Journal of Engineering and Technology

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

INTELLIGENT INTERFACE BOARD COMPONENT CODE GENERATOR FOR AUTOMATED TESTING EQUIPMENT

N. A. Mutalib*[1], A. Noordin[1], N. Mohamood[1], M. S. Mispan[1], N. Hashim[1], E. Ruslan[1], D. A. Hadi[1], S. F. A. Gani[1], M. Zahari[1], M. F. M. Basar[1], L. L. Fatt[2], H. Y. Kyan[2] and W. C. Seang[2]

[1] Faculty of Electrical and Electronic Engineering Technology, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.

[2] Engineering Department, Testhub Sdn. Bhd., Zon Perindustrian Bebas, 75350 Batu Berendam, Melaka, Malaysia.

corresponding_nurliyana@utem.edu.my

Article history:
Received Date: 4 October 2022
Revised Date: 1 February 2023
Accepted Date: 9 March 2023

Keywords: Interface Board, Automatic Test Equipment,

Abstract— Automatic test equipment (ATE) has been extensively used in the semiconductor industry to test the integrated circuit (IC). The interface between the ATE machine and the circuit under test (CUT) is known as the interface board or load board. Ensuring the functionality of all these electronic components on the load board is as crucial as reducing the IC testing time. Early detection of malfunctioning electronic components on the load board could save testing time and ensure the correct testing process of CUT. Hence, in this study, we

This is an open-access journal that the content is freely available without charge to the user or corresponding institution licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0).

Digital Pin Unit,
Smart Pin Unit,
Analog Pin Unit

develop an automatic testing automation system using visual basic to automate the generation of a testing program for testing the resistor, capacitor, and relay components on the load board via Digital Pin Unit (DPU), Smart Pin Unit (SPU), and Analog Pin Unit (APU). The generated test program can be used for ATE testing to identify the malfunctioning resistor, capacitor, and relay on the load board. The automation of the testing program managed to increase the efficiency of components testing on the load board where the test time reduce by 75% on average of the total time taken to generate the script file using the manual process. The project has been successfully designed and developed.

I. Introduction

Nowadays, testing integrated circuits (ICs) is challenging as the complexity of the IC increases due to the smaller dimension and finer geometries of the ICs. Moreover, the ICs are equipped with many features to support sophisticated applications or software which require more complex circuitries. It is very crucial to ensure that these IC products operate as expected since the device failure may lead to a catastrophic disaster, especially for critical applications such as military, biomedical, security, etc. Automatic test equipment (ATE) is used to perform Circuit Under Test (CUT) testing to ensure the ICs released to the market are meeting all the specifications, reliable and fault-free [1-2]. Figure 1 illustrates the ETS-364 ATE machine which consists of the test head and mainframe. The mainframe contains the power supplies and control for the test system. The

test head has an interface with the computer using PC bus interface. The load board is mounted onto the test head where it acts as an interface between the test head and the pins of CUT.

Figure 1: ETS-364 ATE machine

The test program generated by the test engineer is passed to the CUT using ATE machine via the interface board or the load board [3]. Figure 2 illustrates the load board where the CUT is mounted on the load board using a test socket. As the complexity of CUT increases, the complexity of interface board design increases which as well manifested in enormous electronic components on the load board for CUT testing.

It is crucial to ensure all the interface board's electronic components are in good operating condition to avoid unreliable testing being performed on CUT by the ATE machine. Traditionally, the test program is manually generated by the test engineer using a line-by-line coding approach. Subsequently, the test program is passed to the ATE and the electrical testing is performed via Analog Pin Unit (APU), Smart Pin Unit (SPU), or Digital Pin Unit (DPU) to identify the broken or malfunctioning components on the load board. The manual process of generating a test program is inefficient and time-consuming.

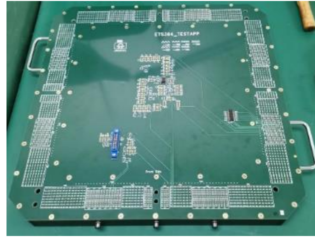

Figure 2: Test interface board

Thus, we introduced a circuit testing automation system that can increase the efficiency and automate the testing of electronic components on the load board such as a resistor, capacitor, and relay. By using the program user can select the desired component according to its labelling, configure the test pins, and the test program will be generated automatically. Subsequently, the test program is ready to be used. With this automation, the process of generating a test script has been simplified. The efficiency increases and the human error that can occur during the manual generation process has been successfully eliminated. This work is arranged as follows. Section 2 explains the literature review. The methodology used to create the automation of the test program is presented in Section 3. Section 4 will elaborate more on the result and findings and finally, conclusions are drawn in Section 5.

II. Previous Related Works

Research in the IC testing field has been focusing on optimizing the ATE and CUT with the goal of reducing testing time and cost. Few works have been proposed on the optimization of ATE and CUT. Xiang et al., [4] invented an ATE that can perform IC testing in extreme conditions such as thermal vacuum, vibration, and shock for the space navigation application. The memory ICs have been used as a case study such as an Electrically Erasable Programmable Read-Only Memory (EEPROM), flash, and Static-Random Access Memory (SRAM). In another study, a system for Field-Programmable

Gate Array (FPGA) aging test is developed by incorporating a temperature sensor and portable ATE [5].

Elsewhere, Garcia et al., [6] proposed an additional pin known as a pilot pin to be embedded into CUT which aims to reduce the waiting time of Built-In Self-Test (BIST). The readout time of the test response is set based on the lowest frequency. Thus, the pilot pin is used to trigger the ATE for readout and avoid unused waiting time experienced by a faster run-time BIST in the CUT in which its expected data. The author in [7] introduced a low-cost ATE system that is able in generating QAM test signals and detect symbol errors in the received QAM signals. In other studies [8-9], a technique to perform the test on active devices is developed by integrating a highly accurate Parametric Measurement Unit (PMU) in ATE. In the latest research, the ATE capabilities of testing DUTs which are targeted for avionics and military applications are improved by adding high speed input-output using synthetic instruments [10].

A study focusing on optimizing the number of pin counts is conducted by Marcelo et al., [11] which targeted RFID applications. A technique of combining the functional and structural tests is adapted. As a result, the pin count is reduced to only three external test pins. In another study [12], the BIST is incorporated in Xilinx Virtex-4 FPGA to detect a single missing clock and test the high-frequency clock generated by the digital clock management (DCM) unit. Elsewhere, Bayrakci, [13] proposed an ELATE, a low-cost ATE based on FPGA that can be used to conduct power consumption and functional delay/speed tests. Another study focusing on developing ATE framework on FPGA is proposed in [14] whereby the developed ATE can be used to run a functional test on SRAM. Several other works have been focusing on developing the ATE on the FPGA mainly to test the memory blocks [15]– [19]. All the above studies have been

focusing on optimizing the ATE and/or CUT by building a low-cost ATE, integrating ATE with additional features, and incorporating CUT with BIST. In our project, we study ATE workflow improvement in testing the electronic components on the interface board to avoid errors when performing testing on CUT.

III. Material and Methods

The algorithm was designed and created using Visual Basic where the program will be used to check the component functionality on the interface board. The broken and damaged components will be replaced once the location is detected using the program. Once the checking process is completed, the interface board is ready to be used at ATE for the device electrical testing process. The designed algorithm can be divided into two configurations. The first configuration is when the component is connected to the ground and the second configuration is when the component is connected to a resource either APU, DPU, or SPU. The configuration is depicted in Figure 3. The simplified block diagram of ETS-364 ATE is illustrated in Figure 4 where it shows the APU, DPU (i.e., other floating resources), and SPU resources used in our study.

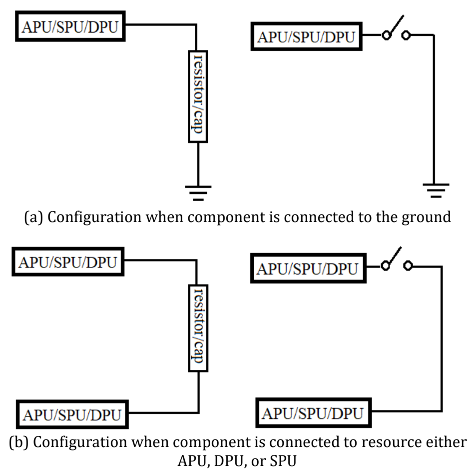

Figure 3: Configuration for the designed algorithm

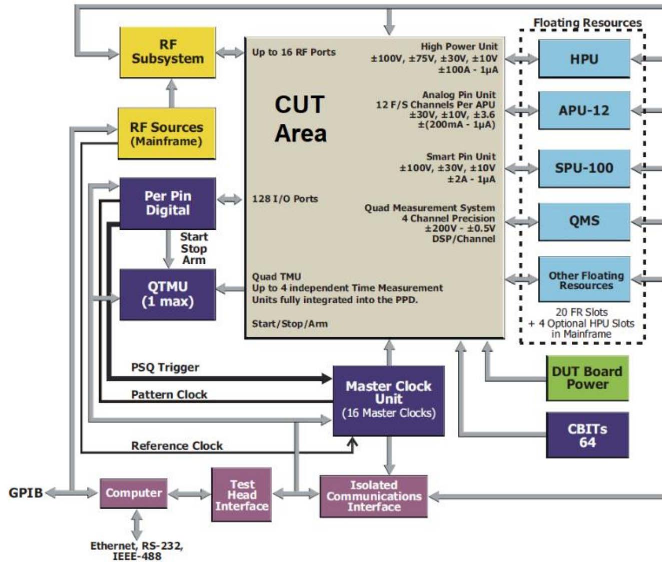

Figure 4: ETS-364 simplified block diagram

A. Grouping Component Value with Suitable Force Voltage and Current Range

The program will force either voltage or current value into the circuit and the component value will be measured and compared with the original value indicated by the label. For example, if we want to test whether the 1 kΩ resistor on the interface board is well functioning, we need to force either current or voltage value to the interface board and measure the resistor value on the other side. If the value falls within tolerances of 1 kΩ, we can consider the component in good condition. Once the output file is generated from the program, the engineer can compare the measured and the original component's values and decided whether the component is in good function. This is the important feature of this program where the engineer does not need to calculate the suitable value for force voltage and force current. This is intelligently done by the algorithm which will automatically calculate and force the suitable voltage or current into the circuit. The forced voltage or current values are determined based on APU,

SPU, or DPU specifications and component power rating. Calculations are done extensively; tables are created, and the suitable ranges are inputted into the program. Previously calculating the force voltage or current value is conducted manually by the engineer which could lead to errors while calculating them. With this program, the engineer does not need to take care of the minor setting and all the manual calculations. Table 1 represents the voltage and current range used in the program for resistor components only. This program will cover resistor values up to 1 MΩ. There are separate tables for capacitors and relays too.

Table 1: Table represents the voltage range, current range, resistor values with 10% tolerances, expected measured voltage, and component power rating for APU

v_range (V)	v_range code	i_range (mA)	i_range code	Tolerance (10%)		Expected Measured Voltage		Power Rating (I^2R)	
				min	max	min	max	min	max
10	APU12_ 10V	100	APU12_ 100MA	0.95	94.50	0.095	9.45	0.01	0.9
10	APU12_ 10V	10	APU12_ 10MA	86.45	945.00	0.8645	9.45	0.0091	0.09
10	APU12_ 10V	1	APU12_ 1MA	855.95	9,450.0 0	0.85595	9.45	0.0009 01	0.009
10	APU12_ 10V	0.1	APU12_ 100UA	8,550.9 5	94,500. 00	0.855095	9.45	9E-05	0.0009
10	APU12_ 10V	0.01	APU12_ 10UA	85,500. 95	945,00 0.00	0.8550095	9.45	9E-06	0.0000 9
30	APU12_ 30V	0.01	APU12_ 10UA	855,00 0.95	1,050,0 00.00	8.5500095	10.5	9E-05	0.0001

B. Program Design Structured

Figure 5 shows the design structure of the program algorithm. First, the user selects the component to test, and this can be either resistor, capacitor, or relay. Next, the user needs to set the component label and value and choose its tolerance accordingly. The tolerance values are automatically calculated by the program algorithm. The resource available to be chosen is either APU, SPU, or DPU. Once the resource is selected, the user must select the configuration as

described in Figure 3 and set the pins. Finally, the test program and script file will be generated based on the setting that has been configured. The results contain the forced voltage or current used to measure the component value via APU, SPU, or DPU and also the limit of the resistance. The results obtained reduces the effort of the test engineer to manually calculate the forced and current values in order to determine the functionality of the tested components. This will save time and avoid errors due to manual calculation.

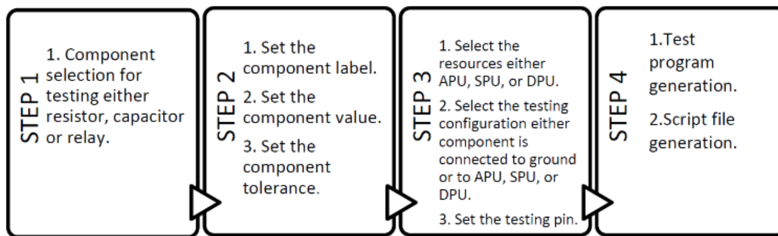

Figure 5: Structure of the design program algorithm

IV. Results and Discussion

Figure 6 to 9 represents the Graphical User Interface (GUI) namely Multilayer Tester with an onion trademark. GUI is created to ease the user when using the program. Figure 6 shows the lists of all the available tests as in step 1 of Figure 5. As mentioned previously, there are three components involved in this program which are resistor, capacitor, and relay. Figure 7 illustrates step 2 which is the selection of the resistor's label, value, and tolerance. There is also an option for the unlisted resistor value by clicking "Others" in the dropdown lists menu where the value can be entered manually. The resistor label is useful for identifying which resistor is to be tested on the interface board. After selecting the tolerance value and clicking the confirm button, the minimum and maximum values of the resistor are automatically calculated based on the selected

tolerance value. Subsequently, the next button is enabled to continue to the further step. The example of the selected configuration is shown in Figure 8. The reset button is available to clear the memory if wrong values were entered previously. After clicking the next button, the menu for testing APU, SPU, or DPU is invoked as in Figure 9. When either one of them is selected, the other menus are disabled automatically. There are two options available for testing configuration as explained in Figure 3. In summary, there are a total of 12 pins available to be selected for the APU, 10 pins available for the SPU, and only one pin for the DPU. Once the pin number has been selected, the user can click the generate button for the output and limit file formulation.

Figure 6: GUI main display

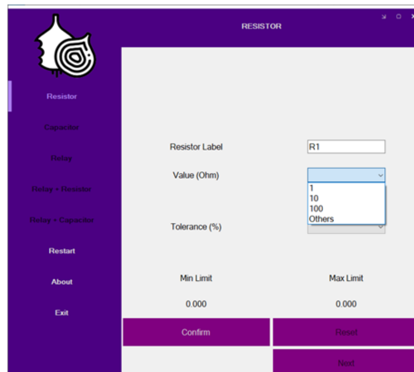

Figure 7: Resistor testing menu

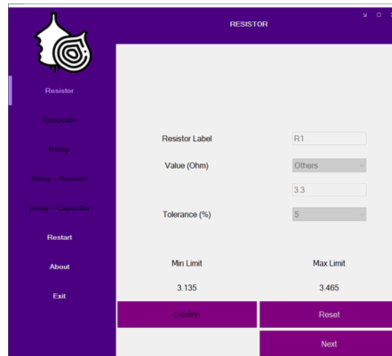

Figure 8: Calculation of the maximum and minimum limit of the resistor

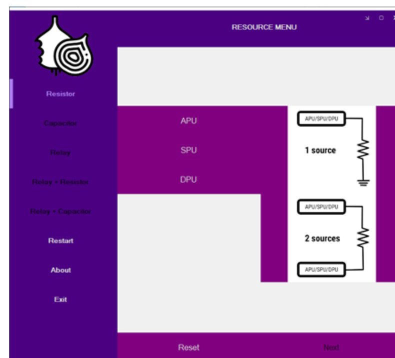

Figure 9: Testing configuration option

Figure 10 shows the output of the test program. It is observed that the force voltage and the force current were automatically calculated by the program and the time taken is faster than manually calculated. This important feature reduces the time consumed for code development, debugging, and testing the CUT. While Figure 11 depicted the limit file which contains all the information about the user activities and operations within the program. Based on estimation, the script file and limit file can be generated in less than a minute after a few clicks on the program. An experienced engineer spends at least 3 to 4 times the amount of time to complete the same output manually, not to mention newbies. The automation program managed to save 75% on average of the code

development time as shown in Figure 12.

```
//DPU TO RESISTOR (ReSource to ReSource)  (10Ohm)
double res;
double volt;
// dpu channel IO38 is high side resource, apu channel IO39 is low side resource
dpinviset("IO38", DPIN_FV, 0.0,APU12_10V,APU12_100MA, DPIN_CLAMP_OFF, DPIN_CLAMP_OFF, MS_ALL);
dpinviset("IO39", DPIN_FV, 1,APU12_10V,, DPIN_CLAMP_OFF, DPIN_CLAMP_OFF, MS_ALL);
lwait(1000);
dpinviset("IO38", DPIN_FI, 0.01,APU12_10V,APU12_100MA, DPIN_CLAMP_OFF, DPIN_CLAMP_OFF, MS_ALL);//100mA
lwait(1000);
volt = dpinmv("IO38", 10, 13.0, MS_ALL);        //expect 1V
res = (volt - 1) / (100/1000);
dpinviset("IO38", DPIN_FV, 0.0,APU12_10V,APU12_100MA, DPIN_CLAMP_OFF, DPIN_CLAMP_OFF, MS_ALL);
dpinviset("IO39", DPIN_FV, 0.0,APU12_10V,, DPIN_CLAMP_OFF, DPIN_CLAMP_OFF, MS_ALL);
dpinviset("IO38", DPIN_OFF, 0.0,APU12_10V,APU12_100MA, DPIN_CLAMP_OFF, DPIN_CLAMP_OFF, MS_ALL);
dpinviset("IO39", DPIN_OFF, 0.0,APU12_10V,, DPIN_CLAMP_OFF, DPIN_CLAMP_OFF, MS_ALL);
msLogResult(0, DSIndex++, res);
```

Figure 10: Output generated file

```
Test Num:000001,Test Description:Resistor ,loLimit:0.000,hilimit:0.000,Unit: Ohm.
Test Num:000001,Test Description:Resistor ,loLimit:0.000,hilimit:0.000,Unit: Ohm.
Test Num:000001,Test Description:Resistor ,loLimit:0.000,hilimit:0.000,Unit: Ohm.
Test Num:000001,Test Description:Resistor ,loLimit:0.000,hilimit:0.000,Unit: Ohm.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Capacitor ,loLimit:0.000000,hilimit:0.000000,Unit: Farad.
Test Num:000001,Test Description:Relay ON ,loLimit: -0.5,hilimit: 0.5,Unit: NA.
Test Num:000002,Test Description:Relay OFF ,loLimit: 7.5,hilimit: 8,Unit: NA.
```

Figure 11: Limit file

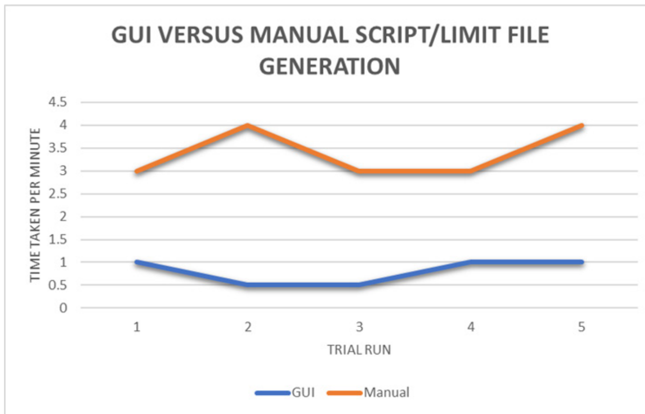

Figure 12: Time consumption of generating test program using manual and automation methods

V. Conclusion

The developed program has been successfully designed and

tested in a semiconductor company. It manages to improve efficiency in the testing process and can reduce at least 75% of the total time taken to generate the script file using the manual process. The error due to the manual calculation of force voltage and force current has been eliminated. The program serves as a straightforward testing methodology for a different kind of passive component testing. Future work will be focusing on improving the interface of the program to make it more convenient and practical for users. Internet of Things (IoT) using latest technologies could be proposed in the future to further ease the monitoring and controlling processes of the overall testing system.

VI. Acknowledgement

We would like to show our gratitude to Testhub Sdn. Bhd. for funding this project. We like to thank the Faculty of Electrical and Electronic Engineering Technology (FTKEE), Universiti Teknikal Malaysia Melaka (UTeM) and those who provided insight and expertise that greatly assisted the research under the grant vote number INDUSTRI/TESTHUB/2021/FTKEE/I00051.

VII. References

[1] 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.

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

[3] M. Radu, “Testing digital circuits using a mixed-signal automatic test equipment,” in IEEE International Conference on Automation, Quality and Testing, Robotics, 2014, pp. 1–4.

[4] Z. J. Xiang, X. Hu, L. Wang, and Z. Li, “An automatic test equipment for integrated circuits in the extreme environments,” in IEEE International Conference on Integrated Circuits and Microsystems, 2016, pp. 46–49.

[5] Z. J. Xiang, W. Liu, L. H. Wang, and L. Lai Wang, “A system for FPGA aging test,” in International Conference on Communications, Circuits and Systems, 2018, pp. 471–474.

[6] 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.

[7] M. Ishida and K. Ichiyama, "An ATE system for testing RF digital communication devices with QAM signal interfaces," IEEE Des. Test, vol. 33, no. 6, pp. 15–22, 2016.

[8] 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.

[9] 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.

[10] 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," IEEE Instrum. Meas. Mag., vol. 23, no. 5, pp. 19–26, 2020.

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

[12] Z. J. Xiang and D. J. Xu, "A built-in-self-test method for DCMs in Xilinx Virtex-4 FPGAs," in IEEE International Conference on Electronics and Communication Engineering, 2019, pp. 51–54.

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

[14] 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.

[15] P. B. Kelly, "A new class of test instrument: The FPGA based module," in IEEE Instrumentation and Measurement Magazine, 2012, pp. 269–271.

[16] S. Fransi, G. L. Farré, L. G. Deiros, and S. B. Manich, "Design and implementation of automatic test equipment IP module," in IEEE European Test Symposium, 2010, pp. 244–249.

[17] M. Ma, X. Chen, and J. Liu, "Characterize the DRAM with FPGA," in IEEE International Conference on Computer Science and Network Technology, 2020, pp. 142–145.

[18] D. De Carvalho, B. Sanches, M. De Carvalho, and W. Van Noije, "A flexible stand-alone FPGA-based ATE for ASIC manufacturing tests," in IEEE Latin-American Test Symposium, 2018, pp. 1–6.

[19] D. C. Keezer et al., "An FPGA-based ATE extension module for low-cost multi-GHz memory test," in IEEE European Test Symposium, 2015, pp. 1–6.