_POR=H, TEST_PASS_CM0_POR=H ) < RPT=5 >
3
4

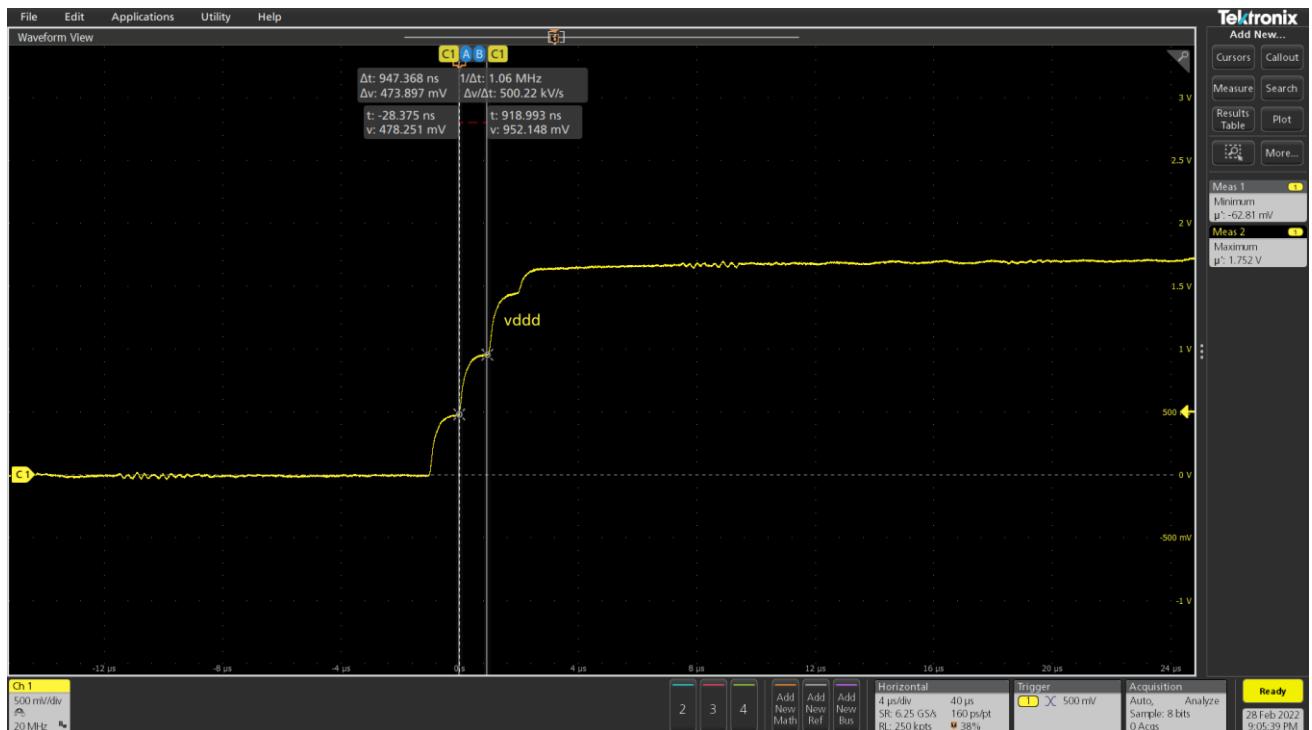
```

**Fig6.3c: Pattern check for successful boot up of Device**

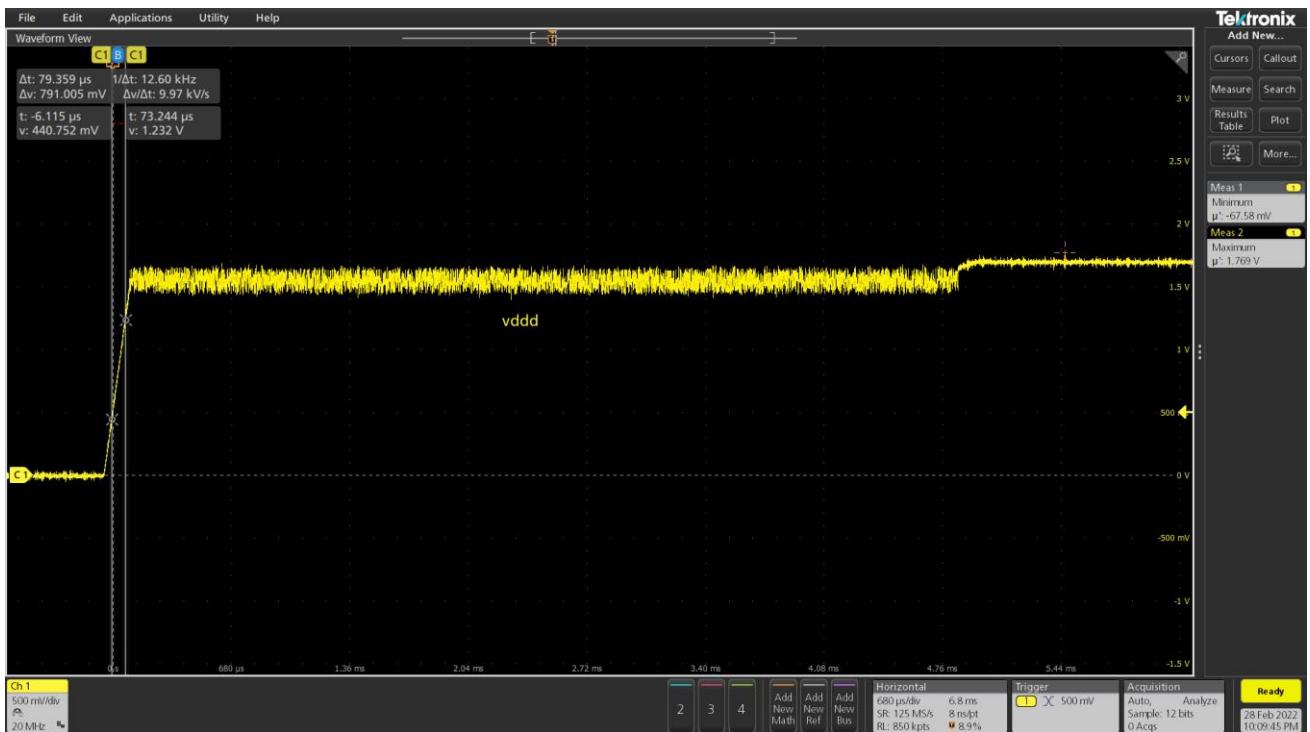
### 6.3 Validated Results



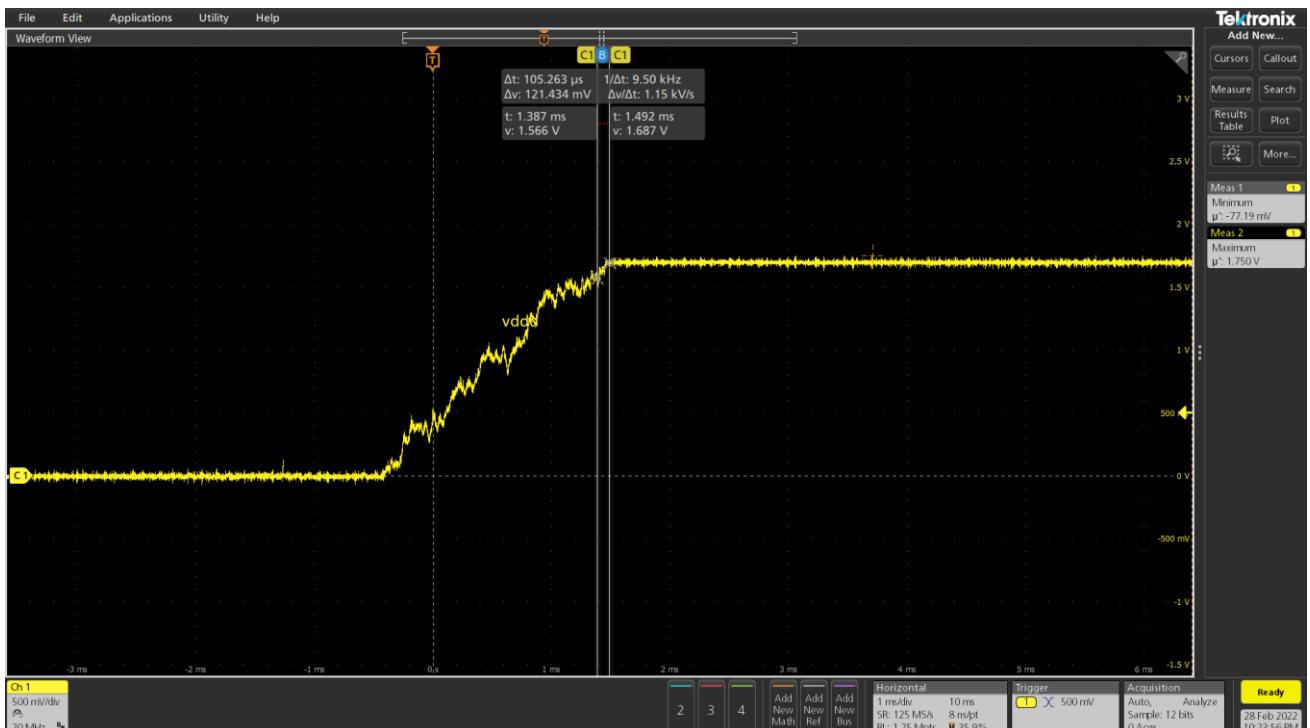
**Fig6.3a: Linear Monotonic Slow ramp applied to the Device for POR testing**



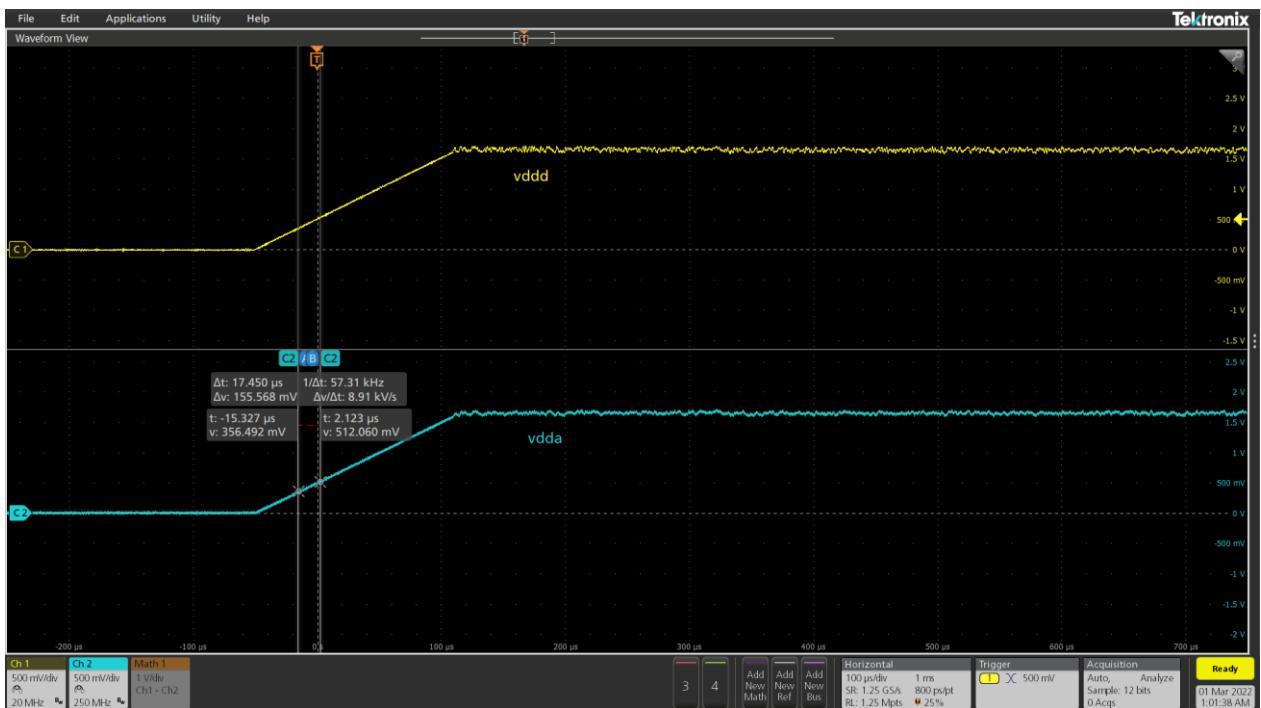
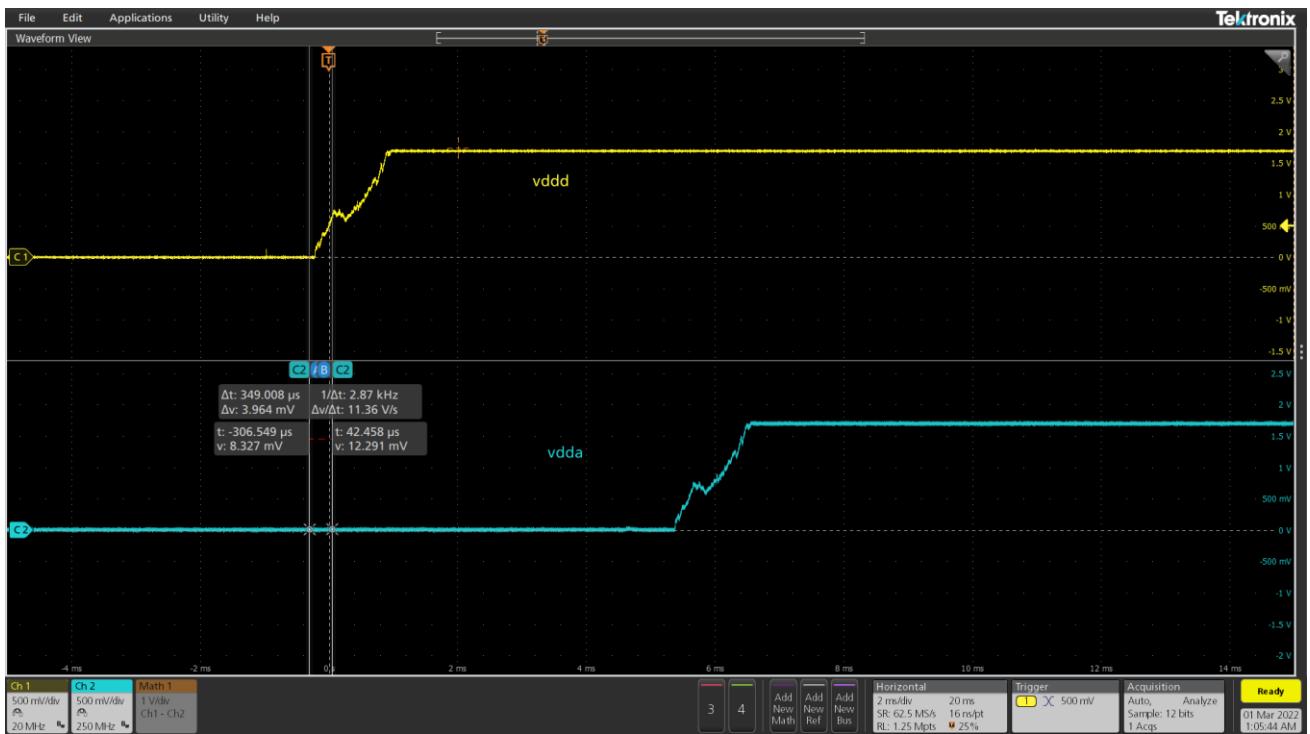
**Fig6.3b: Linear Monotonic Fast ramp applied to the Device for POR testing**



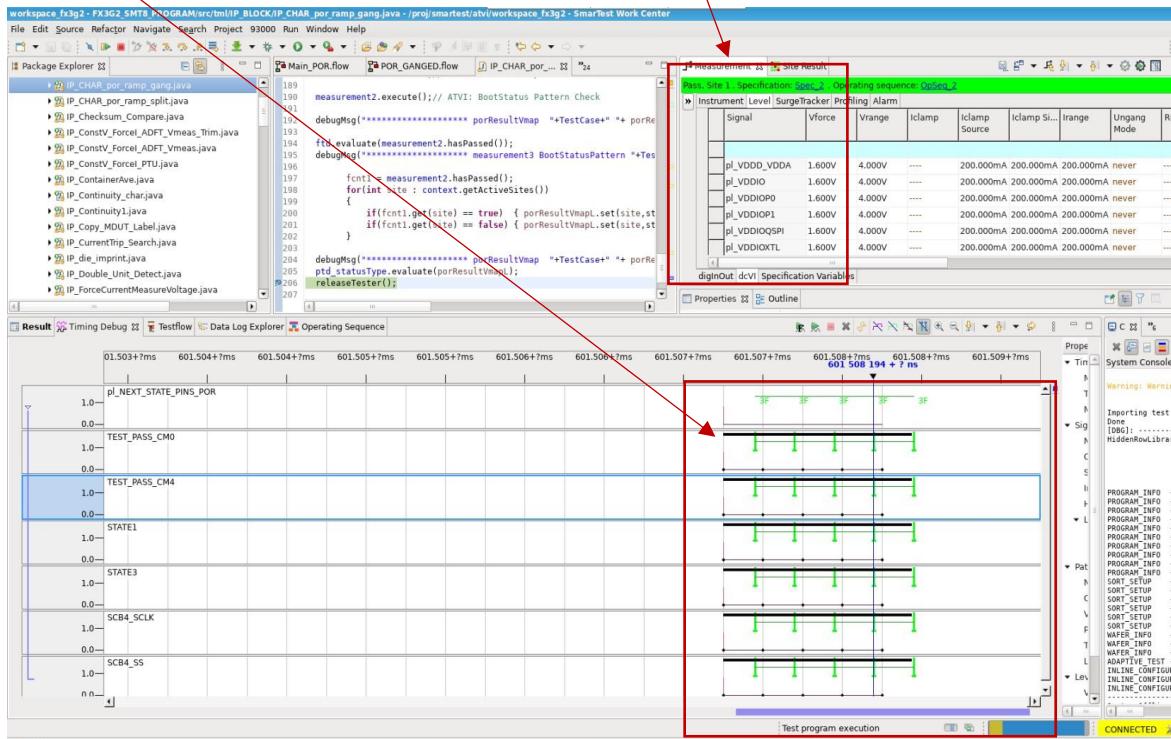
**Fig6.3c: Focused ramp targeted at Device POR point**



**Fig6.3d: Ramped ramp subject's device to varied voltages**

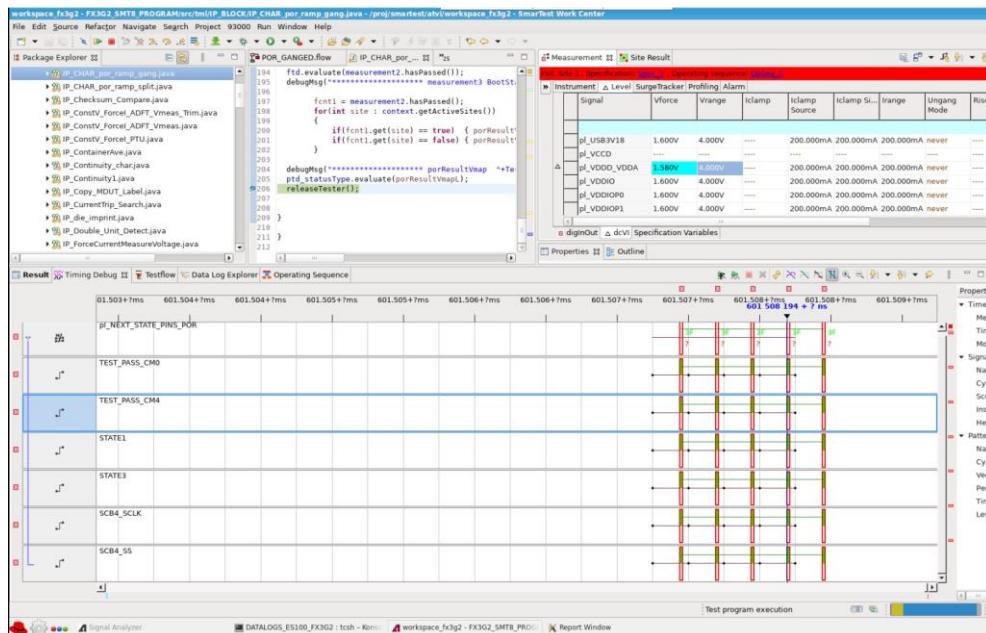


- ✓ Fig6.3g shows the 6 pins checked after the ramps are applied and it shows that the compared for High are passing, here the supplies are ramped till 1.6V as highlighted in red- datasheet spec for which the device is trimmed and screened for.



**Fig6.3g: Pattern checks for H pass after successful part boot up**

- ✓ Fig6.3h shows the pattern check fail as the voltage ramped for device is only 1.580V, which is below its datasheet POR spec and the part is in reset state and not booted up yet.



**Fig6.3h: Pattern checks for 'H' failed as part dint boot up**

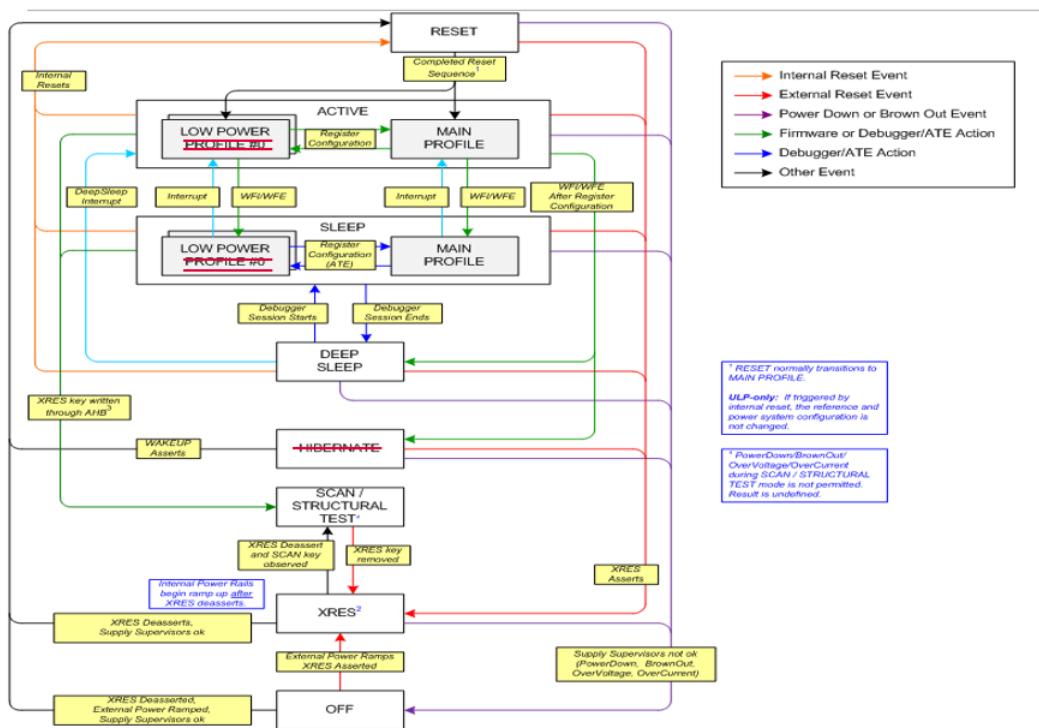
## **7 MODE TRANSITION (MT) FUNCTIONAL VERIFICATION ON PACKAGED UNITS**

### **7.1 Introduction**

- ✓ Every device during its functionality has many operating modes like active mode, reset mode, sleep mode and Deepsleep mode, each mode a certain pre-defined characteristic, and when interrupts are services it might cause the device to transit from one operating mode to another. From a testing perspective, initially we have firmware programmed from ATE to Device. The Firmware has sequence of events and functionalities of each blocks of the chip, the ATE serves are the stimulus generator for the device running or executing the Firmware, and even acts as a response Capture mechanism for the outputs sent out by the Firmware. The interrupts are either provided through the ATE or it is internally taken care in the Firmware which cover each block of the chip logic functionality. The Firmware consists of code which does the performs the Interrupt Service Routines (ISR's).

### **7.2 Test Program Development**

- ✓ **Fig7.2a** shows the state diagram of the device transition states when the interrupts are provided. Here in our MT validation, majority of these state transition of the device will be validated and at each state the device current consumption will be monitored.



**Fig7.2a: MT Validation state diagram**

- ✓ **Fig7.2b** shows the MT test cases sample list. Ex: A device in ACTIVE state moves to RESET state upon some event (POR, BOD, Watch Dog Event, System Request).

B	C	D	G
1			
2 Current State	Event	Next State	TEST CASES
15 ACTIVE	external supply removed	OFF	ACT_POR_OFF_VNR
16 ACTIVE	LVBOD/HVBOD triggers	OFF	ACT_BOD_OFF_VNR
17 ACTIVE	Assert XRES	XRES	ACT_XRES_XRES_VNR
18 ACTIVE	Watchdog resetfires	RESET	ACT_WDT_R_RESET_VNR
19 ACTIVE	MCWDT reset fires (one or more)	RESET	ACT_MCWDT0_R_RESET_VNR, ACT_MCWDT1_R_RESET_VNR
20 ACTIVE	sysresetreq	RESET	ACT_CM0_R_RESET_VNR
21 ACTIVE	fault_act_resetreq	RESET	ACT_FAULT_R_RESET_VNR
23 ACTIVE	Debugger reset	RESET	ACT_DBG_R_RESET_VNR
28 ACTIVE	Bit bang XRES_DFT_ENABLE key into PWR_DDOFT_XRES	ACTIVE	ACT_BB_XRES_KEY_DFT_ENABLE_VNR
29 SLEEP	external supply removed	OFF	SLP_POR_OFF_VNR
30 SLEEP	LVBOD/HVBOD triggers	OFF	SLP_BOD_OFF_VNR
31 SLEEP	Assert XRES	XRES	SLP_XRES_XRES_VNR
32 SLEEP	Watchdog reset fires	RESET	SLP_WDT_R_RESET_VNR
33 SLEEP	MCWDT reset fires (one or more)	RESET	SLP_MCWDT0_R_RESET_VNR, SLP_MCWDT0_R_RESET_VNR
34 SLEEP	sysresetreq	RESET	SLP_CM0_R_RESET_VNR
35 SLEEP	fault_act_resetreq	RESET	SLP_FAULT_R_RESET_VNR
37 SLEEP	Debugger reset	RESET	SLP_DBG_R_RESET_VNR
41 DEEPSLEEP	external supply removed	OFF	DPSLP_POR_OFF_VNR
42 DEEPSLEEP	LVBOD/HVBOD triggers	OFF	DPSLP_BOD_OFF_VNR
43 DEEPSLEEP	Assert XRES	XRES	DPSLP_XRES_XRES_VNR
44 DEEPSLEEP	Watchdog reset fires	RESET	DPSLP_WDT_R_RESET_VNR
45 DEEPSLEEP	MCWDT reset fires (one or more)	RESET	DPSLP_MCWDT0_R_RESET_VNR, DPSLP_MCWDT1_R_RESET_VNR
47 DEEPSLEEP	Debugger reset	RESET	DPSLP_DBG_R_RESET_VNR
49 ACTIVE	CM0+ in active mode CM4 in sleep mode	ACTIVE	CM0_ACT_CM4_SLP_NOM_VNR
50 ACTIVE	CM0+ in active mode CM4 in deepsleep mode	ACTIVE	CM0_ACT_CM4_DPSLP_NOM_VNR
51 ACTIVE	CM4 in retain fed mode with poweroff CM4_CTL_RETAIN=1;	ACTIVE	CM0_ACT_CM4_RET_NOM_VNR
52 ACTIVE	CM4 in off mode (without retention) CM4_CTL_RETAIN=0; (CM4_CTL_POWER=0)	ACTIVE	CM0_ACT_CM4_OFF_NOM_VNR
53 ACTIVE	CM4 in active mode	ACTIVE	CM0_ACT_CM4_NOM_NOM_VNR

**Fig7.2b: MT Test Cases Sample List**

- ✓ Prior to MT test suite validation-**Fig7.2c**, a .hex file having the MT Firmware will be programmed to device.

```

suite ACT_MCWDT0_R_RESET_VNR calls tml.IP_BLOCK.IP_CHAR_NextStateTest
{
    TN = 1135;
    HardBinOnFail = 4;
    ExecOn = #[ "SORT1", "SORT2", "SORT3", "CRI", "CHI", "CLI" ];
    baseSpecs = #[ setupRef(common.level.VDDD_NOM), setupRef(common.timings.STANDARD_TIME_MT) ];
    InstructionFile = "InsFile/MT/MT_NO_APG.ins";
    POW_NO = true;
    PinlistLog = "pl_NEXT_STATE_PINS";
    PowerOffDelayTime = 0.1;
    BootUpDelay = 0.5;
    XRES_Toggle = "InsFile/MT/XRES_TOGGLE.ins";
    CheckInitialStatePat = "InsFile/MT/MT_TPHH_STHH.ins";
    TestAdvancePat = "InsFile/MT/MT_CS_TRIGGER.ins";
    TweakDelay = 0.2;
    CheckTweakPat = "InsFile/MT/MT_TPHH_STHH.ins";
    ConfigDelay = 0.02;
    CheckConfigDonePat = "InsFile/MT/MT_TPLL_STLL.ins";
    AdvanceDelay = 0.3;
    CheckCurrentStatePat = "InsFile/MT/MT_TPLL_STLL_DDFTL.ins";
    TransitionEventPat = "InsFile/MT/MT_NO_APG.ins";
    TransitionEventPatCM0 = "InsFile/MT/MT_NO_APG.ins";
    TransitionDelay = 2;
    CheckNextStatePat_A = "InsFile/MT/MT_TPHH_STHH_DDFTL.ins";
    initialPowerdown = 1;
    TestID1 = 0x000000C9;
    TestID0 = 0x7C970000;
    Pinlist_im eas = "pl_VDD";
    NODFTACQPattern = "InsFile/IP_MX540SRSS/FX3G2/EFUSE_NO_DFT_CM0_DAP_SWD_ACQ.ins";
    acquire_mode_pattern = NOACQUIRE;
    nbootno = 1;
}

```

**Fig7.2c: Mode Transition Test Suite for Microcontroller Watch Dog Interrupt**

- ✓ As in **Fig7.2c** the test methods- **IP\_CHAR\_NextStateTest.java** has the sequence of events getting executed upon device power up and Firmware parallelly executing from device, the patterns corresponding to the sequence of events as in **Fig7.2d** will be executed from the ATE. The Test Advance Pattern is the Trigger patterns which enabled transition to the next state. The transitions event pattern provides an event or interrupt for transiting from current state of device to next state, this can be given through ATE or internally firmware programmed. Rest of the patterns are used for checking the device status through GPIO pins toggling at all stages/states.

```

1)Initial Pattern
2)TEST ID send
3)CheckInitialStatePat
4)TestAdvancePat
5)CheckTweakPat
6)TestAdvancePat
7)CheckConfigDonePat
8)TestAdvancePat
9)CheckCurrentStatePat
10)TransitionEventPat
11)TEST ID resend - only for resend id test cases for nboot=1
12)CheckNextStatePat_A

```

**Fig7.2d: Mode Transition sequence of Event checks from ATE**

### 7.3 Validation Results

- ✓ EX: in MCWDT Active mode test cases, the Firmware identifies it as a MCWDT test case when it receives the TESTID0 and TESTID1 which is unique for each test case, once it receives the Test ID, it puts some GPIO pins in HIGH state- which will be verified in ATE, next the ATE send a trigger pulse, this will enable the firmware to tweak some of the registers that is checking the registers for default value after boot up, upon successful tweak- the Firmware will toggle the state of some GPIO PINS, then the ATE send another trigger pulse, now the Firmware will perform a configuration of registers for the MCWDT functionality- once completed, the Firmware will toggle the GPIO PINS again- which will be checked by ATE, then the ATE sends another triggers pulse, now the Firmware will start executing the interrupt either it can be internally coded in Firmware or if it is external it will be given though ATE. The Current and Next State- refers to the device mode prior and post the interrupt is being services accordingly. For both the Current and Next state

Executions – the Firmware toggles the GPIO pins- which will be checked by ATE. The **Fig7.3a to Fig7.3f** shows the sequence of events are results captured at each transition stage.

```

setup protocolInterface SWD1
{
    transactSeq dsa_gen.CRI.Main.MT_ACTIVE_NS_NOM.ACT_MCWDTO_R_RESET_vNVR.TrSeq_3.TransAct_EFUSE_NO_DFT_CMO_DAP_SWD_ACQ1 TransSeqDef_EFUSE_NO_DFT_CMO_DAP_SWD_ACQ1

    transactSeq dsa_gen.CRI.Main.MT_ACTIVE_NS_NOM.ACT_MCWDTO_R_RESET_vNVR.TrSeq_3.iow _call_iow_trSeq_1
    {
        addr = 0x08014f08;
        data = 0x00000000;
    }

    transactSeq dsa_gen.CRI.Main.MT_ACTIVE_NS_NOM.ACT_MCWDTO_R_RESET_vNVR.TrSeq_3.iow _call_iow_trSeq_2
    {
        addr = 0x08014f00;
        data = 0x7c970000;
    }

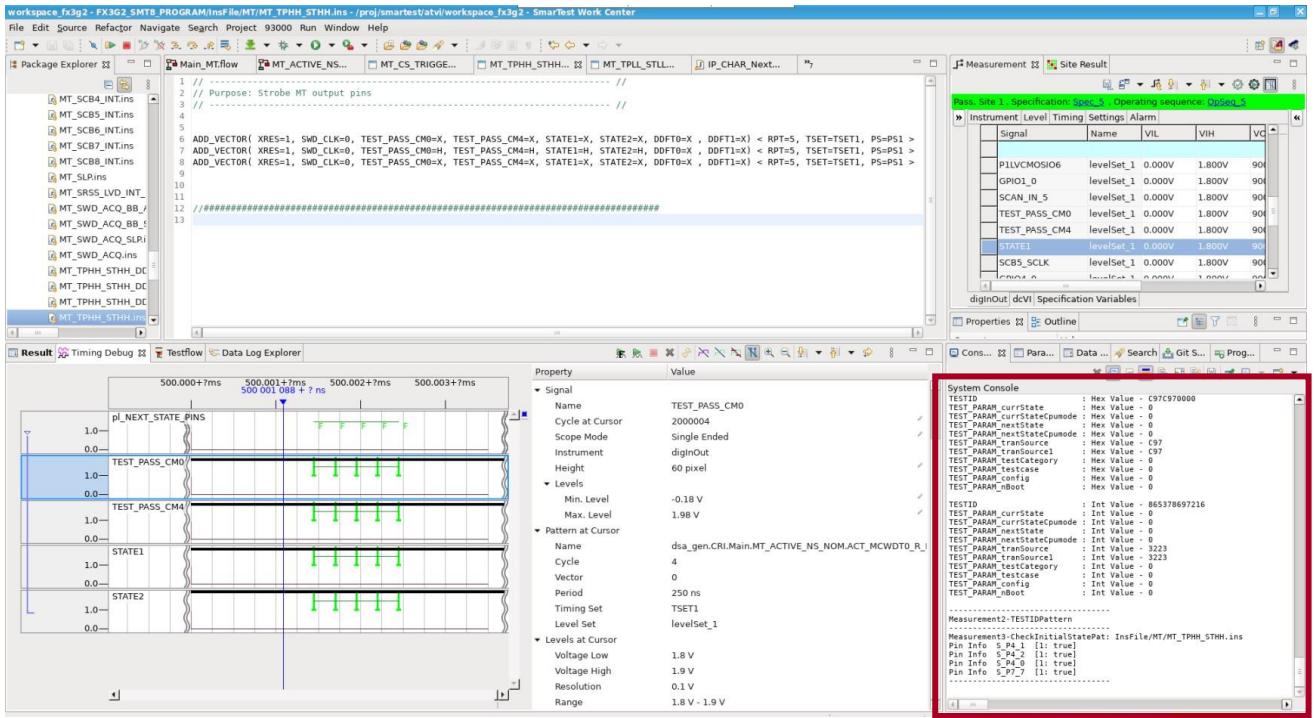
    transactSeq dsa_gen.CRI.Main.MT_ACTIVE_NS_NOM.ACT_MCWDTO_R_RESET_vNVR.TrSeq_3.iow _call_iow_trSeq_3
    {
        addr = 0x08014f04;
        data = 0x000000c9;
    }

    transactSeq dsa_gen.CRI.Main.MT_ACTIVE_NS_NOM.ACT_MCWDTO_R_RESET_vNVR.TrSeq_3.iow _call_iow_trSeq_4
    {
        addr = 0x08014f08;
        data = 0x4d544944;
    }

    transactSeq TransSeqDef_EFUSE_NO_DFT_CMO_DAP_SWD_ACQ1;
    transactSeq _call_iow_trSeq_1;
    transactSeq _call_iow_trSeq_2;
    transactSeq _call_iow_trSeq_3;
    transactSeq _call_iow_trSeq_4;
}

```

**Fig7.3a: TEST ID0(address:0x008014f04 and data:0x7c970000) and TEST ID1(address:0x008014f00 and data:0x000000c9)**



**Fig7.3b: Check Initial State**

Console printing of the MT GPIO PINS pattern checked results - PASS(true) and FAIL(false)

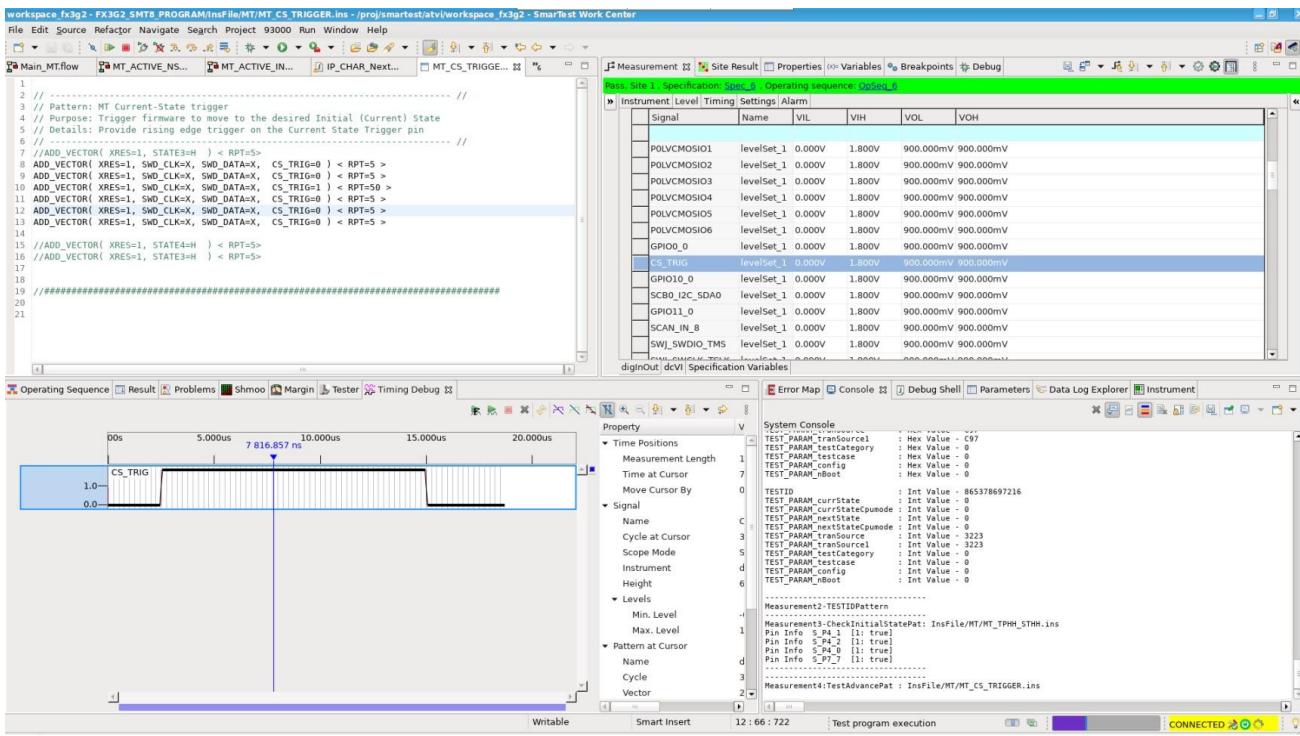


Fig7.3c: CS Trigger provided after every step pattern execution- Test Advance Pattern

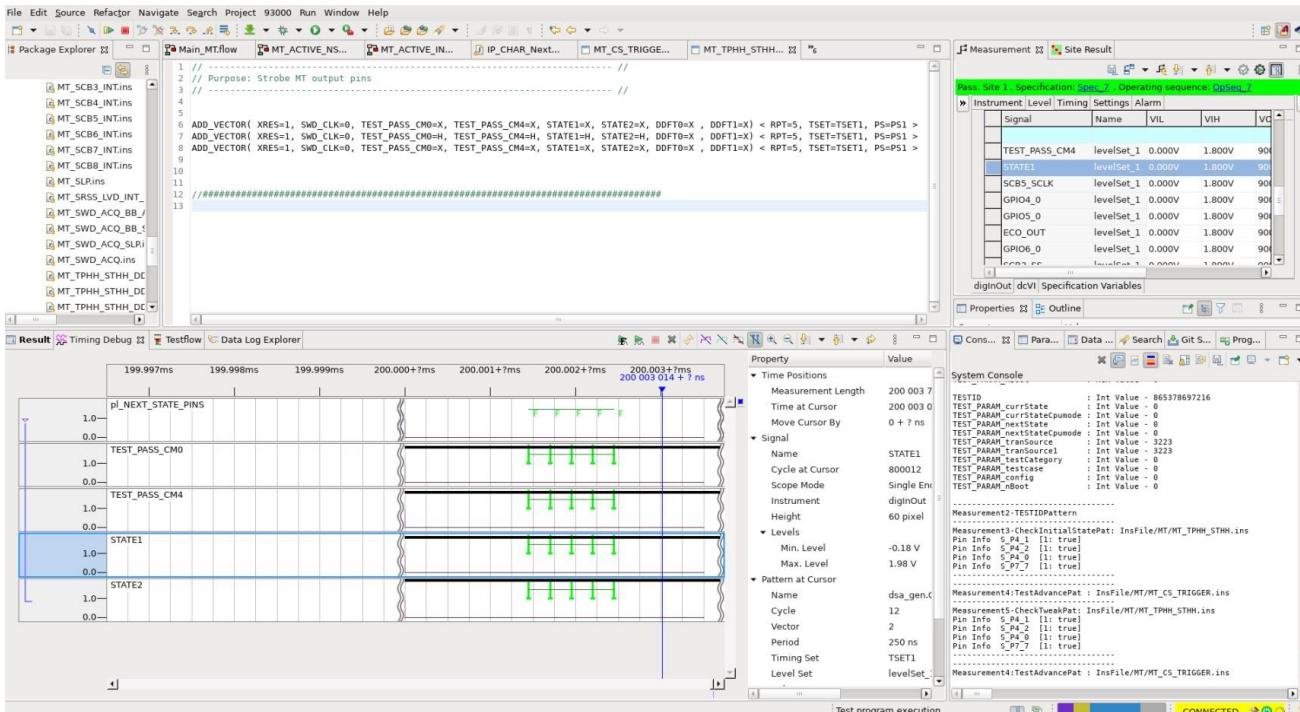
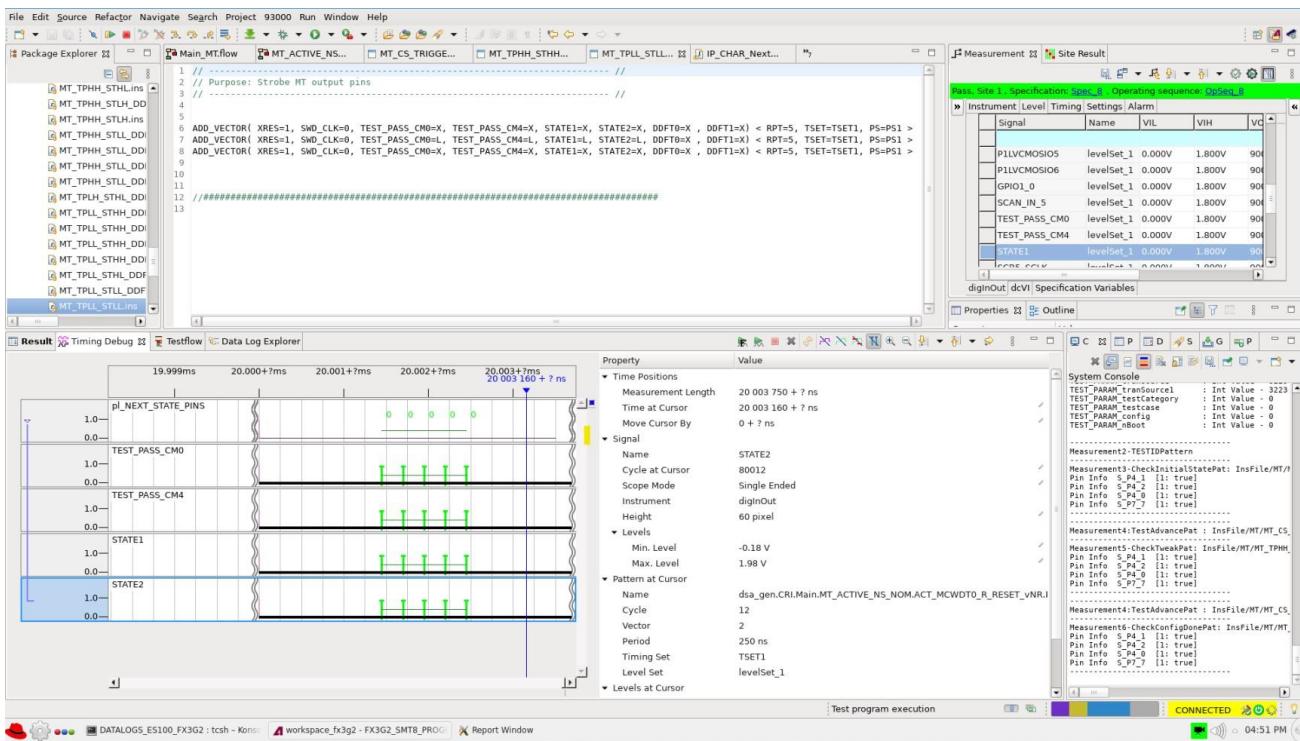
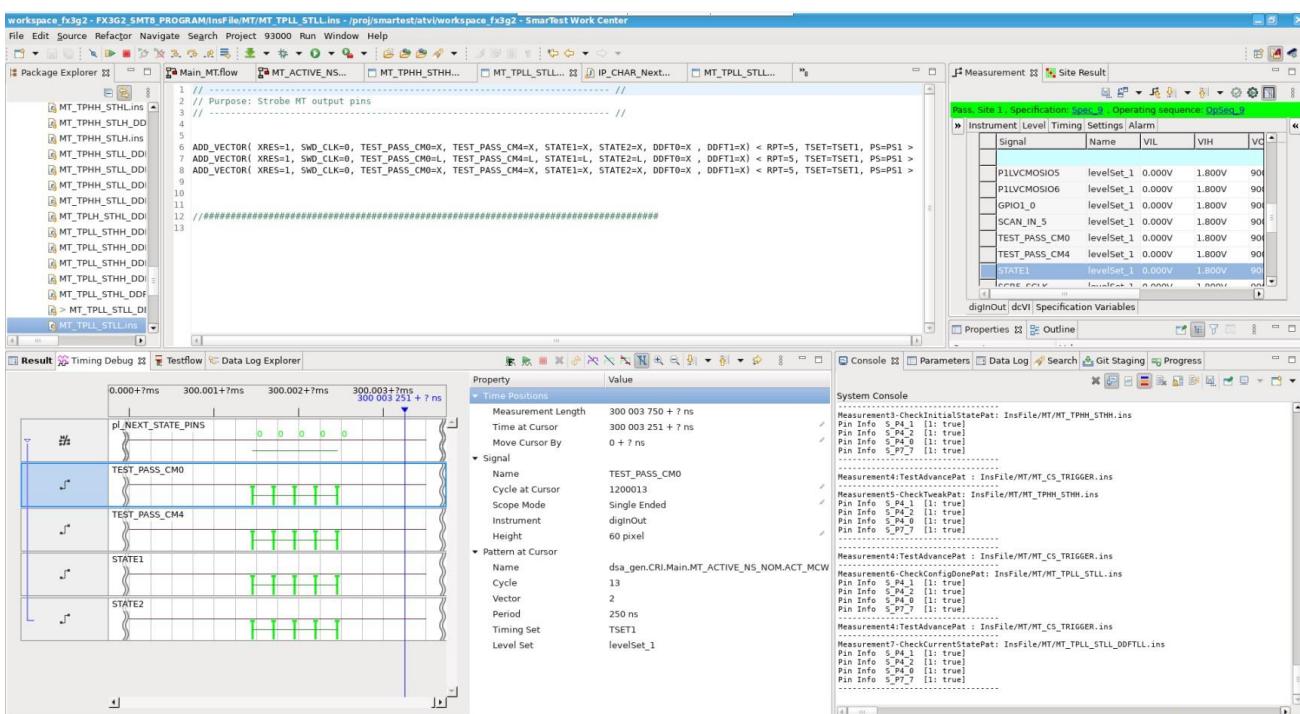


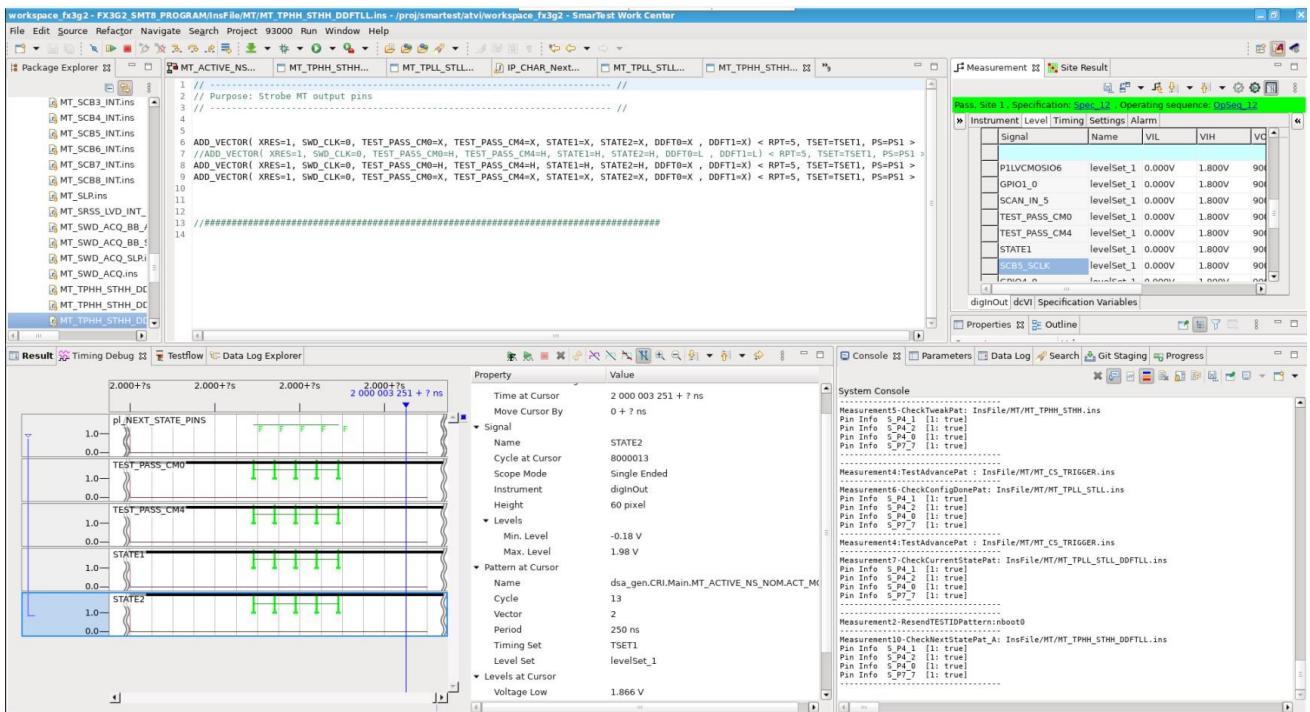
Fig7.3d: Check Tweak State



**Fig7.3e: Check Config Done State**

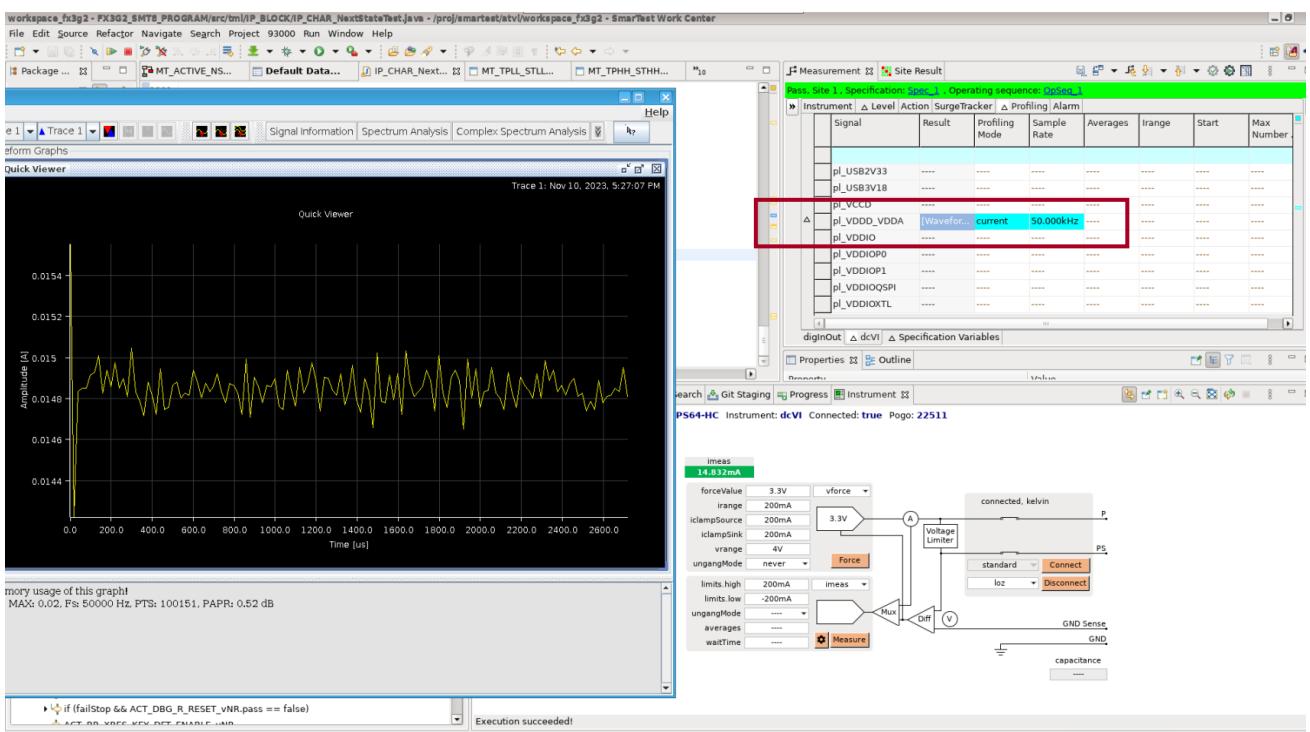


**Fig7.3f: Check Current State**

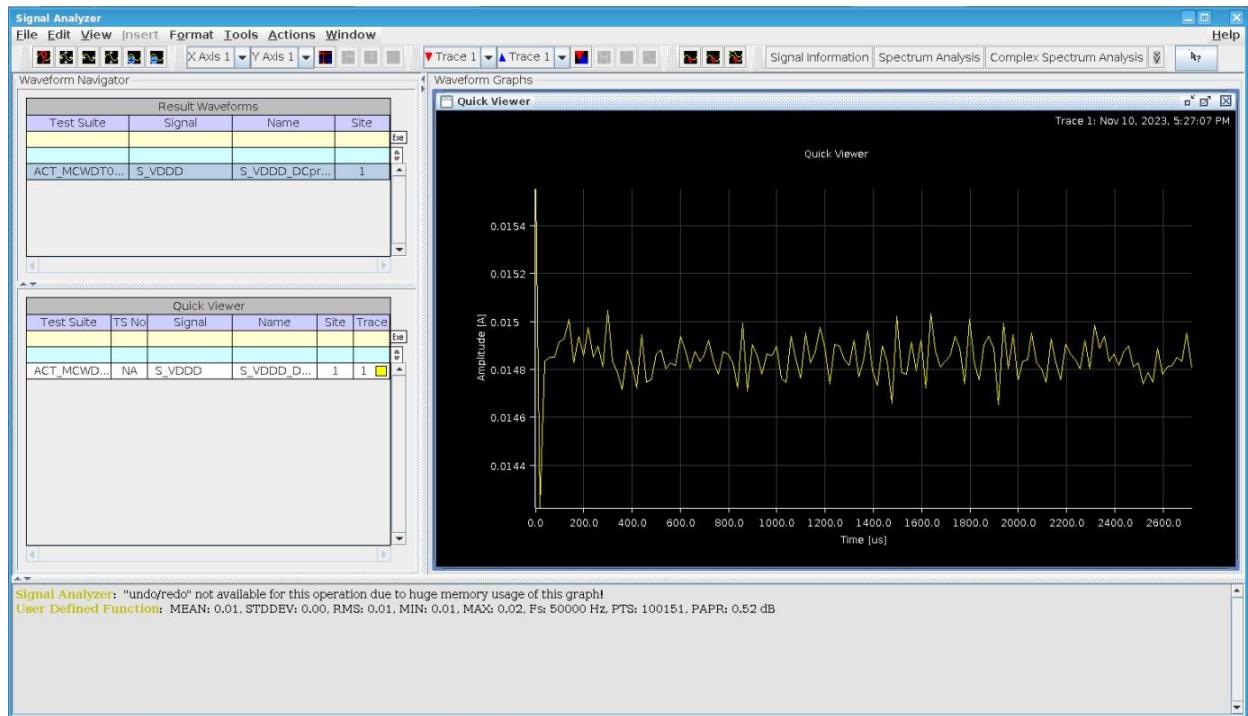


**Fig7.3g: Check Next State- here for this test case interrupt is given inside the firmware- so directly Next State is checked.**

- ✓ A Current Profiling on the main supply pin VDDD was done to capture the current consumption during the MT execution. - **Fig7.3h and Fig7.3i**

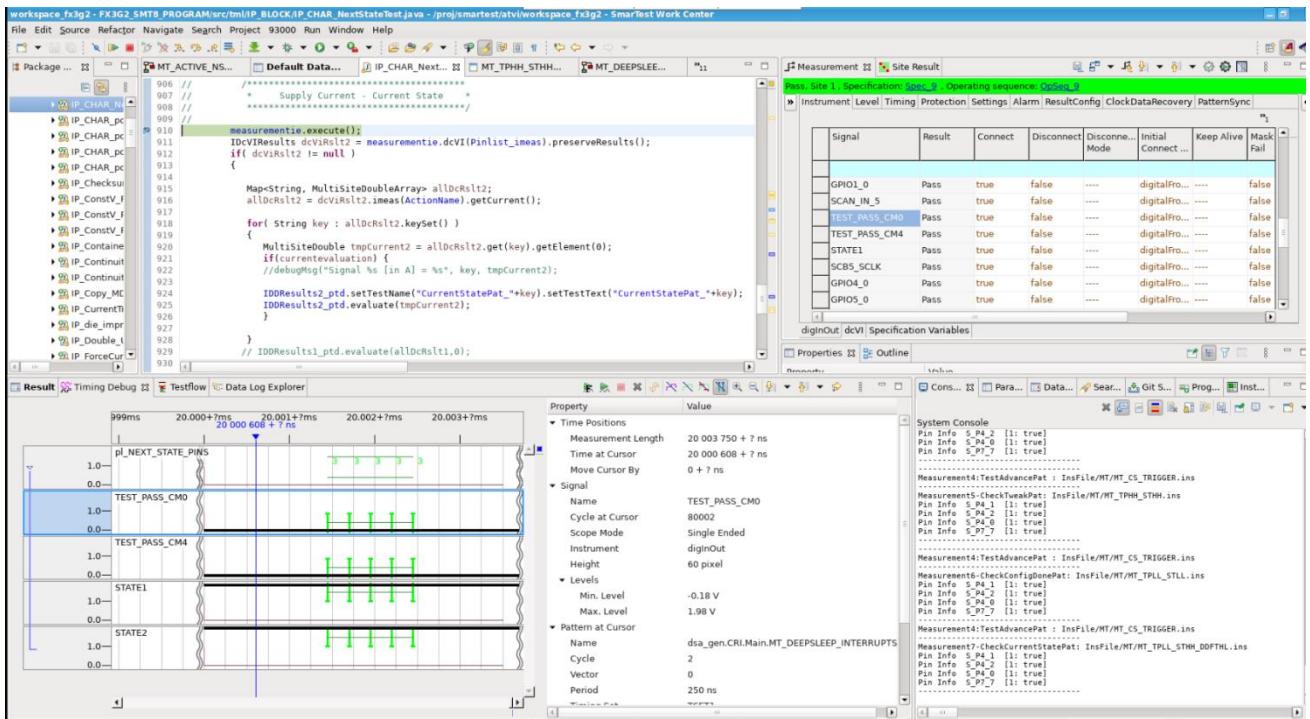


**Fig7.3h: MT Current Profiling on VDDD main supply pin**

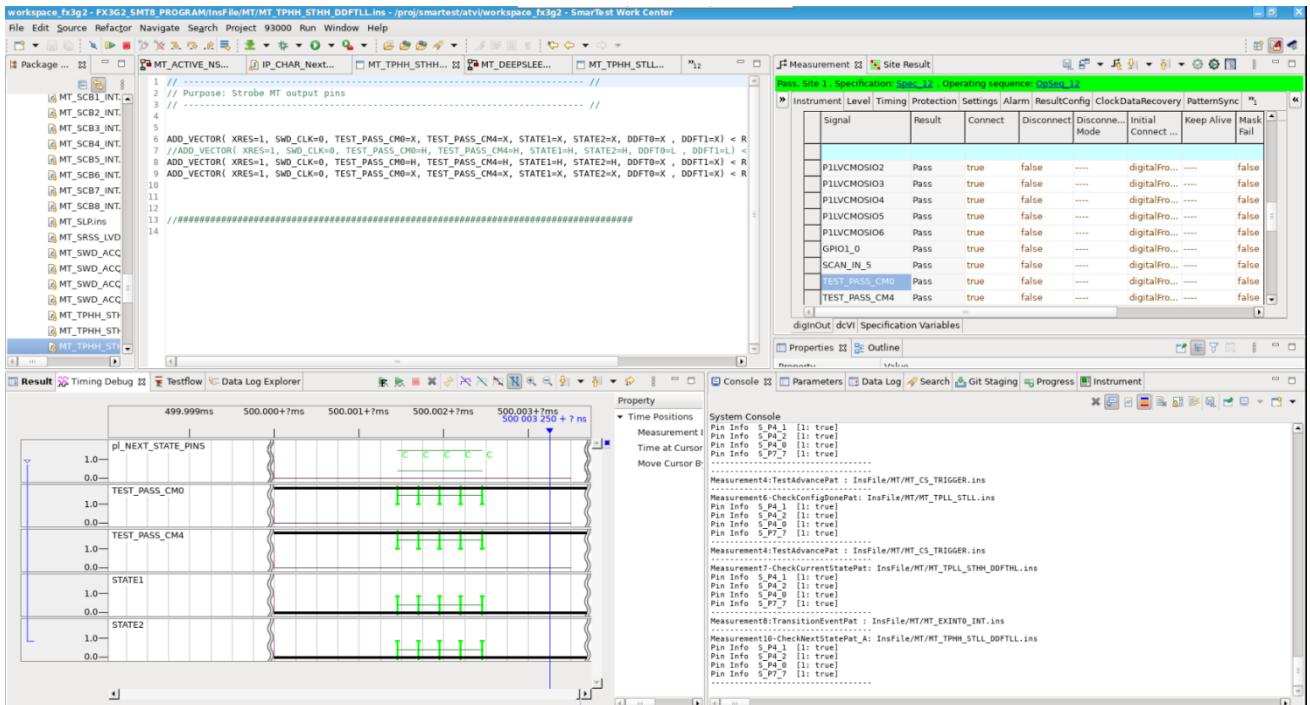


**Fig7.3i: MT Current Profiling captured live on Signal Analyzer during debug for VDDD main supply pin**

- ✓ For each test cases and each operating mode, the GPIO pins are toggled differently. And the ATE also does a current measurement on the main VDD supply pins at Initial, Current and Next States.
- Snippets for final pattern check for Deepsleep Mode-**Fig7.3j and Fig7.3k**

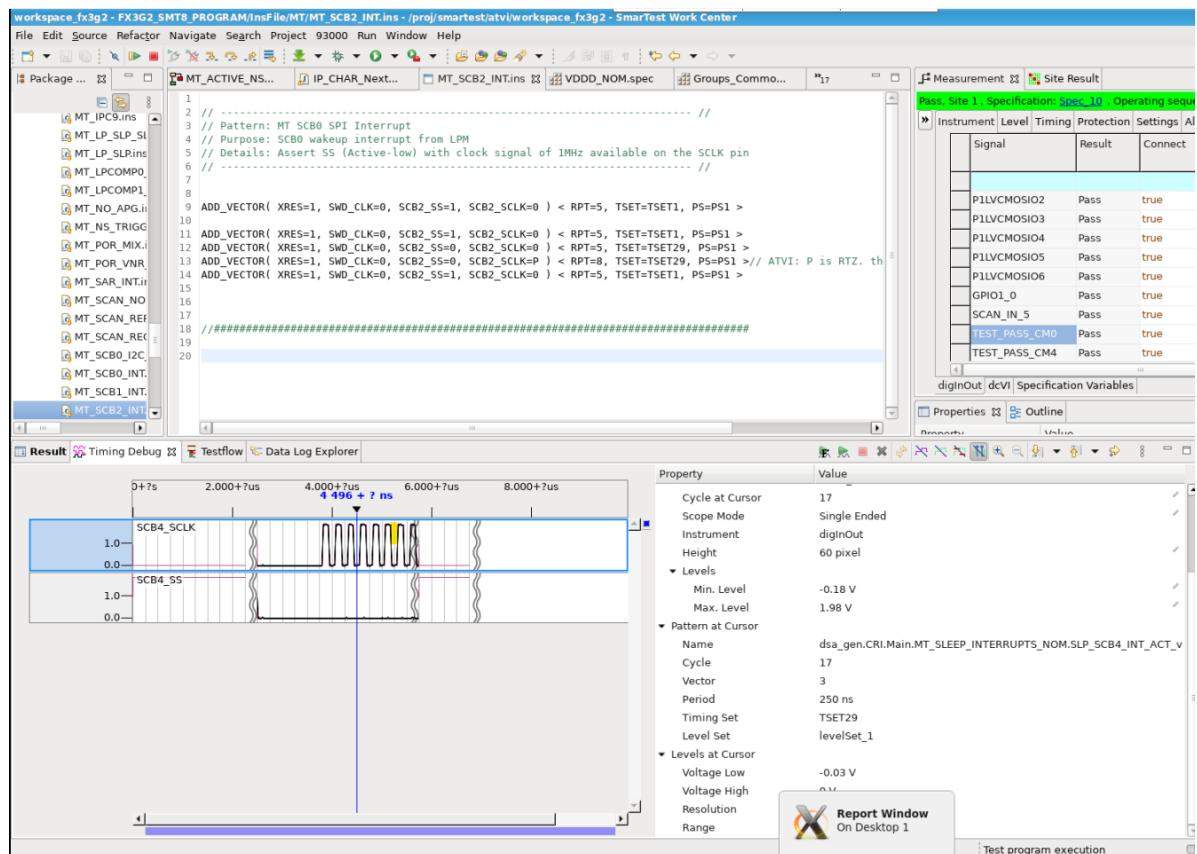


**Fig7.3j: Check Current State Pattern for Deepsleep Verification**

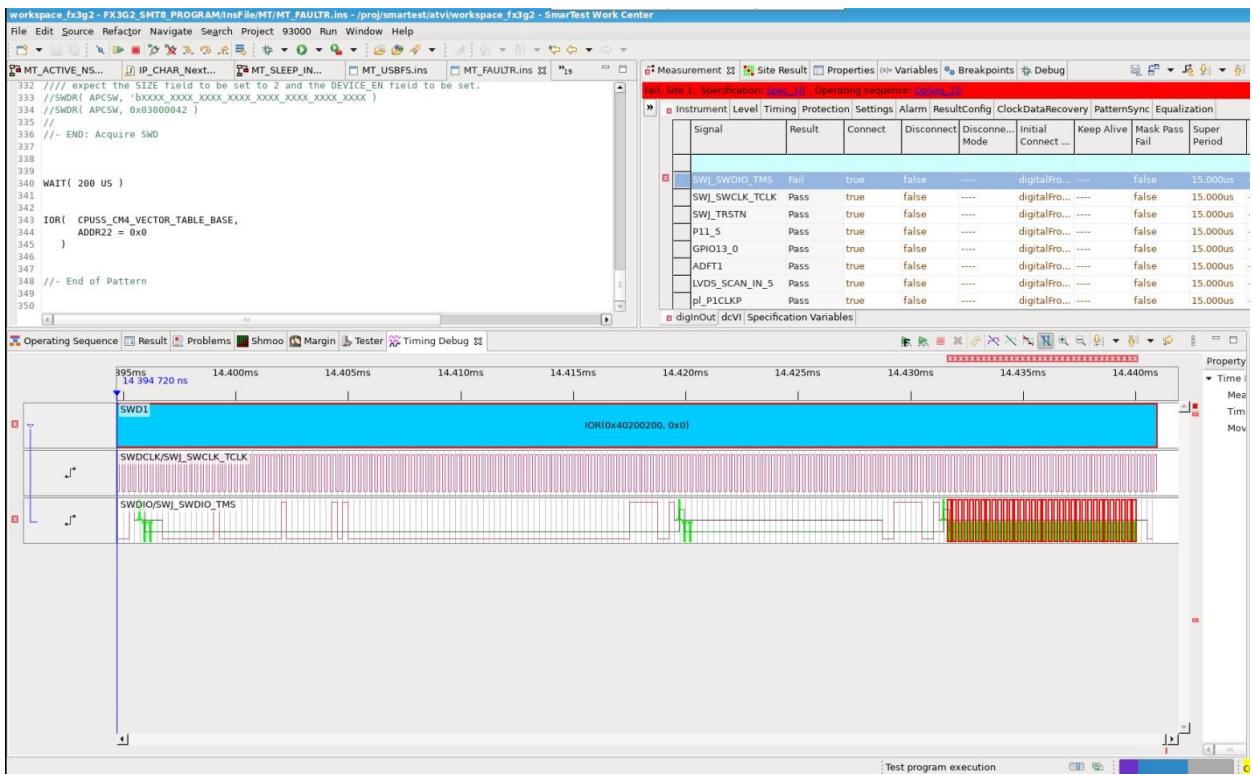


**Fig7.3k: Check Next State Pattern for Deepsleep Verification**

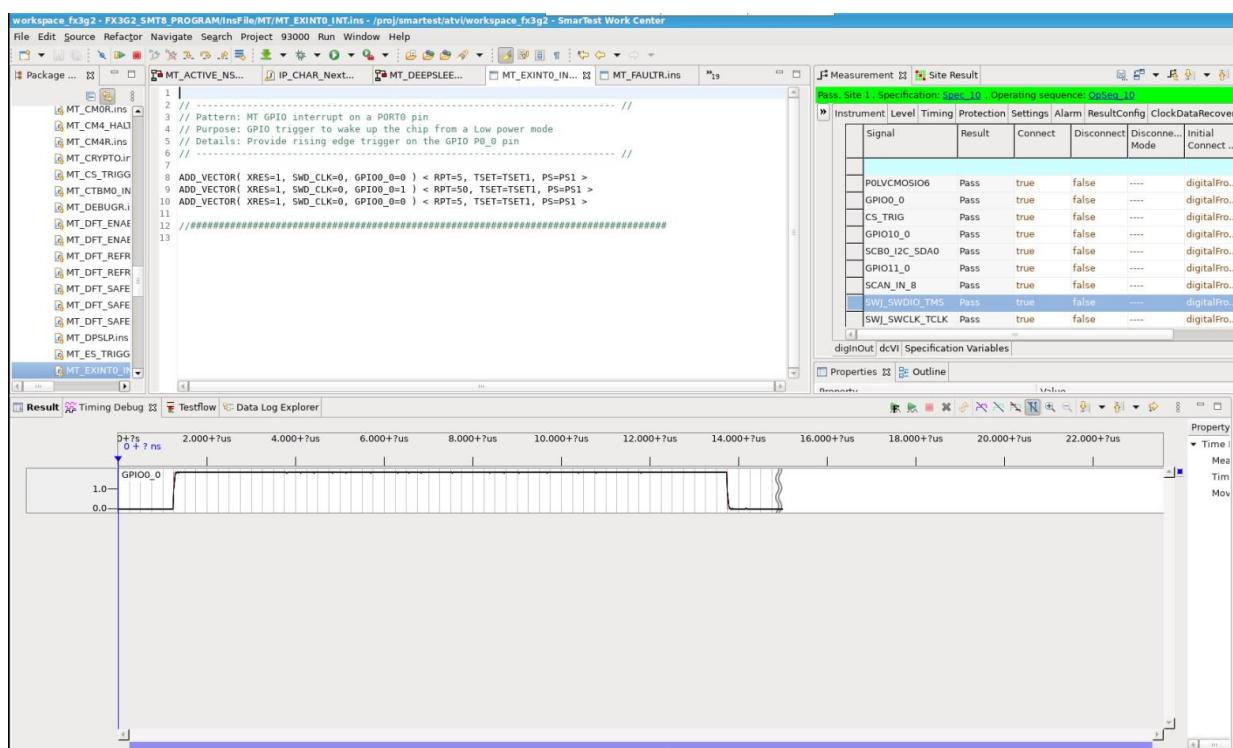
- The Fig7.3l to Fig7.3n- shows the various Interrupts provided from the ATE.



**Fig7.3l: SCB Interrupt- SPI based**



**Fig7.3m: FAULT INTERRUPT- access to the CPUSS\_CM4\_VECTOR\_TABLE\_BASE register itself causes the fault to arise and causes the interrupt**



**Fig7.3n: A pulse on GPIO pin causes an Interrupt**

- ✓ **Fig7.3o** and **Fig7.3p** shows the datalog for test cases failing(even the pattern step which fails is displayed) and passing respectively

Fully Qualified Name	Test#	Type	Signal	Low	High	Unit	Min	Max	Mean	Device#	Pass%	Fail#	Alarm#	Max. Foreground Time	Min. Fore
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMO_CTO10_INT_ACT_vNR.passOrFail_ftd	1862	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	0.0	1	0	7127.1ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMO_CTO10_INT_ACT_vNR.failingStep_ptd	1862	Parametric	@	N/A	N/A	uA	10.00...	10.00...	10.00...	1	100.0	0	0	7127.1ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMO_CTO11_INT_ACT_vNR.measurement2_ftd	1863	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	7154.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMO_CTO11_INT_ACT_vNR.initStatePat_pl_VDDD_VDDA	1863	Parametric	@	N/A	N/A	uA	14808.7	14808.7	14808.7	1	100.0	0	0	7154.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMO_CTO11_INT_ACT_vNR.CurrentStatePat_pl_VDDD_VDDA	1863	Parametric	@	N/A	N/A	uA	16453.1	16453.1	16453.1	1	100.0	0	0	7154.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMO_CTO11_INT_ACT_vNR.NextStatePat_pl_VDDD_VDDA	1863	Parametric	@	N/A	N/A	uA	17054.0	17054.0	17054.0	1	100.0	0	0	7154.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMO_CTO11_INT_ACT_vNR.passOrFail_ftd	1863	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	0.0	1	0	7154.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMO_CTO11_INT_ACT_vNR.failingStep_ptd	1863	Parametric	@	N/A	N/A	uA	10.00...	10.00...	10.00...	1	100.0	0	0	7154.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO10_INT_ACT_vNR.measurement2_ftd	1864	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	7185.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO10_INT_ACT_vNR.initStatePat_pl_VDDD_VDDA	1864	Parametric	@	N/A	N/A	uA	14816.6	14816.6	14816.6	1	100.0	0	0	7185.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO10_INT_ACT_vNR.CurrentStatePat_pl_VDDD_VDDA	1864	Parametric	@	N/A	N/A	uA	16919.6	16919.6	16919.6	1	100.0	0	0	7185.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO10_INT_ACT_vNR.NextStatePat_pl_VDDD_VDDA	1864	Parametric	@	N/A	N/A	uA	17496.8	17496.8	17496.8	1	100.0	0	0	7185.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO11_INT_ACT_vNR.failingStep_ptd	1864	Functional	@	N/A	N/A	uA	6.000...	6.000...	6.000...	1	100.0	0	0	7185.0ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO11_INT_ACT_vNR.measurement2_ftd	1865	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	7107.5ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO11_INT_ACT_vNR.initStatePat_pl_VDDD_VDDA	1865	Parametric	@	N/A	N/A	uA	14800.7	14800.7	14800.7	1	100.0	0	0	7107.5ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO11_INT_ACT_vNR.CurrentStatePat_pl_VDDD_VDDA	1865	Parametric	@	N/A	N/A	uA	16927.5	16927.5	16927.5	1	100.0	0	0	7107.5ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO11_INT_ACT_vNR.NextStatePat_pl_VDDD_VDDA	1865	Parametric	@	N/A	N/A	uA	17504.7	17504.7	17504.7	1	100.0	0	0	7107.5ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO11_INT_ACT_vNR.passOrFail_ftd	1865	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	0.0	1	0	7107.5ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_ACT_CMA_CTO11_INT_ACT_vNR.failingStep_ptd	1865	Parametric	@	N/A	N/A	uA	6.000...	6.000...	6.000...	1	100.0	0	0	7107.5ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_SLP_CMO_CTO10_INT_ACT_vNR.measurement2_ftd	1866	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	7177.3ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_SLP_CMO_CTO10_INT_ACT_vNR.initStatePat_pl_VDDD_VDDA	1866	Parametric	@	N/A	N/A	uA	14808.7	14808.7	14808.7	1	100.0	0	0	7177.3ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_SLP_CMO_CTO10_INT_ACT_vNR.CurrentStatePat_pl_VDDD_VDDA	1866	Parametric	@	N/A	N/A	uA	9891.01	9891.01	9891.01	1	100.0	0	0	7177.3ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_SLP_CMO_CTO10_INT_ACT_vNR.NextStatePat_pl_VDDD_VDDA	1866	Parametric	@	N/A	N/A	uA	13622.7	13622.7	13622.7	1	100.0	0	0	7177.3ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_SLP_CMO_CTO10_INT_ACT_vNR.passOrFail_ftd	1866	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	0.0	1	0	7177.3ms	
Main.MAIN_MT_MT_ACTIVE_INTERRUPTS_NOM_SLP_CMO_CTO10_INT_ACT_vNR.failingStep_ptd	1866	Parametric	@	N/A	N/A	uA	10.00...	10.00...	10.00...	1	100.0	0	0	7177.3ms	
Main.MAIN_MT_MT_SLEEP_NS_NOM_SLP_POR_OFF_vNR.measurement2_ftd	1148	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	4980.1ms	
Main.MAIN_MT_MT_SLEEP_NS_NOM_SLP_POR_OFF_vNR.failingStep_ptd	1148	Parametric	@	N/A	N/A	uA	0	0	0	1	100.0	0	0	4980.1ms	

**Fig7.3o: Known Failures along with failing step for MT Test Cases**

Fully Qualified Name	Test#	Type	Signal	Low	High	Unit	Min	Max	Mean	Device#	Pass%	Fail#	Alarm#	Max. Foreground Time	Min
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_SLP_CM4_RET_NOM_vNR.failingStep_ptd	1211	Parametric	@	N/A	N/A	uA	0	0	0	1	100.0	0	0	3580.3ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_SLP_CM4_OFF_NOM_vNR.measurement2_ftd	1212	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	3540.3ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_SLP_CM4_NOM_vNR.initStatePat_pl_VDDD_VDDA	1212	Parametric	@	N/A	N/A	uA	14808.7	14808.7	14808.7	1	100.0	0	0	3540.3ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_SLP_CM4_NOM_vNR.CurrentStatePat_pl_VDDD_VDDA	1212	Parametric	@	N/A	N/A	uA	9748.70	9748.70	9748.70	1	100.0	0	0	3540.3ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_SLP_CM4_NOM_vNR.NextStatePat_pl_VDDD_VDDA	1212	Parametric	@	N/A	N/A	uA	9740.79	9740.79	9740.79	1	100.0	0	0	3540.3ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_SLP_CM4_NOM_vNR.passOrFail_ftd	1212	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	3540.3ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_SLP_CM4_NOM_vNR.failingStep_ptd	1212	Parametric	@	N/A	N/A	uA	0	0	0	1	100.0	0	0	3540.3ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_DPSPL_CM4_RET_SRSS_DPSPL_vNR.measurement2_ftd	1218	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	3530.1ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_DPSPL_CM4_RET_SRSS_DPSPL_vNR.initStatePat_pl_VDDD_VDDA	1218	Parametric	@	N/A	N/A	uA	14800.7	14800.7	14800.7	1	100.0	0	0	3530.1ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_DPSPL_CM4_RET_SRSS_DPSPL_vNR.CurrentStatePat_pl_VDDD_VDDA	1218	Parametric	@	N/A	N/A	uA	561.71	561.71	561.71	1	100.0	0	0	3530.1ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_DPSPL_CM4_RET_SRSS_DPSPL_vNR.NextStatePat_pl_VDDD_VDDA	1218	Parametric	@	N/A	N/A	uA	593.341	593.341	593.341	1	100.0	0	0	3530.1ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CHO_DPSPL_CM4_RET_SRSS_DPSPL_vNR.passOrFail_ftd	1218	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	3530.1ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CHO_DPSPL_CM4_RET_SRSS_DPSPL_vNR.failingStep_ptd	1218	Parametric	@	N/A	N/A	uA	0	0	0	1	100.0	0	0	3530.1ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_DPSPL_CM4_OFF_SRSS_DPSPL_vNR.measurement2_ftd	1219	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	3571.8ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_DPSPL_CM4_OFF_SRSS_DPSPL_vNR.initStatePat_pl_VDDD_VDDA	1219	Parametric	@	N/A	N/A	uA	14800.7	14800.7	14800.7	1	100.0	0	0	3571.8ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_DPSPL_CM4_OFF_SRSS_DPSPL_vNR.CurrentStatePat_pl_VDDD_VDDA	1219	Parametric	@	N/A	N/A	uA	561.717	561.717	561.717	1	100.0	0	0	3571.8ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_DPSPL_CM4_OFF_SRSS_DPSPL_vNR.NextStatePat_pl_VDDD_VDDA	1219	Parametric	@	N/A	N/A	uA	601.247	601.247	601.247	1	100.0	0	0	3571.8ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_DPSPL_CM4_OFF_SRSS_DPSPL_vNR.passOrFail_ftd	1219	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	3571.8ms	
Main.MAIN_MT_ACTIVE_NS_NOM.CMO_DPSPL_CM4_OFF_SRSS_DPSPL_vNR.failingStep_ptd	1219	Parametric	@	N/A	N/A	uA	0	0	0	1	100.0	0	0	3571.8ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT0_INT_ACT_vNR.measurement2_ftd	1431	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	4264.5ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT0_INT_ACT_vNR.initStatePat_pl_VDDD_VDDA	1431	Parametric	@	N/A	N/A	uA	14824.5	14824.5	14824.5	1	100.0	0	0	4264.5ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT0_INT_ACT_vNR.CurrentStatePat_pl_VDDD_VDDA	1431	Parametric	@	N/A	N/A	uA	16603.4	16603.4	16603.4	1	100.0	0	0	4264.5ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT0_INT_ACT_vNR.NextStatePat_pl_VDDD_VDDA	1431	Parametric	@	N/A	N/A	uA	17093.5	17093.5	17093.5	1	100.0	0	0	4264.5ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT0_INT_ACT_vNR.passOrFail_ftd	1431	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	4264.5ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT0_INT_ACT_vNR.failingStep_ptd	1431	Parametric	@	N/A	N/A	uA	0	0	0	1	100.0	0	0	4264.5ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT1_INT_ACT_vNR.measurement2_ftd	1432	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	4257.8ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT1_INT_ACT_vNR.initStatePat_pl_VDDD_VDDA	1432	Parametric	@	N/A	N/A	uA	14808.7	14808.7	14808.7	1	100.0	0	0	4257.8ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT1_INT_ACT_vNR.CurrentStatePat_pl_VDDD_VDDA	1432	Parametric	@	N/A	N/A	uA	16579.6	16579.6	16579.6	1	100.0	0	0	4257.8ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT1_INT_ACT_vNR.NextStatePat_pl_VDDD_VDDA	1432	Parametric	@	N/A	N/A	uA	17077.7	17077.7	17077.7	1	100.0	0	0	4257.8ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT1_INT_ACT_vNR.passOrFail_ftd	1432	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	4257.8ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT1_INT_ACT_vNR.failingStep_ptd	1432	Parametric	@	N/A	N/A	uA	0	0	0	1	100.0	0	0	4257.8ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT2_INT_ACT_vNR.measurement2_ftd	1433	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	4300.6ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT2_INT_ACT_vNR.initStatePat_pl_VDDD_VDDA	1433	Parametric	@	N/A	N/A	uA	14792.8	14792.8	14792.8	1	100.0	0	0	4300.6ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT2_INT_ACT_vNR.CurrentStatePat_pl_VDDD_VDDA	1433	Parametric	@	N/A	N/A	uA	16579.6	16579.6	16579.6	1	100.0	0	0	4300.6ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT2_INT_ACT_vNR.NextStatePat_pl_VDDD_VDDA	1433	Parametric	@	N/A	N/A	uA	17069.8	17069.8	17069.8	1	100.0	0	0	4300.6ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT2_INT_ACT_vNR.passOrFail_ftd	1433	Functional	@	N/A	N/A	N/A	N/A	N/A	N/A	1	100.0	0	0	4300.6ms	
Main.MAIN_MT_ACTIVE_INTERRUPTS_NOM_ACT_TCPWM0_CNT2_INT_ACT_vNR.failingStep_ptd	1433	Parametric	@	N/A	N/A	uA	0	0	0	1	100.0	0	0	4300.6ms	

**Fig7.3p: Passing MT Test Cases**

## **8.CONCLUSION**

- ✓ The Test Program for Mode Transition and Power on Reset Validation were developed from Scratch and validated Offline to catch Logical, Build and Run Time errors.
- ✓ The Test Program verification was done on both Wafer Sort and Packed units for the specific test cases.
- ✓ The stability and quality of the test program was validated across multiple devices and across PVT (Process, Voltage and Temperatures).
- ✓ During the validation process to qualify the results, devices were validated on Bench, and lab equipment's like Digital Multimeter, Oscilloscope were used.
- ✓ Automation scripts were also developed to reduce repetitive work moving further in terms of test program development and reduce device bring up turnaround time.
- ✓ Automation was even done to Temperature Forcing units to characterize a device across temperatures without human intervention in-between and data logs collection was also automated.
- ✓ The Automatic Test Equipment features were thoroughly explored for its capabilities and debug features to bring up the device faster and efficiently.
- ✓ Final Results were reviewed with Design Engineer to make sure the Datasheet specs are met properly.

## 9.APPENDICES

- ✓ Snippets shows parts of JAVA ATE Test Program Code Written.

```

197 //CALLINSRLE(pn_, TESTADVANCEPAT); //cs_trig for firmware
198 ds2.sequentialEnd();
199 ph2.setMeasSetups(measurement2);
200
201 IDeviceSetup ds2a = createDeviceSetup();
202 setupBaseImport(ds2a, measurement2a);
203 ph2a = createProtocolHelper(ds2a, UseSwd2);
204 TestID2=TestID0 | 0x1;
205 ds2a.sequentialBegin();
206 callInsFile(ph2a, NODFTACOPattern);
207 DirectInsComposer_atvi dca = new DirectInsComposer_atvi (ph2a);
208 dca.IOW( sramIdentifieraddress, 0x00000000);
209 dca.IOW( sram0, TestID2 );
210 dca.IOW( sram1, TestID1 );
211 dca.IOW( sramIdentifieraddress, sramIdentifier);
212 callInsFile(ph2a, XRES_Toggle);
213 ds2a.waitCall( BootUpDelay );
214 callInsFile(ph2a, TestAdvancePat); //cs_trig for firmware
215 ds2a.sequentialEnd();
216 ph2a.setMeasSetups(measurement2a);
217
218 IDeviceSetup ds3 = createDeviceSetup();
219 ph3 = createProtocolHelper(ds3, UseSwd2);
220 setupBaseImport(ds3, measurement3);
221 ds3.sequentialBegin();
222 //ds3.waitCall( BootUpDelay );
223 callInsFile(ph3, CheckInitialStatePat);
224 ds3.sequentialEnd();
225 ph3.setMeasSetups(measurement3);
226
227 IDeviceSetup ds4 = createDeviceSetup();
228 ph4 = createProtocolHelper(ds4, UseSwd2);
229 setupBaseImport(ds4, measurement4);
230 ds4.sequentialBegin();
231 callInsFile(ph4, TestAdvancePat);
232 ds4.sequentialEnd();
233 ph4.setMeasSetups(measurement4);
234
235 IDeviceSetup ds5 = createDeviceSetup();
236 ph5= createProtocolHelper(ds5, UseSwd2);
237 setupBaseImport(ds5, measurement5);
238 ds5.sequentialBegin();
239 ds5.waitCall( TweakDelay );
240 callInsFile(ph5, CheckTweakPat);
241

```

**Fig9a: Part of Mode Transition ATE Test Program JAVA Code**

```

102 arr_supply_pins= PowerSupplyPin.replace("*", ",");
103 arr_supply_pins= PowerSupplyPin.split(",");
104 String rampname = TestCase;
105 String rampfile = "RampBased"+TestCase+".cwf";
106 ISetupWaveformSampleBased wfx = ds1.addWaveformSampleBased(rampname).setSampleRate(TweakCyclekhz);
107 wfx.setDatafile(rampfile);
108 for (int i =0; i<arr_supply_pins.length; i++)
109 {
110
111 ISetupDcVI dcSetup = ds1.addDcVI(arr_supply_pins[i]);
112 ISetupDcVI.ISourceWaveform vfrcx = dcSetup.sourceWaveform();
113 vfrcx.setWaveform(wfx).setForceType(ISetupDcVI.ISourceWaveform.SetupForceType.VOLTAGE).setIrange(200e-3).setIclamp(200e-3);
114 ds1.actionCall(vfrcx);
115
116 }
117 measurement1.setSetups(ds1);
118
119 IDeviceSetup ds2 = createDeviceSetup();
120 ph2 = createProtocolHelper(ds2);
121 setupBaseImport(ds2, measurement2);
122 ds2.sequentialBegin();
123 callInsFile(ph2, XRESPattern);
124 ds2.waitCall( NoDelayTime );
125 callInsFile(ph2, BootstatusPattern);
126 ds2.sequentialEnd();
127 ph2.setMeasSetups(measurement2);
128
129 }
130
131 @Override
132 public void execute()
133 {
134
135 logParameter(this.getClass());
136 MultiSiteBoolean swdPass = null;
137 MultiSiteBoolean fcntl = new MultiSiteBoolean();
138 MultiSiteLong porResultVmapL = new MultiSiteLong(statusType.POR_STATUS_DEFAULT.value);
139
140 measurement1.execute(); // ATVI: run POR ramp
141 measurement2.execute(); // ATVI: BootStatus Pattern Check
142
143 debugMsg("***** porResultVmap "+porResultVmapL);
144 ftd.evaluate(measurement2.hasPassed());
145 debugMsg("***** measurement3 BootStatusPattern "+measurement2.hasPassed());
146

```

**Fig9b: Part of Power on Reset ATE Test Program JAVA Code**

## **10. REFERENCES**

- ✓ Matthew Knowles, Marc Hütner: Simplifying Silicon Bring-Up and Debug on Automatic Test Equipment With ATE-Connect, White Paper, Siemens
- ✓ Ashish Gupta; Gaurav Verma: Bridging Validation and Automatic Test Equipment (ATE) Environment. In Proceedings of International Symposium on Electronic System Design (ISED), 2012
- ✓ Advantest Technical Documentation Center  
<https://cs.advantest.com/V93000/help/index.jsp#topic/com.verigy.itee.help.smartest.ui.8.6.1/247382.htm>
- ✓ **Datasheet1-**  
[https://www.infineon.com/dgdl/Infineon-CYUSB301X\\_CYUSB201X\\_EZ-USB\\_FX3\\_SUPERSPEED\\_USB\\_CONTROLLER-Datasheet-v21\\_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0eca1e7442aa](https://www.infineon.com/dgdl/Infineon-CYUSB301X_CYUSB201X_EZ-USB_FX3_SUPERSPEED_USB_CONTROLLER-Datasheet-v21_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0eca1e7442aa)
- ✓ **Datasheet2**  
[https://www.infineon.com/dgdl/Infineon-PSoC\\_6 MCU CY8C63x6 CY8C63x7 Datasheet\\_PSoC\\_63 MCU with Bluetooth LE-Datasheet-v19\\_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ee4efe46c37](https://www.infineon.com/dgdl/Infineon-PSoC_6 MCU CY8C63x6 CY8C63x7 Datasheet_PSoC_63 MCU with Bluetooth LE-Datasheet-v19_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ee4efe46c37)

## **11.GLOSSARY**

<b>Abbreviation</b>	<b>Meaning</b>
ATE	Automatic Test Equipment
POR	Power on Reset
MT	Mode Transition
DUT	Device Under Test
SWD	Arm Serial Wire Debug
IC	Integrated Circuits

### **Checklist of Items for the Final Dissertation / Project / Project Work Report**

This checklist is to be attached as the last page of the final report.

**This checklist is to be duly completed, verified and signed by the student.**

1.	<b>Is the final report neatly formatted with all the elements required for a technical Report?</b>	Yes
2.	Is the Cover page in proper format as given in Annexure A?	Yes
3.	Is the Title page (Inner cover page) in proper format?	Yes
4.	(a) Is the Certificate from the Supervisor in proper format? (b) Has it been signed by the Supervisor?	Yes Yes
5.	Is the Abstract included in the report properly written within one page? Have the technical keywords been specified properly?	Yes Yes
6.	Is the title of your report appropriate? <b>The title should be adequately descriptive, precise and must reflect scope of the actual work done.</b> Uncommon abbreviations / Acronyms should not be used in the title	Yes
7.	Have you included the List of abbreviations / Acronyms?	Yes
8.	Does the Report contain a summary of the literature survey?	Yes
9.	Does the Table of Contents include page numbers? (i). Are the Pages numbered properly? (Ch. 1 should start on Page # 1) (ii). Are the Figures numbered properly? (Figure Numbers and Figure Titles should be at the bottom of the figures) (iii). Are the Tables numbered properly? (Table Numbers and Table Titles should be at the top of the tables) (iv). Are the Captions for the Figures and Tables proper? (v). Are the Appendices numbered properly? Are their titles appropriate	Yes Yes Yes Yes Yes
10.	Is the conclusion of the Report based on discussion of the work?	Yes
11.	Are References or Bibliography given at the end of the Report? Have the References been cited properly inside the text of the Report? Are all the references cited in the body of the report	Yes Yes Yes
12.	Is the report format and content according to the guidelines? The report should not be a mere printout of a PowerPoint Presentation, or a user manual. Source code of software need not be included in the report.	Yes

#### **Declaration by Student:**

I certify that I have properly verified all the items in this checklist and ensure that the report is in proper format as specified in the course handout.



**Place: BANGALORE**

**Signature of the Student**

**Date: 13<sup>th</sup> November 2023**

**Name: A Vipula  
ID No.: 2021HT80530**