Journal of Engineering and Technology

ISSN 2180-3811 eISSN 2289-814X https://jet.utem.edu.my/jet/index

DESIGN AND IMPLEMENTATION OF GRAPHICAL USER INTERFACE FOR SMART LOAD BOARD WITH RELAY TRACKER COUNTER

N. K. A.Majid*[1], A. A. A. Halim[1], R. Jaafar[1], R. Hashim[1], R. M. Zain[1] and T. T. Chuan[2]
[1] Faculty of Electrical and Electronic Engineering Technology,
Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.
[2] Testhub Sdn Bhd, 75350 Melaka, Malaysia.
*corresponding_nurulkausar@utem.edu.my

Article history:
Received Date:
1 September 2022
Revised Date:
23 November 2022
Accepted Date:
20 December 2022

Keywords:
Graphical User Interface,
Automatic Test

Abstract— A circuit board created to act as an interface between the testing device and automatic test equipment (ATE) is known as a load board or device under test (DUT) board. The load board is made up of many electronic components such as relays that are used to test the DUT by setting it up. Mechanical relays (SPST, SPDT, DPDT) all have operational life time. After the relays have exceeded operational life time (defined as number of contact operations) as specified by the manufacturer, the mechanical relays will cease to operate properly or even stop working completely. This will affect the testing of the

Equipment, Device Under Test, Relay, Visual Basic

semiconductor IC and may inadvertently cause a quality issue. As a result, to automate the test routine, we designed a graphical user interface (GUI) programmed development to track the operation life of each relay of particular load board as preventive maintenance. Programming in Visual Basic was used to construct the graphical user interface.

I. Introduction

In the integrated circuit (IC) manufacturing process, testing is crucial. Before being released into the market, all integrated circuits must be thoroughly verified at the package level. A packed device is subjected to parametric, functional, and structural tests, among other things [1][2]. The aforementioned tests for complex ICs must be carried out at the package level without the use of conventional testing tools such a multimeter, voltmeter, and ammeter. Such testing can be done with automated test equipment (ATE). Furthermore, a complicated ATE design includes the ability to execute various logic tests as well as boundary-scan testing. The device under test (DUT) is the packed device that is tested on ATE [3].

Nonetheless, the ATE engineer faces a significant problem, as the load board's design complexity rises in tandem with the DUT's complexity. A circuit board known as the load board serves as a connection between the ATE and the DUT. A complicated load board is made up of massive electronic components that are used to set up the DUT for testing. To guarantee that the ATE performs proper testing on the DUT, The load board's electronic components must all function properly. To detect the number of operations of all relays on a particular load board with reference to individual relay Id's, a test engineer must manually build the test programmed using line-by-line coding before inputting the test programmed into the ATE. This procedure is inefficient and time-consuming.

To address the problem, we developed a graphical user interface (GUI) that can automatically generate test programmed to monitor the operational life of each relay of a specific load board and save this information in a searchable database with access for continuous monitoring. The system will then transmit an alarm to the technician when a certain relay has approached the end of its operational life, allowing for preventive maintenance to be performed. The remainder of the paper is laid out as follows. The background study behind this project is described in Section II. The method for creating GUI-based test programmed creation software is described in Section III. Section IV presents the findings and analyses. Finally, in Section V, the conclusion is drawn, and future work is briefly outlined in Section VI.

II. Background Study

A. Relay Architecture

Relay control systems are widely used in a variety of technology especially in manufacturing sectors as a switching control circuit. The main reason that contributes to that is because the control system is simpler and have very basic operational principle compared to other types of control system. Therefore, the relay functionality is very crucial to consider making sure the control system is well functioning.

- **Relay Failures**

There are many factors that cause the relay operation failure depending on the relay types. The most command relay failure is due to end-of-life failures (lifespan). When the switching operation exceeded the lifespan counter tested by the relay manufacturers, normally the relay will fail to operate. However, switching voltages and currents over a relay's rated limits might also result in its failure. Although they are not very common, there are some failures that are linked to how long a relay has been in use, particularly when it switches signals at extremely low levels or when it is not used frequently, and oxidation builds up on the contacts.

- **Relay Lifetimes**

In designing the switching control system, the main criteria that need to be considered is the system lifespan. This factor is depending on the relay mechanical life and full load life. Mechanical life of the relay is the operation of relay under low-level switching conditions without mechanical failure due to failure of the actuation mechanism, contact wear, or coil damage. Hence full load life is the relay life under hot switching load.

- **Relay Counting Operation**

Relay operation counters are included in some switching systems to anticipate the relay failure. Despite the awareness of the relay's usage is helpful, it is not a reliable indicator by itself. A relay's operating life can be impacted by load conditions alone by a factor of three orders of magnitude. The reliability of a switching system can quickly decline when relays are replaced after a predefined number of operations and used as a predictive maintenance tool. When replacing a relay, other components on the board (not only relays) may also experience issues, especially if the board has surface-mount components. Pickering only employs surface-mount devices where the switching characteristics call for it (for example for RF applications). The warning of danger is also always present.

B. ATE Block Diagram

DUT's pass/fail status is determined by a test programme. In a production setting, the programme must also convey information to an operator, an automatic handler, or an external instrument, as well as recognise the start and end of test signals. There are multiple ports on an ATE that manage this information exchange [11]. The simplified ATE block diagram is shown in Figure 1. The entire testing regimen can be broken down into five sections. Each portion is further subdivided into two subcategories: resource to resistor and resource to resource.

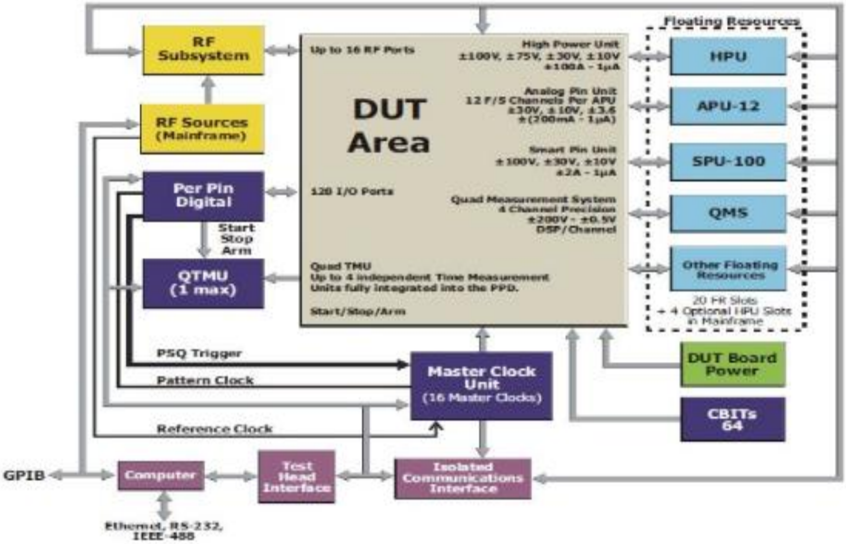


Figure 1: ATE Block Diagram

Several works concentrating on ATE optimization have been proposed in the past [4-8]. Pilot pin integrated into the DUT to shorten the built-in self-test (BIST) waiting time when employing ATE was presented by Garcia et.al [4]. The ATE's wait time before reading the test response is determined by the lowest repetition rate of the DUT domain of core clocks. As a result, the DUT experiences unnecessary waiting time for a faster run-time BIST. (i.e., the highest frequency of core clock domain). As a result, the pilot pin is used to alert the ATE that a particular BIST in the DUT.

DUT has been accomplished and its test outcome is ready to also be read out from the result register. Other research [5-7] have suggested designing and integrating an extremely accurate parametric measurement unit (PMU) in ATE to conduct the test on active devices. In recent work, Ungar et al. [8] expanded high-speed input-output capabilities to ATE by using synthetic instruments targeted at avionics and military DUTs. The use of a field programmable gate array (FPGA) to reduce the cost of ATE has also been investigated [10-14]. A study [12] developed the ELATE, an FPGA-based embedded low-cost ATE that can run functional speed/delay and power consumption tests.

A functional test in FGPA can be executed on fixed random-access memory (SRAM) using an ATE framework, according to a different paper by Xiang et al [11]-[13]. A functional test of an FPGA and a host computer setup is made up of PCB boards. The proposed framework starts by configuring the BIST in an FPGA before testing and gathering the test answer. All the preceding projects have centred on combining ATE with new features and developing a low-cost ATE. The goal of our research is to improve the ATE methodology for testing the relay on load board.

III. Methodology

A. Existing Program Flow Templates

This study is based on Eagle Test System 364 automatic test equipment. To test relay operation limit, a program flows as shown in Figure 2 below is created. This flow template consists of 7 important functions which are necessary for the program to work.

- **UserInit()**

The "TestMain" module only calls this function once at the beginning of programmed loading. It initializes all Global Variables and Global Switches. In this program it initializes each side of load board, normally it been single quad or 4 quads. Grouping all the APU, FSS and CBIT.

- **OnTestInit()**

This process is invoked prior to the Test Main Menu's presentation. Prior to invoking the first function in the programme flow structure, it will be called. Initialize all cbits and reset or reinitialize variables can be placed in this function. In this study, new coding has been added in this function to create a pop-up window to select JIG iD and read relay counter from text file.

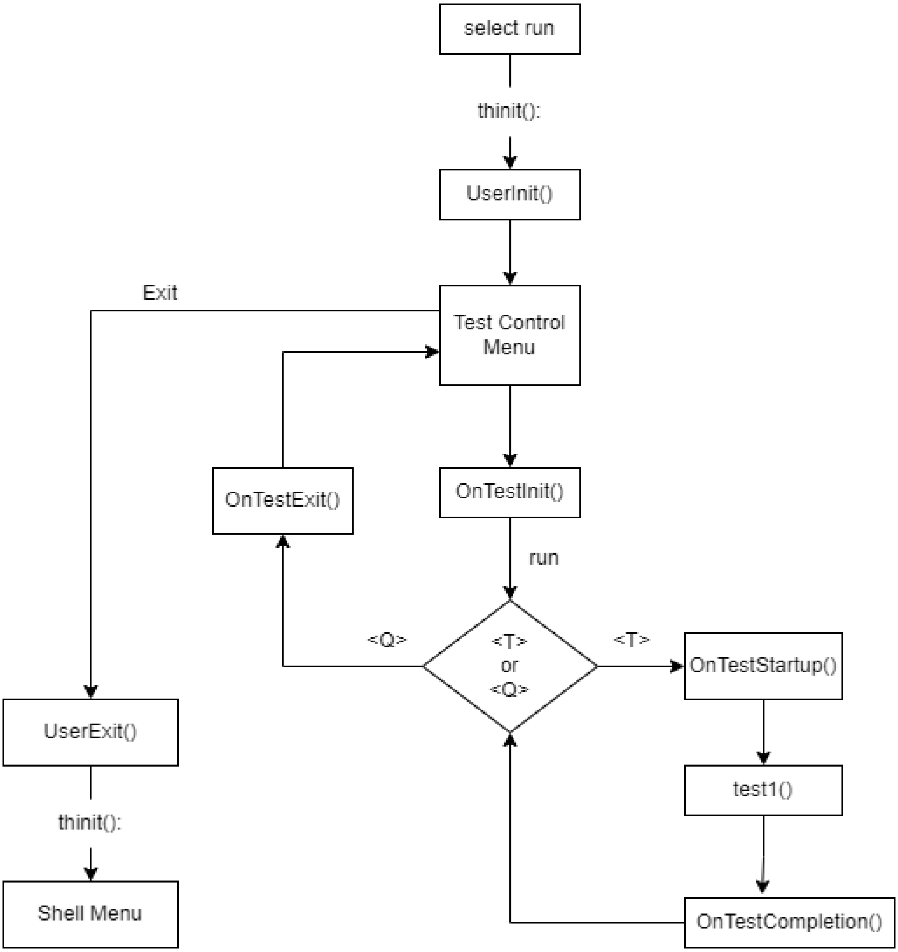


Figure 2: Flowchart in ETS 356 ATE system

- **OnTestStartup()**

Before each test run, this function is called to initialize any system resources or programmed variables that might need to be reset before a device is tested. Same as OnTestInit() function, it will invoked prior to calling the first program flow structure. Reset variables and set active sites to test also been calling in this function. In this function the loadboard side (single quad or 4 quads) availability is determined here.

- **Test()**

Test() function is designed to force the program to measure and store datalog in textfile. In this system, relay operation has been count before the data is stored in textfile.

- **OnTestCompletion()**

This function is called after each test-run and invoke calling the last function in the program flow structure. System resources is turned off here. Cbits is reset and tested device is binned in this function. The counter of relay data is written to textfile in this function.

B. GUI –based Load Board Management System Flow Templates

The Visual Basic computer language is used to create the GUI-based jig management system. In this study, only relay component on the load board is considered to be tested. The test programmed flow for testing the relay component on the load board is shown in Figure 3. In this test program, 2 main users are allowed to access the GUI. First, administrator user. This user is allowed to view and edit the window with valid credential and select the ID number of jig to be tested. Otherwise, another pop-up windows will prompt to acknowledge that is not the authorized user to view and edit the system. The program will display the relay counter and administrator user not only able to view but able to edit and reset the relay counter value. In this study, life-time limit for relays is set to be maximum 1 million operations for each testing. Once the operation of relay reaches 1 million, a new pop-up window will be generated to show that the relay should be change before it burns. Therefore, authorized user must reset the relay counter value after changing the new relay. Second, common user which can view, select jig and read the display relay counter value only.

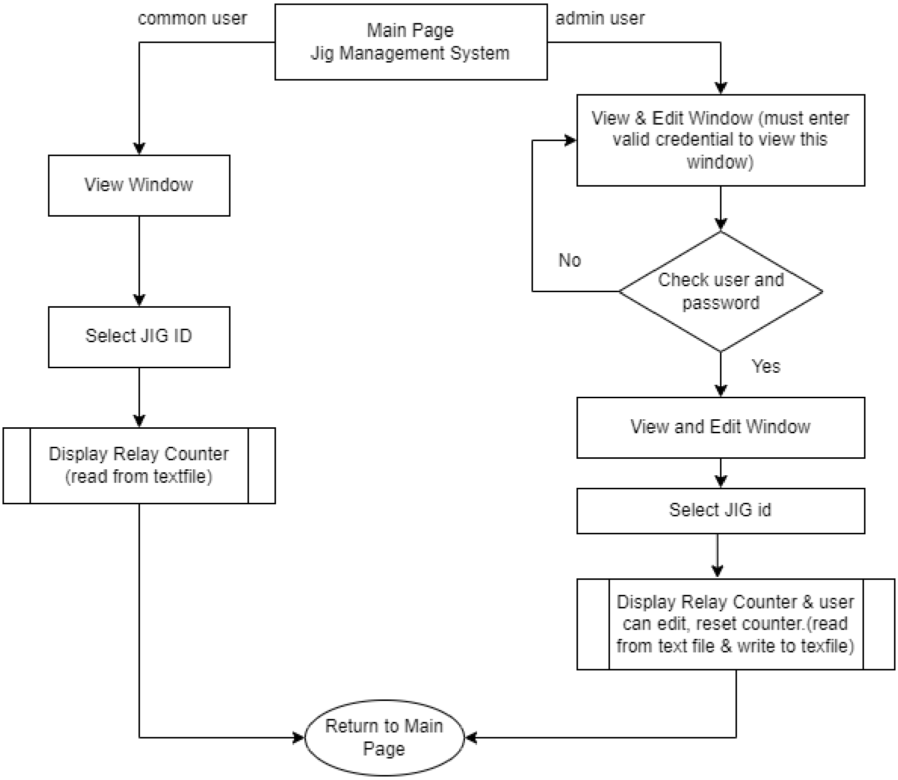


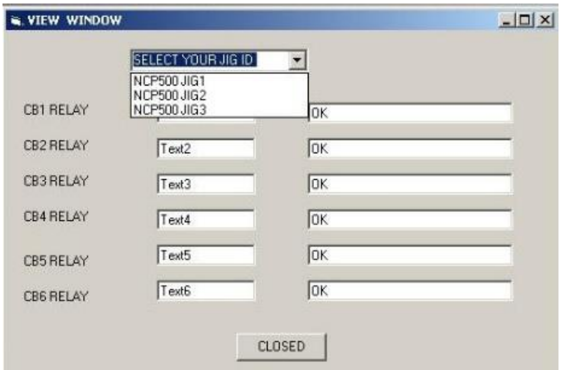
Figure 3: Flowchart of GUI for Load Board Management System

IV. System Output

The developed GUI-based test program generator for checking relay's lifetime on the load board or also called as jig is presented and discussed in this section.

The GUI was created using Visual Basic which represents a graphical user interface that acts as a front end to various programming functions. There is two ways of accessing the GUI, one by invoke from an eagle test system used for monitoring and maintenance team will access from GUI itself for resetting the counter once the relay is change. Most of graphical user interfaces (GUIs) include control elements such as buttons, menus, and labels. In our study two (2) test programs were created which are for single site testing (NCP500 device) and quad site testing (555 Timer device).

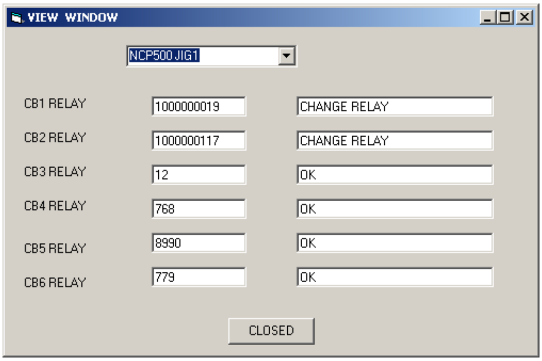
Form an Eagle Test System (ETS), once production finish running a lot, and press quit, instantly launch, and display the relay counter used for monitoring. Figure 4 depicts the GUI interface invoked by Eagle for device NCP500 with single site testing but having three number of load board. Thus, user must choose which load board used accordingly.



CB1 RELAY		OK
CB2 RELAY	Text2	OK
CB3 RELAY	Text3	OK
CB4 RELAY	Text4	OK
CB5 RELAY	Text5	OK
CB6 RELAY	Text6	OK

Figure 4: Relay counter display for NCP500

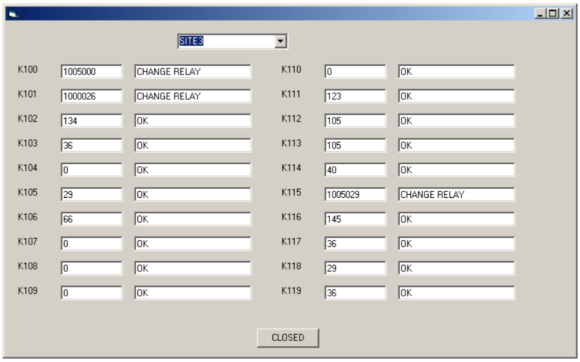
The user must first select the load board/jig number utilized. The GUI will display a list of relays and the number of operations for each relay after the jig number has been selected, as illustrated in Figure 5. Any relays with more than 1,000,000 operations will prompt the user to replace the relay.



NCP500 JIG1

CB1 RELAY	1000000019	CHANGE RELAY
CB2 RELAY	1000000117	CHANGE RELAY
CB3 RELAY	12	OK
CB4 RELAY	768	OK
CB5 RELAY	8990	OK
CB6 RELAY	779	OK

Figure 5: Number of relays' operation display for NCP500

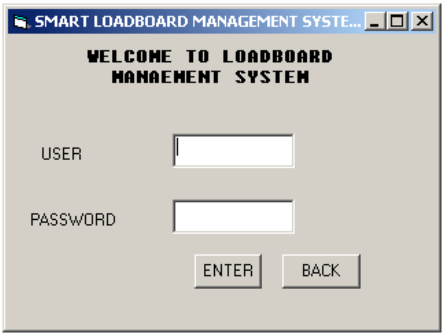


SITE3

K100	1005000	CHANGE RELAY	K110	0	OK
K101	1000026	CHANGE RELAY	K111	123	OK
K102	134	OK	K112	105	OK
K103	36	OK	K113	105	OK
K104	0	OK	K114	40	OK
K105	29	OK	K115	1005029	CHANGE RELAY
K106	66	OK	K116	145	OK
K107	0	OK	K117	36	OK
K108	0	OK	K118	29	OK
K109	0	OK	K119	36	OK

Figure 6: Number of relay's operation for 555 Timer

Figure 6 shows the GUI invoke for monitoring of relay's lifetime for 555 timer which is a device with a single jig with quad side testing. There is a selection of different site to choose from site number one to site number four to view the relay's lifetime.



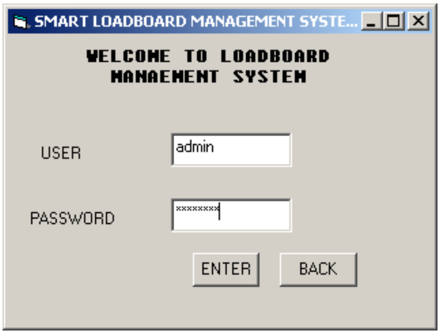
SMART LOADBOARD MANAGEMENT SYSTE...

WELCOME TO LOADBOARD
MANAEMENT SYSTEM

USER

PASSWORD

Figure 7: Main Front of GUI



SMART LOADBOARD MANAGEMENT SYSTE...

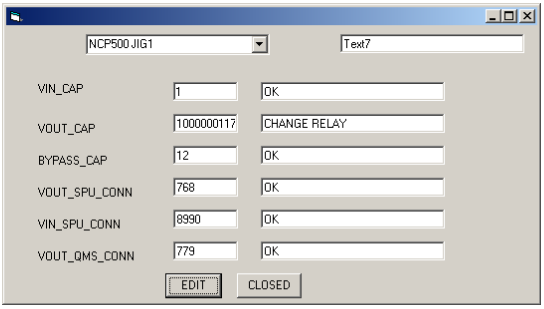
WELCOME TO LOADBOARD
MANAEMENT SYSTEM

USER admin

PASSWORD ********

Figure 8: Credential Check Prior to Use A System

In contrast, a maintenance GUI was also developed so that the maintenance team could reset the counter after replacing the relay. The main page of a maintenance GUI is shown in Figure 7. This system can only be accessible by personnel who have been granted a credential. When the erroneous credential is entered, Figure 8 shows a credential check with a message prompt.



NCP500 JIG1

Text7

VIN_CAP	1	OK
VOUT_CAP	1000000117	CHANGE RELAY
BYPASS_CAP	12	OK
VOUT_SPU_CONN	768	OK
VIN_SPU_CONN	8990	OK
VOUT_QMS_CONN	779	OK

Figure 9: Reset counter interface accessible by maintenance team

In the maintenance system, the extra feature added which is a reset counter. The maintenance will need to select a device name, a site or load board number, and reset the relay counter inside of a GUI system as shown in Figure 9.

V. Conclusion

The design and implementation of GUI to automate the test programmed development to track the operation life of each relay of particular load board as preventive maintenance have been presented in this paper. The test programmed generation serves to track the operation life of each relay of particular load board and store this data in an accessible database with access for continuous tracking. Hence, it will reduce the down time and will increase the productivity and consequently will increase the revenue of the company.

VI. Acknowledgement

The authors would like to acknowledge Testhub Sdn. Bhd., Faculty of Electrical and Electronic Engineering

Technology, Universiti Teknikal Malaysia Melaka (UTeM) and those who give energetic and full support in carrying this research under the grant vote number INDUSTRI/TESTHUB/2021/FTKEE/I00051.

VII. References

[1] I. A. Grout, “Automatic Test Equipment (ATE) and Production Test,” in *Integrated Circuit Test Engineering*, Springer, London, 2005, pp. 257–266.

[2] Sun. C, Zhang. M, “Automated Calibration Methos of Loadboard of Integrated circuit ATE,” in *IEEE International Conference on Electronics and Communication Engineering (ICECE)*, 2018, pp. 130-133.

[3] N. S. Rai, N. Palecha, and M. Nagarai, “A brief overview of test solution development for semiconductor testing,” in *IEEE International Conference on Recent Trends on Electronics, Information, Communication and Technology*, 2019, pp. 205–209.

[4] F. Garcia, J. Padilla, and E. Rosaria, “Waiting time optimization of non-deterministic tests at ATE,” in *IEEE Electronics Packaging Technology Conference*, 2013, pp. 723–725.

[5] E. Collins, I. S. Jung, Y.-B. Kim, and K. Ki Kim, “A design and integration of parametric measurement unit on to a 600MHz DCL,” in *International SoC Design Conference*, 2012, pp. 435–438.

[6] N. K. Ab Majid and F. S. Ismail, “Electronics system thermal management optimization using finite element and Nelder-Mead method,” in *Telecommunication, Computer, Electronics and Control TELKOMNIKA*, vol. 17, no. 5, pp. 2268-2275, Oct 2019.

[7] E. Collins, I.-S. Jung, Y.-B. Kim, and K.-K. Kim, “A design approach of a parametric measurement unit on to a 600MHz DCL,” in *International SoC Design Conference*, 2011, pp. 37–39.

[8] L. Y. Ungar, N. G. Jacobson, and T. M. Mak, “High-speed I/O capabilities added to military automatic test equipment (ATE) using synthetic instruments,” in *IEEE Instrum. Meas. Mag.*, vol. 23, no. 5, pp. 19–26, 2020.

[9] Marcelo De Souza Moraes, M. B. Herve, and M. S. Lubaszewski, “Low pin count DFT techinique for RFID ICs,” in *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2012, pp. 31–36.

[10] P. B. Kelly, “A new class of test instrument: The FPGA based module,” *IEEE Instrum. Meas. Mag.*, vol.16, no. 4, pp. 22–24, 2013.

[11] McFeely. D. E,“ The Process and challenges of high-speed DUT

board project," in *Proceeding. International Test Conference*, pp. 565-573, 2002.

[12] A. A. Bayrakci, "FPGA based low cost automatic test equipment for digital circuits," in *Electrica*, vol. 19, no. 1, pp. 12–21, 2019.

[13] Z. J. Xiang, L. Wang, Y. Jiang, D. Liu, and J. Wang, "A framework for Xilinx SRAM-based FPGA functional test," in *IEEE International Conference on Electronics and Communication Engineering*, 2020, pp. 207–210.

[14] ETS-364 System Manual, Eagle Test Systems Inc, Illinois, USA, 2002.