

“DDR TEST FRAMEWORK”

Project Report

*Submitted in Partial Fulfillment of the
Requirements for the Degree of*

MASTER OF TECHNOLOGY

IN

EMBEDDED SYSTEM

ELECTRONICS AND COMMUNICATION ENGINEERING

By

**Saumya Oza
(19MECE11)**



**Department of Electronics and Communication Engineering,
Institute of Technology,
Nirma University,
Ahmedabad 382 481**

May 2021

CERTIFICATE

This is to certify that the Major Project Report entitled “**DDR TEST FRAMEWORK**” submitted by **Oza Saumya Gaurangbhai (19MECE11)** towards the partial fulfillment of the requirements for the award of degree in Master of Technology in the Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. To the best of our knowledge, the findings of this major project work have not been sent to any other university or institution for the award of any degree or diploma.

Date:18/05/2021

Industry Guide

Mr. Biswaprakash Navajeevan

Engineer, Staff

Qualcomm India Private Limited, Hyderabad

Institute Guide

Dr. Manisha A. Upadhyay

Associate Professor

Department of Electronics & Communication Engineering,

Institute of Technology, Nirma University, Ahmedabad

Program Coordinator,

Dr. Nagendra Gajjar

Professor & PG Coordinator

M. Tech Embedded Systems

Department of Electronics & Communication Engineering

Institute of Technology, Nirma University

Head of Department

Dr. Dhaval Pujara

Department of Electronics & Communication Engineering,

Institute of Technology, Nirma University, Ahmedabad

Undertaking for Originality of the Work

I, **Oza Saumya Gaurangbhai**, Roll No.19MECE11, certify that the Major Project named "**DDR TEST FRAMEWORK**" submitted by me in partial fulfilment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad, is my original work and that no plagiarism attempt has been made. I understand that if any correlation is discovered later with any other published work or project study, it will result in serious disciplinary action.

Date: 18/05/2021

Place: Ahmedabad

Endorsed by:

Dr. Manisha A. Upadhyay

Acknowledgement

It is my duty to humbly remember, acknowledge, and respect all those who put in their heart and hard work which went into this work. This report is the result of work whereby I have been accompanied and supported by many people. It's a nice bonus that I now have the ability to thank each of them individually.

It is a pleasure to be associated with Qualcomm, and I would like to thank my manager Biswaprakash Navajeevan, mentor Raveesh Sinha and the DDR team for their excellent guidance, advice, invaluable suggestions, and continuous encouragement at all the stages of my work.

With immense pleasure, I express my sincere gratitude, regards, and thanks to my Guide Dr. Manisha Upadhyay for her guidance, and encouragement for this project. I have been fortunate to have her as my guide as she has been a great influence on me, both as a person and as a professional.

I am obliged to Dr. Nagendra Gajjar (Professor & PG Coordinator), M. Tech Embedded System, and Dr. Dhaval Pujara (Head of Department), Department of Electronics and Communication, Nirma University for providing me this opportunity and for rendering their invaluable time and knowledge towards helping us during the entire course of our training.

Oza Saumya Gaurangbhai

ABSTRACT

Qualcomm, which pioneered 3G and 4G, is now leading the world into 5G. Qualcomm has created semiconductor components and applications for automobiles, watches, tablets, wireless networks, smartphones, the Internet of Things (IoT), and other devices. And every computer necessitates the use of memory. The DDR Test Framework suggests developing a variety of tools to test the DDR interface's stability. DDR Test Framework is primarily a DDR memory stress test tool that executes on any processor the tool supports. DDR Test Framework is a bare-metal test that is loaded into the target system and executed using T32 CMM scripts. It does not have any operating system dependencies. However, it does use libraries from the build environment for manipulating the application processor cache subsystem, timing and interrupt services, standard input output (printf) services and others. It executes a set of predefined memory tests with the purpose of establishing DDR interface stability. It can also be used to establish the stability of different memories.

INDEX

Chapter No.	Title	Page No.
	Acknowledgement	i
	Abstract	ii
	Index	iii
	List of Figures	iv
	Nomenclature	viii
1	Introduction	
	1.1 Company Profile	1
	1.2 Group Profile	2
	1.3 Gantt Chart	2
2	Literature Review	
	2.1 Trainings Undergone	4
3	Hardware	
	3.1 Lauterbach Hardware Debugger	5
4	Software	
	4.1 DDR Test Framework	7
	4.2 Trace32 Utility	8
	4.3 DART Tool	10
5	Work Done	11
6	Future Work	13
	References	14

LIST OF FIGURES

Figure 1 Gantt Chart	2
Figure 2 Power Debug Module.....	5
Figure 3 Trace32 Tool	9
Figure 4 DART Tool.....	10

NOMENCLATURE

QCT	Qualcomm CDMA Technologies
QTL	Qualcomm Technology Licensing
QSL	Qualcomm Strategic Initiatives
DDR	Double/Dual Data Rate
CDMA	Code Division Multiple Access
TD-SCDMA	Time Division-Synchronous Code Division Multiple Access
WCDMA	Wideband Code Division Multiple Access

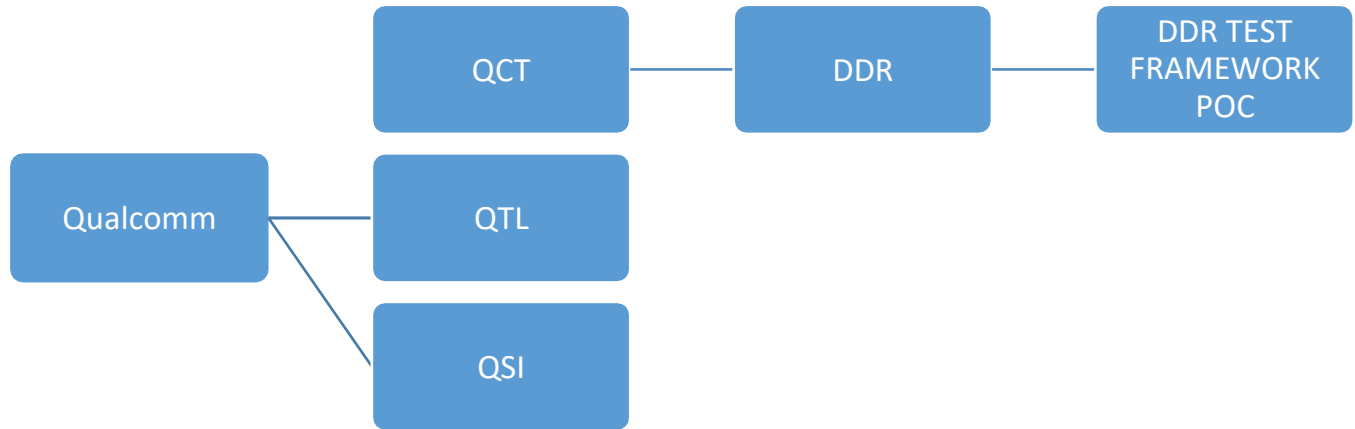
Chapter 1

Introduction

1.1 Company Profile

- For the purpose of creating “Quality Communication”, seven individuals met in San Diego in 1985 and It is incorporated on 15th August 1991 as Qualcomm Incorporated. Qualcomm is a wireless technology company. Qualcomm headquartered in San Diego, California is an American public multinational corporation. It provide services related to wireless technology and creates software, Ips(Intellectual property), Processors, etc. It owns patents critical to the CDMA2000, TD-SCDMA and WCDMA mobile communications standards.
- Today Qualcomm is spread across 40 countries and more than 170 offices. The idea, however, remains the same. By selling satellite communication systems for commercial trucks, they gathered the fund for their research in CDMA wireless cell-phone technology. With Qualcomm's CDMA patents incorporated the 3G standard was adopted.
- Over the years, With Qualcomm’s fabless manufacturing model, Qualcomm has started selling semiconductor products. Qualcomm has also developed watches, vehicles, laptops, smartphones, Internet of Things (IoT), wi-fi, and other devices semiconductor components or software.
- Now a days, Qualcomm is involved in the development of 5G (fifth-generation) and also engaged in launch and expansion of 5G (fifth-generation). The Company operates through three segments: QCT (Qualcomm CDMA Technologies) segment, QSI (Qualcomm Strategic Initiatives) segment and QTL (Qualcomm Technology Licensing) segment. Qualcomm is specialized in offering technologies like 3G (third-generation), 4G (fourth-generation) and 5G (fifth-generation) wireless technologies. These technologies and products offered by Qualcomm is used in broadband gateway, network equipment, consumer electronic devices, mobile devices and other wireless and connected devices.

1.2 Group Profile



1.2 Gantt Chart

Project – I

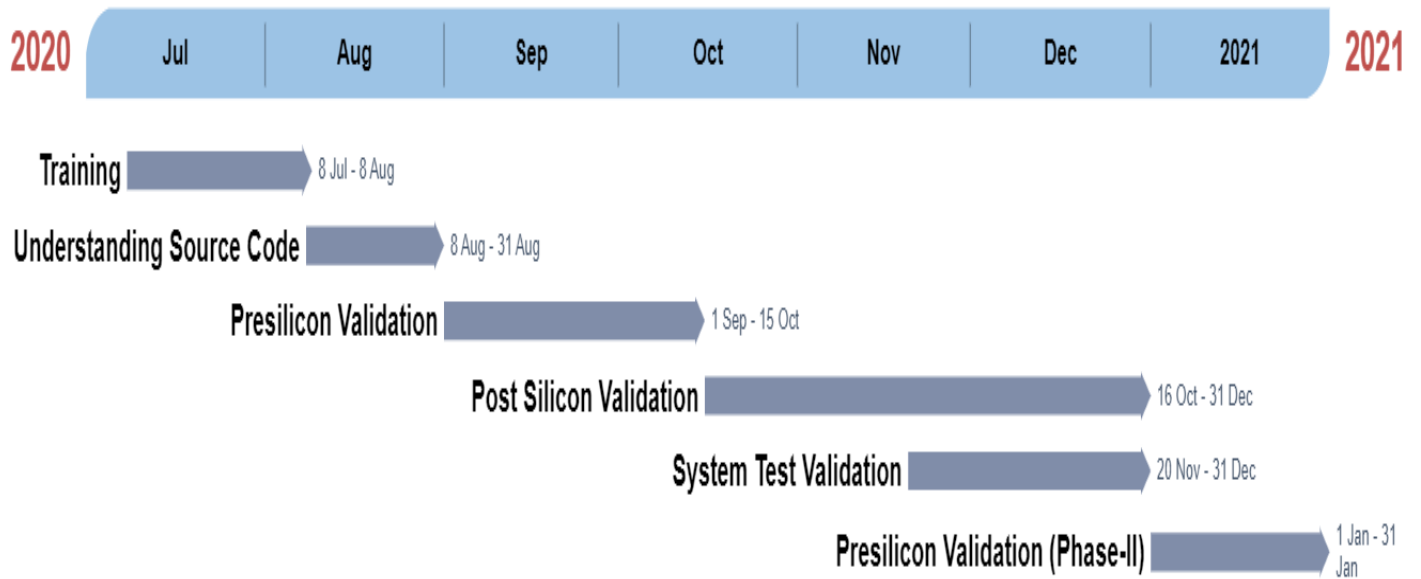
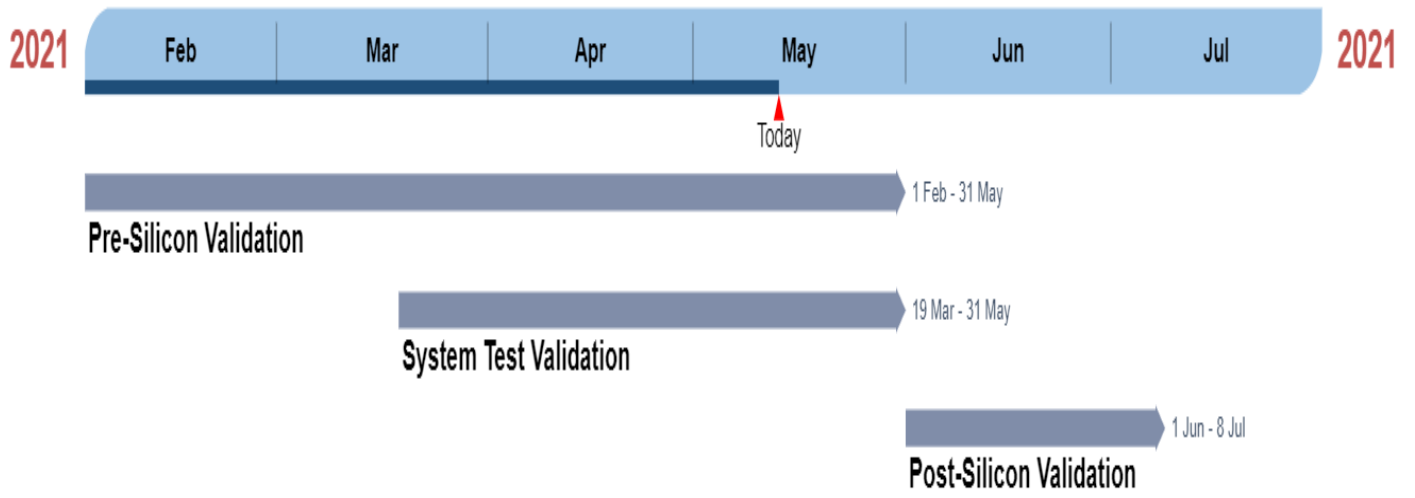


Figure 1 Gantt Chart

Project – II



Chapter 2

Literature Review

2.1 Trainings Undergone

- As a prerequisite of the project I've went through some trainings listed below
 - C/C++
 - Lauterbach Overview Training
- C/C++ training by Mr. Dan sakes is very insightful. During this training I've learnt concepts such as how to use Preprocessor effectively, new and misunderstood features of C, Use of Inline functions, Static and Dynamic Initialization etc.
- Lauterbach Overview Training give insight of the features of the hardware debugger and it's tracing feature. This training also explains the Practice scripts and the Trace32 tool. Training explains how to start T32, load code on target and then how to debug using the features provided by the tool.
- Then I've understood the source code of DDR Test Framework and the flow of the code with the help of some sessions by the manager and DDR team.

Chapter 3

Hardware

3.1 Lauterbach Hardware Debugger

- Lothar Lauterbach founded Lauterbach company in 1979. Since 1979, Lauterbach has been producing world class debuggers and real time trace. Lauterbach is the world's largest manufacturer of complete, modular and upgradeable microprocessor development tools. The company has grown into the global market leader for hardware supported debug tools, having already developed itself as a technology leader.

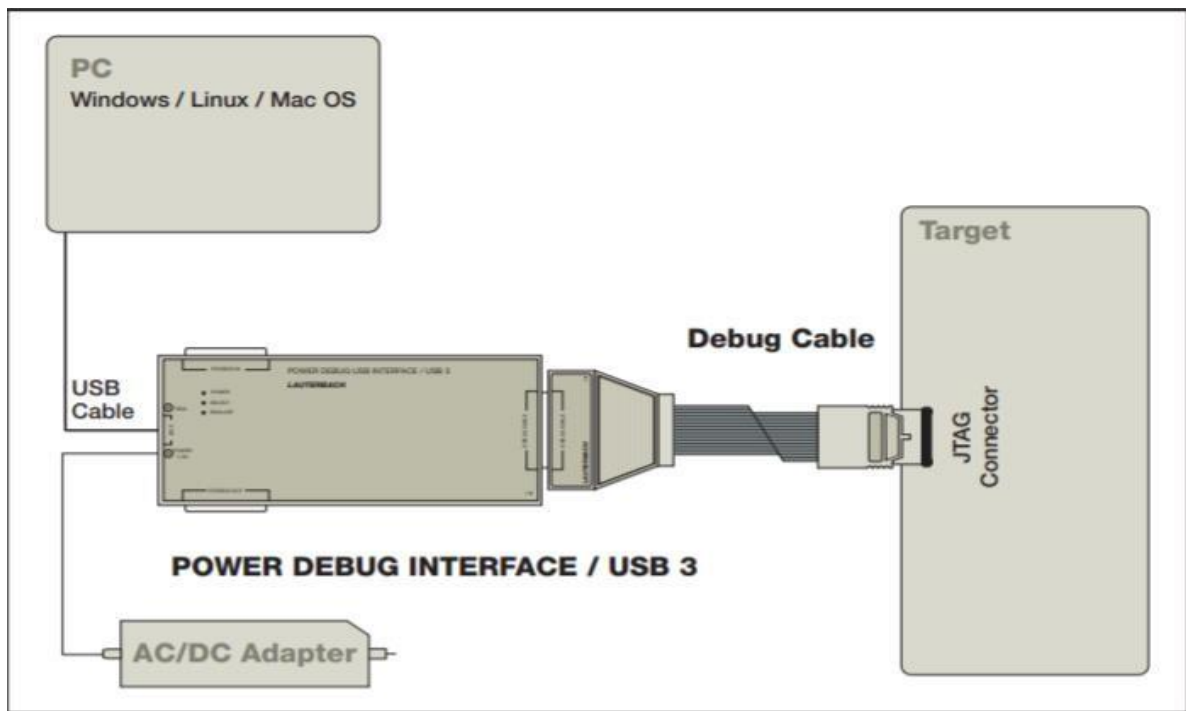


Figure 2 Power Debug Module

- Lauterbach is a globally recognized organization with blue-chip clients from all corners of the globe. It is present in every country at the cutting edge of embedded design, and its international distribution, marketing, and support operations are coordinated from its headquarters in Germany. The corporation has branches in the United Kingdom, France, Italy, Japan, China, and the United States' east and west coasts. Lauterbach works with competent partner companies in all other countries and has long-standing relationships with the majority of them, ensuring global supply and support for Lauterbach goods.

- Lauterbach is a German electronic design automation company that specializes in debugging embedded systems with in-circuit emulators and logic analyzers. Lauterbach Datentechnik GmbH changed its name to Lauterbach GmbH in 2009.
- The company sells in-circuit emulators under the TRACE32 brand, which are used mostly on systems that use JTAG. With over 80000 production seats sold, it is the global market leader in hardware supported debugging software for embedded systems.
- It supports embedded debuggers, software and hardware trace and logic analyzer systems for nearly 100 cores deployed in over 5000 chip families such as ARC, Arm Cortex-A/-R/-M, RISC-V, Power Architecture, TriCore, RH850, Xtensa, and others.
- Some of the key Features of the Debugger are as below
 - Multicore Debugging
 - Peripheral Browser
 - HLL Debugger
 - MMU Support
 - OS-aware Debugging
 - Android Debugging

Chapter 4

Software

4.1 DDR Test Framework

- DDR Test Framework is primarily a DDR memory stress test tool that executes on any processor the tool supports. DDR Test Framework is a bare-metal test that is loaded into the target system and executed using T32 CMM scripts. It does not have any operating system dependencies. However, it does use libraries from the VI build environment for manipulating the application processor cache subsystem, timing and interrupt services, stdout (printf) services and others. It executes a set of predefined memory tests with the purpose of establishing DDR interface stability. It can also be used to establish the stability of different memories.
- There are several sub-tests inside DDR Test Framework. Each of the tests sends out a particular data pattern over a user-supplied target memory range, and verifies the patterns by reading them back and comparing with data that was written.
- Verify failures are displayed on the T32 debugger term.gate window and are constructed to be debug friendly with information about the failing memory location, expected data, observed data, and the XOR value of the two.
- Several user-configurable knobs allow the user to set the operational behavior of DDR Test Framework per debug needs, such as target range start and end, which sub-tests to run, how many iterations to run the tests for, and others that are detailed in later sections.
- This Framework comes with several tools such as FreqSw, Shmoo, Power etc. Each tool serves a different purpose.
- The DDR frequency switching tool supports pseudo random switching between specified frequencies while running traffic generators.
- The power tool supports collecting data for DDRSS comparisons with UPM use-case estimates, DDR_PHY (including DDR_CC) model correlation, and BIMC model correlation.
- The shmoo tool supports sweeping some parameters (such as Vref, RD-WR CDC Delay) in an N-dimensional shmoo at a specified DDR frequency while running traffic generators.

4.1.1 Configuration parameters

- Range Start : Start of the target memory range over which the tests will be run.
- Range End : End of the target memory range over which the tests will be run.
- Access Pattern : Bit mask of the access modes that each pattern should be tested with.
- Requested Iterations : Number of iterations that each pattern should execute.
- Error Handling : Bit mask of whether the test should stop or continue upon failure.
- Result Code : Indicates the Test state i.e. pass/fail.
- Retry Count : Number of times to retry a read that fails
- User Defined Pattern Size : Size of user-defined pattern.
- Requested Tests : Indicates which all tests to be executed.
- Requested SuperIterations : Number of iterations that the entire test suite will be executed.

4.2 Trace32 Utility

- Trace32 is a debugging tool for the Lauterbach debugger. The tool is used for debugging, and it includes several features that make the debugging process quick and fast. Debugging is done with the PRACTICE script. PRACTICE is a line-oriented test language that was first created for in-circuit emulators in 1984 and can now be used to solve many of the common problems in digital measurement engineering. The test language enables interactive script creation with the ability to quickly remove errors and execute scripts right away. PRACTICE testing script execution can be paused and resumed at any time. PRACTICE includes a valuable concept for dealing with script variables and order parameters. This macro concept enables parameter substitution at any point within the commands. Conflicts between target program names are eliminated because PRACTICE variables can only appear as PRACTICE macros.
- Some of the Feature provided by Trace32 are as below
 - Read/write access to registers
 - Read/write access to memories
 - Start/stop of program execution
 - on-chip breakpoints
 - Read/write access to memory while the program execution is running
 - Additional features as benchmark counters, triggers etc.

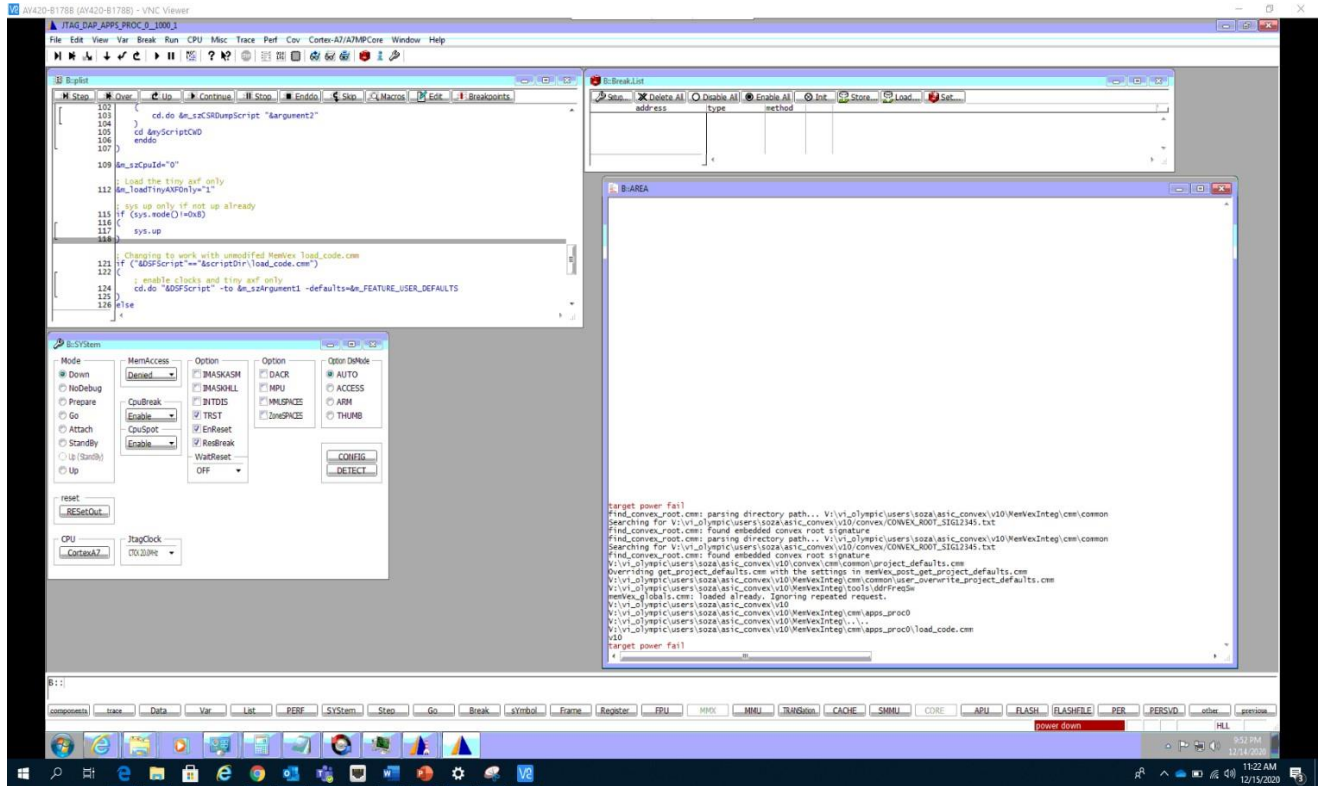


Figure 3 Trace32 Tool

- Some Features of PRACTICE scripts in Trace32 are as
 - Execute commands right on start of the debugger
 - Customize the TRACE32 PowerView user interface to your project requirements
 - Set up the debugger with settings for the target board
 - Standardize repetitive and complex actions
 - Initialize the target
 - Load the application and / or the symbols
 - Add your own and extend available features
 - Speed up debugging through automation
 - Make debug actions reproducible for verification purposes and regression test

4.3 DART Tool

- DART (DDR Analysis and Release Tool) is a Python based tool intended to automate test development and execution processes that are done on a daily basis.
- The automated processes are:
 - Workspace creation and source code population onto a common network path
 - Various types of builds related to overnight testing and code release, which includes source code analysis to be included in release notes
 - Source code diffing analysis on code delivery to the software team
 - Overnight test report generation

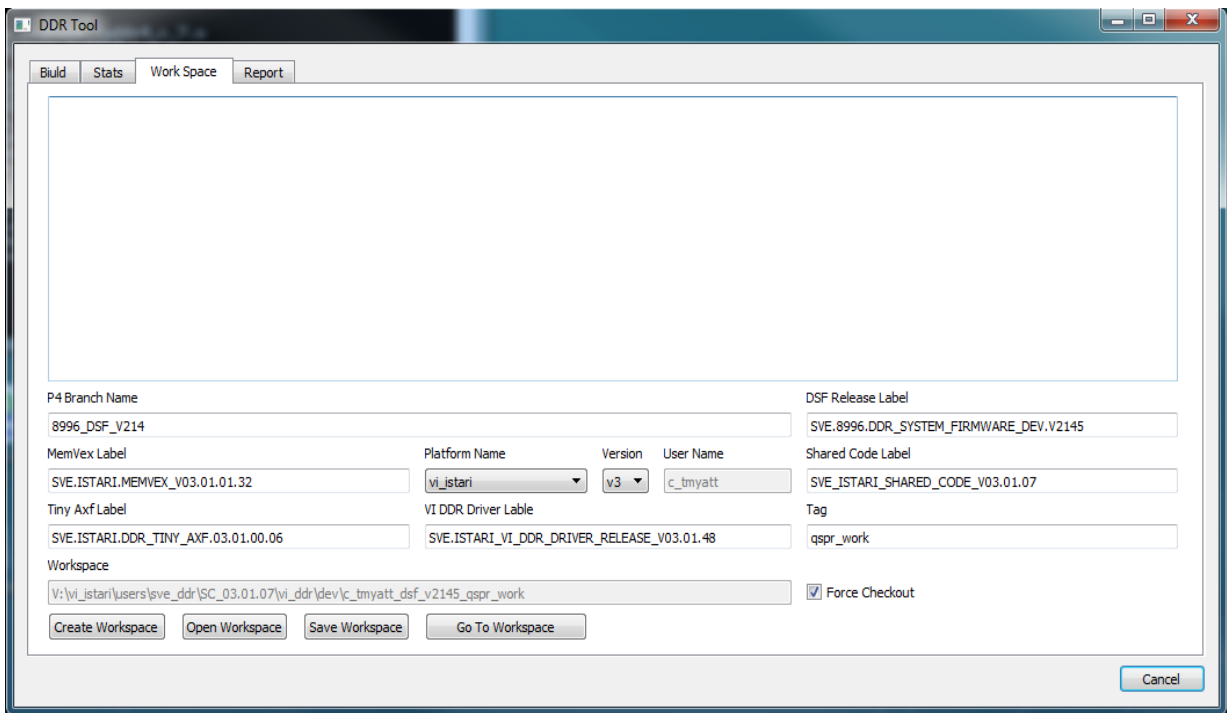


Figure 4 DART Tool

- This tool should eliminate Having to manually read through test log data and prepare a test results report.
- The DDR Tool has been divided into three sub-systems; server application, client application, and interface application. The server application will run on Linux and Harvester build systems. The client application will be controlled by the interface application.

Chapter 5

Work Done

5.1 Project-1

Pre-silicon Validation

- Ported the DDR Test framework to every release of another framework and tested it
 - Make some changes required to bring up the standalone framework
- Solved the errors occurred during the porting and executing phase
 - Error observed while compilation and execution of different tools
- Enablement and testing of different tools
 - Tools such as FreqSw, power, shmoo.

Post-silicon Validation

- Ported the DDR Test framework to every release of another framework and tested it
 - Make some changes required to bring up the standalone framework
- Solved the errors occurred during the porting and executing phase
 - Error observed while compilation and execution of different tools
- Enablement and testing of different tools
 - Tools such as FreqSw, power, shmoo
- System Test Validation for post-silicon
- Support the DDR team to reproduce the error by changing the code as per their need
- Support the Bench team for any issue related to the framework
- Made a document on changes required to port DDR Test Framework and provided it to DDR Team

Pre-silicon Validation (Phase-II)

- Ported the DDR Test framework to another framework and tested it
 - Make some changes required to bring up the standalone framework
- Solved the errors occurred during the porting and executing phase
 - Error observed while compilation and execution of different tools
- Enablement and testing of different tools
 - Tools such as FreqSw, power, shmoo.

5.2 Project-2

Pre-silicon Validation

- Ported the DDR Test framework to another framework and tested it
 - Make some changes required to bring up the standalone framework
- Solved the errors occurred during the porting and executing phase
 - Error observed while compilation and execution of different tools
- Enablement and testing of different tools
 - Tools such as FreqSw, power, shmoo, reboot.
- Ported and Validated DDR FreqSw System Test

Chapter 6

Future Work

- Support Activity on Project-I
- Start working on DART Tool
- Start working on Post-silicon validation for project-II

References

- [1] "Qualcomm," Wikipedia, [Online]. Available: <https://en.wikipedia.org/wiki/Qualcomm>.
- [2] Qualcomm, [Online]. Available: <https://www.qualcomm.com/>.
- [3] "Lauterbach Development Tools," Lauterbach, [Online]. Available: <https://www.lauterbach.com/frames.html?home.html>.