ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P279

Research on Key Technologies for Quality Control in Production of Automotive Powertrain Domain Control Units

Weijie Zhao [1], Biao Wang [2], Chunming Zhao [2], Nenghui Zhou [2]

[1] Nanyang Technological University, Singapore 639798, Singapore

[2] Tianjin Yidingfeng Powertrain Technology Co, Ltd,Tianjin, China

weijiezhao0705@gmail.com

Abstract: Powertrain Domain Control Unit (PDCU) of the new energy vehicle, which is responsible for the vehicle power, energy control, network and safety management functions. In order to effectively ensure the production quality of PDCU based domestical chip solutions, this study focuses on functional testing and End Of Line (EOL) testing techniques for quality control and has developed relevant testing equipment, realize production quality control for PDCU based domestical chip solution.

Keywords：Powertrain Domain Control Unit; PDCU; Functional Testing; End Of Line (EOL) Testing

1. Introduction

Automobiles operate in complex environments, where electronic components must be capable of functioning normally under conditions such as high temperatures, low temperatures, humidity, vibration, and other harsh factors, while maintaining long-term stable operation without damage or degradation.To minimize the risks of failures and accidents and to ensure the safety of drivers and passengers, the safety and reliability of automotive electronics are of paramount importance[1].

However, China's research and development in automotive chip design and manufacturing processes started relatively late, resulting in a significant lack of experience. Issues such as the immaturity of domestic chips and fluctuations in their quality levels[2] have become major obstacles to their application in automotive-grade environments. At the current stage, even though domestic chips undergo relatively comprehensive testing and screening before leaving the factory, inherent quality defects can still lead to quality issues in controller products during vehicle controller production.

To effectively ensure the production quality of automotive Powertrain Domain Control Units, this study conducts research on the key technologies for production quality control of PDCU and develops relevant testing equipment.

2. Research on Key Technologies for Production Quality Control and Development of Testing Equipment

In the production process of automotive-grade electronic control products, functional testing and End-Of-Line (EOL) testing are critical steps to ensure product quality [3,4,5].

2.1 Development of Functional Testing Technology and Equipment

Functional Testing Requirements. In production site functional testing, it is necessary to ensure both sufficient functional coverage and high efficiency in production cycles. The functional testing equipment must support testing for various models of controllers, and account for different configuration versions of the same controller. Therefore, it needs to quickly adapt to testing different controllers by changing the relevant configuration files. Additionally, all test records should be uploaded to a server for convenient traceability and management in the future.

By analyzing failure cases of domestic chips and defining product functionalities, this paper establishes 11 categories with 35 testing items, covering communication tests, analog signal tests, digital signal tests, PWM output port tests, high-side driver tests, low-side driver tests, and more.

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P279

This provides comprehensive functional test coverage for domestically-produced PDCU products, as detailed in Table 1.

Table 1. Test Function List

Number	Item	Content
1	Current Test	13.8V KEY ON
2		13.8V KEY OFF
3		6V KEY ON
4		6V KEY OFF
5	Power Test	Power Accuracy
6	Functional Test	ROM Test
7		EEPROM Test
8		RAM Test
…	…	…
33	Motor Drive Test	On Voltage Test
34		Off Voltage Test
35		Open Load Test

Development of Functional Testing Equipment. To meet the requirements for Functional Circuit Testing (FCT) and program flashing procedures of the vehicle controller, the overall architecture adopts a model consisting of an industrial PC, boards, and signal conditioning boards. It includes button switch inputs and four-color signal light displays. An automated fixture is designed to achieve automatic and fast testing requirements. The overall block diagram of the equipment is shown in Figure 1.

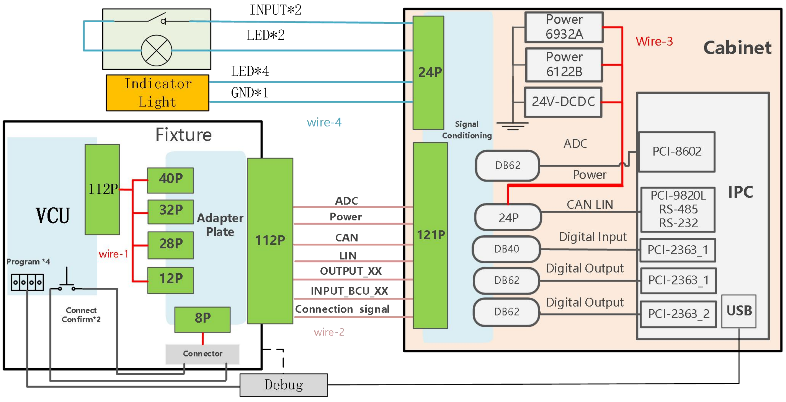

Figure 1. Overall architecture diagram of FCT

The testing software is divided into an upper-layer application software framework and hardware driver abstraction. The hardware driver abstraction mainly includes drivers for boards, signal conditioning, and relay matrix switches. The upper-layer software is primarily responsible for the user HMI interface, MEMS interface, record storage, test sequence code, and configuration files. Figure 2 shows the block diagram of the functional testing software architecture.

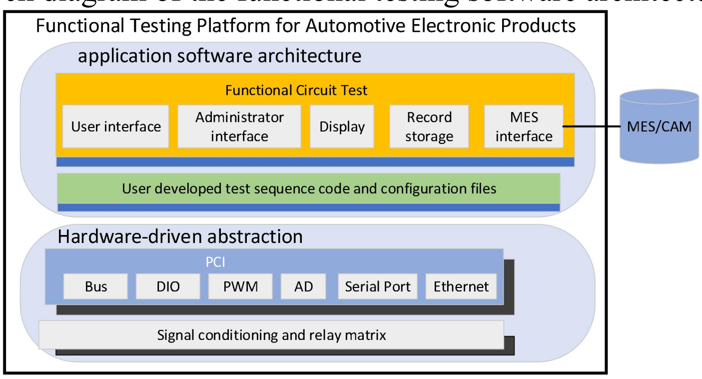

Figure 2. Software Architecture Diagram

The configuration files primarily include settings for the test sequences and hardware delay times.

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P279

Test Sequence Configuration: This is used to configure the testing items for different hardware, enabling flexible testing. All testing items and their expected feedback values can be defined through the configuration files.

Delay Time Configuration: Most testing items involve hardware switching and communication waiting. To minimize testing time, a standardized delay time specification must be defined to reduce unnecessary waiting times. The delay systems involved include switch card switching delays, digital multimeter measurement delays, power-on delays, load card switching delays, electronic load delays, waveform generator delays, input/output delays, and CAN bus communication protocol delays.

To improve efficiency, a 1-to-3 design approach is used. The functional testing cabinet is shown in Figure 3.

Figure 3. FCT Equipment

2.2 Development of EOL Testing Technology and Equipment

To better control the quality of mass-produced controllers, it is necessary to design a testing phase that simulates the vehicle environment at the final stage of production before fully completing the assembly line [5]. The offline automated testing equipment developed in this study is applied to the final automated testing process on the production site.

Development of EOL Testing Equipment. The overall block diagram of the EOL testing equipment is shown in Figure 4. It includes an industrial PC, boards, signal conditioning boards, fixtures, and operation buttons. Operators control the start and stop of the tests using the operation buttons. The testing program automatically imports the test cases, runs the vehicle model, and interacts with the controller under test to exchange data. At the same time, it reads the status of various hardware lines. Through feedback signals, it can determine whether the software and hardware are functioning correctly. Finally, the test results are indicated through signal lights.

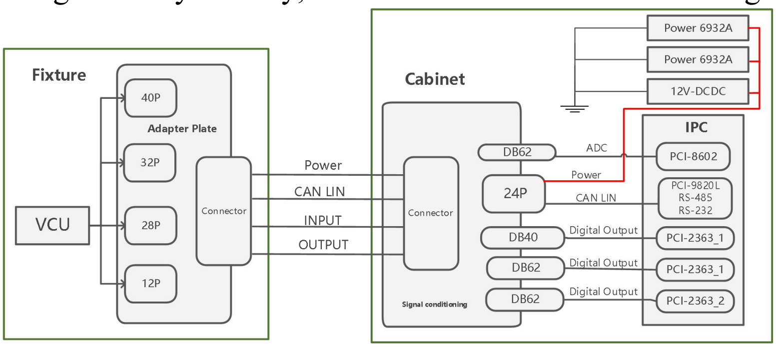

Figure 4. Overall architecture diagram of EOL

The vehicle model used in the EOL testing equipment system is shown in Figure 5. It includes a driver operation model, battery model, motor model, vehicle model, and electrical accessory model. Each sub-model contains a comprehensive logic library, allowing it to be easily adapted to various projects and vehicle types by simply configuring a set of parameters.

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P279

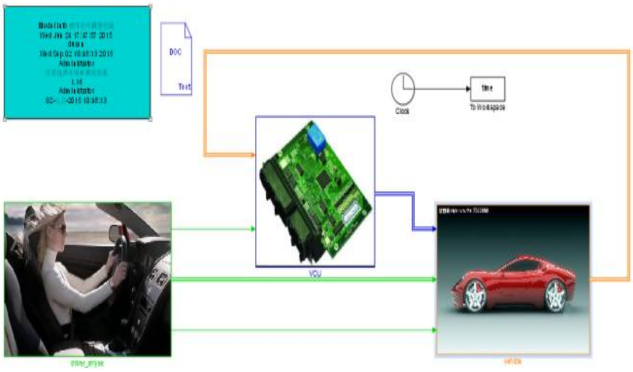

Figure 5. Vehicle Model

The EOL testing equipment integrates automated testing software components, enabling automatic test case import, automatic test case reading, automatic program execution, and automatic test report generation. Based on the operational requirements outlined in the test cases, the system simulates driver operations such as starting, shifting gears, accelerating, and braking. These driver commands are converted into physical or communication signals and sent to the actuators. Once the device under test receives the signals from the actuators, it triggers the corresponding application control programs, which then send feedback to the test processor. The test processor runs models related to the motor, battery, and vehicle powertrain to simulate vehicle operations, verifying whether the application programs of the device under test function correctly. Configuration files in a specific format are used to map the hardware and software of the device under test and the testing equipment. These configuration files are then imported into the testing components for parsing. The detailed process is shown in Figure 6. The software interface and physical equipment of the developed EOL testing system are shown in Figures 7 and 8, respectively.

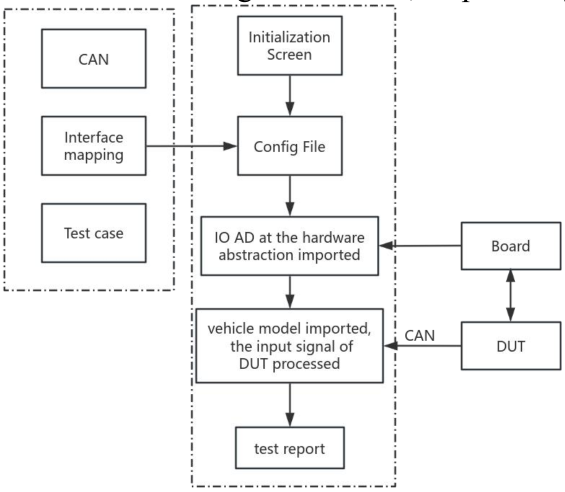

Figure 6. Import Process of Automatic Test Case

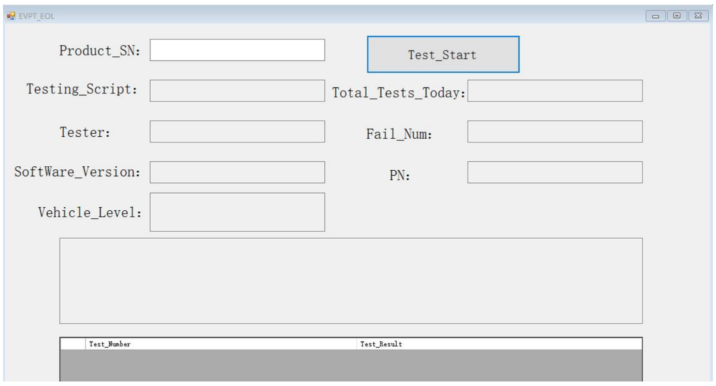

Figure 7. EOL Software Interface

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P279

Figure 8. EOL Testing Equipment

3. Generation of Test Cases

During the application of domestic MCUs, occasional changes to certain fields in the program may occur. In the production process of electronic control units, precise test cases must be designed to meet testing coverage requirements, which often requires significant manpower and time costs. To address this issue, this study developed a method for automatically generating test cases based on symbolic execution, aiming to improve the efficiency of test case design.

Figure 9 shows the flowchart of the symbolic execution analysis process. Symbolic execution involves replacing input variables with symbolic expressions and using the computation of these expressions to simulate the execution of the software under test. During execution, the constraints on symbolic variables at various branches (e.g., `if...else...`) are collected. Essentially, the execution process is a method of analyzing code by extracting the formal semantics of the program, specifically solving the weakest precondition of Hoare Logic.

Subsequently, an SMT (Satisfiability Modulo Theories) solver is used to automatically generate test input data corresponding to the execution paths of the program under the constraints. This approach significantly reduces the time and effort required to design test cases while ensuring comprehensive coverage and accuracy.

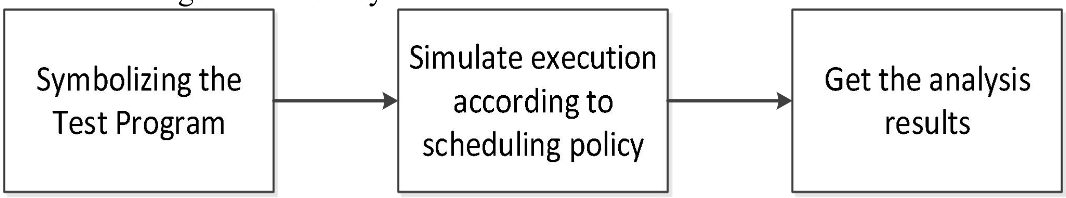

Figure 9. Analysis Flowchart of Symbolic Execution

The specific implementation process is as follows:

First, a program control flow graph (CFG) is constructed from the source code. Then, symbolic execution begins from the starting point of the CFG. During execution, when encountering branch nodes, the constraint solver is used to determine whether the two branches are reachable and whether to continue executing along the path. At the end of the symbolic execution, the analysis of all reachable paths in the program is completed, and the analysis results for each reachable path are obtained.

Next, "systematic search" is used, where the search space is represented by a tree. Each node in the tree corresponds to a Boolean variable, and each edge represents two possible choices for the Boolean variable: true or false. For a tree with n Boolean variable formulas, there will be 2^n leaf

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P279

nodes. Each path from the root to a leaf node assigns a set of truth values. The SMT solver is then invoked at the backend to compute the constraints for each set of conditions.

The generated test cases are automatically saved and recorded. Figure 10 shows the test cases generated in actual applications. This process improves the efficiency of test case design and ensures comprehensive testing coverage.

result	input	input	input	input	input	input	input	input	input	input	output_min	output	output1	output2	output3	output4	output5	output6	output7	output8	output9	output_max
	BMS_fault_II	BMS_uVol_C	BMS_iCurrent_C	BMS_iMaxCurrent_C	BMS_SOC_mp	BMS_tTempreature_mp	BMS_iMAXchr_mp	MCU_fault_II	MCU_tTempreature_mp	MCU_nRspd_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp	ApBms_pwrMAXdis_mp
PASS	0	367	210	420	62	25	220	0	120	4500	100	100	100	100	100	100	100	100	100	100	100	100
PASS	0	365	270	420	55	32	220	0	150	6000	100	100	100	100	100	100	100	100	100	100	100	100
PASS	0	362	330	420	40	46	220	0	160	7500	70	70	70	70	70	70	70	70	70	70	70	70
PASS	0	360	350	420	28	52	220	0	165	8000	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6
PASS	0	356	365	420	19	58	220	0	155	9000	4.6	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7
PASS	0	352	380	420	0	60	220	0	120	10000	0	0	0	0	0	0	0	0	0	0	0	0
PASS	1	380	100	380	100	-10	180	0	-10	0	10	100	99.8	99.6	99.4	99.2	99	98.8	98.6	98.4	98.2	100
PASS	1	375	150	380	90	5	180	0	10	1000	10	98.000026	97.8	97.6	97.4	97.2	97	96.8	96.6	96.4	96.2	100
PASS	1	368	180	380	75	18	180	0	30	3000	10	96.000051	95.8001	95.6001	95.4001	95.2001	95.0001	94.8001	94.6001	94.4001	94.2001	100
PASS	1	367	210	380	62	25	180	0	50	4500	10	94.000077	93.8001	93.6001	93.4001	93.2001	93.0001	92.8001	92.6001	92.4001	92.2001	100
PASS	1	365	270	380	55	32	180	0	80	6000	10	92.000103	91.8001	91.6001	91.4001	91.2001	91.0001	90.8001	90.6001	90.4001	90.2001	100
PASS	1	362	330	380	40	46	180	0	100	7500	10	63.00009	62.8601	62.7201	62.5801	62.4401	62.3001	62.1601	62.0201	61.8801	61.7401	100
PASS	1	360	350	380	28	52	180	0	115	8000	10	26.048046	25.9888	25.9296	25.8704	25.8112	25.752	25.6929	25.6337	25.5745	25.5153	100
PASS	1	356	365	380	19	58	180	0	130	9000	0.47	4.0420085	4.03261	4.02321	4.01381	4.00441	3.99501	3.98561	3.97621	3.96681	3.95741	4.7
PASS	1	352	380	380	0	60	180	0	150	10000	0	0	0	0	0	0	0	0	0	0	0	0
PASS	0	380	100	380	100	-10	180	0	-10	0	10	82.000232	82.4002	82.8002	83.2002	83.6002	84.0002	84.4002	84.8002	85.2002	85.6002	100
PASS	0	375	150	380	90	5	180	0	10	1000	10	86.00024	86.4002	86.8002	87.2002	87.6002	88.0002	88.4002	88.8002	89.2002	89.6002	100
PASS	0	368	180	380	75	18	180	0	30	3000	10	90.000248	90.4002	90.8002	91.2003	91.6003	92.0003	92.4003	92.8003	93.2003	93.6003	100
PASS	0	367	210	380	62	25	180	0	50	4500	10	94.000256	94.4003	94.8003	95.2003	95.6003	96.0003	96.4003	96.8003	97.2003	97.6003	100
PASS	0	365	270	380	55	32	180	0	80	6000	10	98.000264	98.4003	98.8003	99.2003	99.6003	100	100	100	100	100	100
PASS	0	362	330	380	40	46	180	0	100	7500	70	70	70	70	70	70	70	70	70	70	70	70
PASS	0	360	350	380	28	52	180	0	115	8000	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6
PASS	0	356	365	380	19	58	180	0	130	9000	4.6	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7
PASS	0	352	380	380	0	60	180	0	150	10000	0	0	0	0	0	0	0	0	0	0	0	0
PASS	0	380	100	380	100	-10	180	1	-10	0	100	100	100	100	100	100	100	100	100	100	100	100
PASS	0	375	150	380	90	5	180	1	10	1000	100	100	100	100	100	100	100	100	100	100	100	100

Figure 10. Test Case

To ensure that the generated test cases cover all statements in the program, this study also utilizes the coverage testing tool CTC++ to automatically calculate coverage. Relevant software scripts were developed to import the generated test cases, as shown in Figure 11. The test cases are imported into the CTC++ testing project, and the testing scripts are executed to perform automated testing and coverage analysis. The tool automatically generates a coverage report, as shown in Figure 12.

The coverage report provides coverage statistics across various dimensions based on testing requirements, including statement coverage, condition coverage, branch coverage, and MCDC (Modified Condition/Decision Coverage). This makes it easy to assess whether the automatically generated test cases sufficiently cover all program statements, ensuring that all fields in the program are functioning correctly and have not been altered.

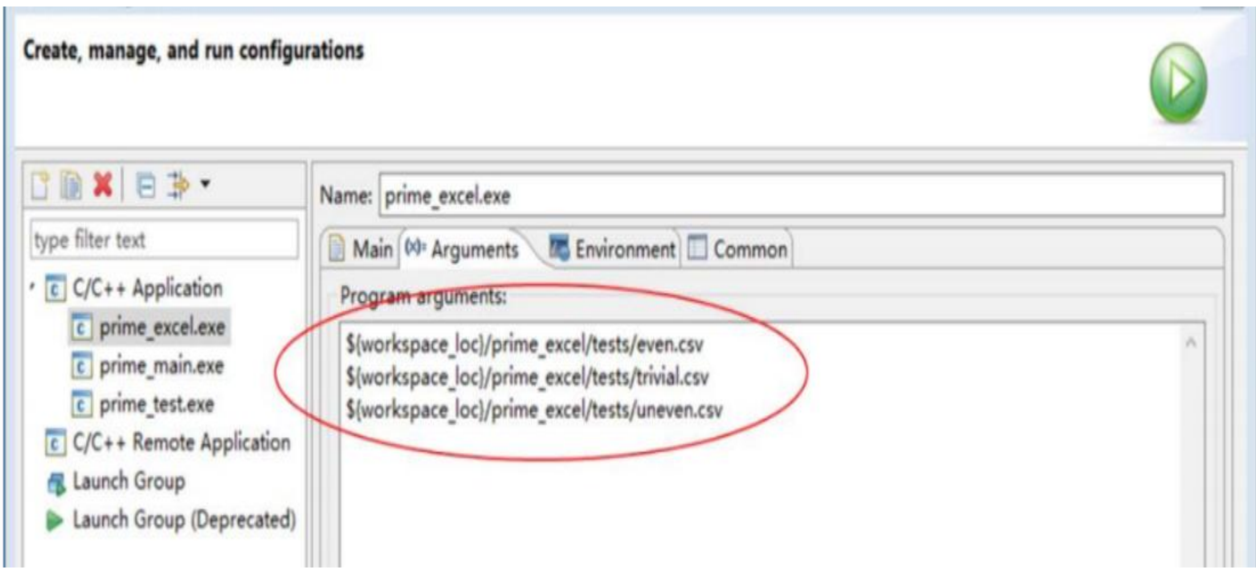

Figure 11. Test Case Import Project

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P279

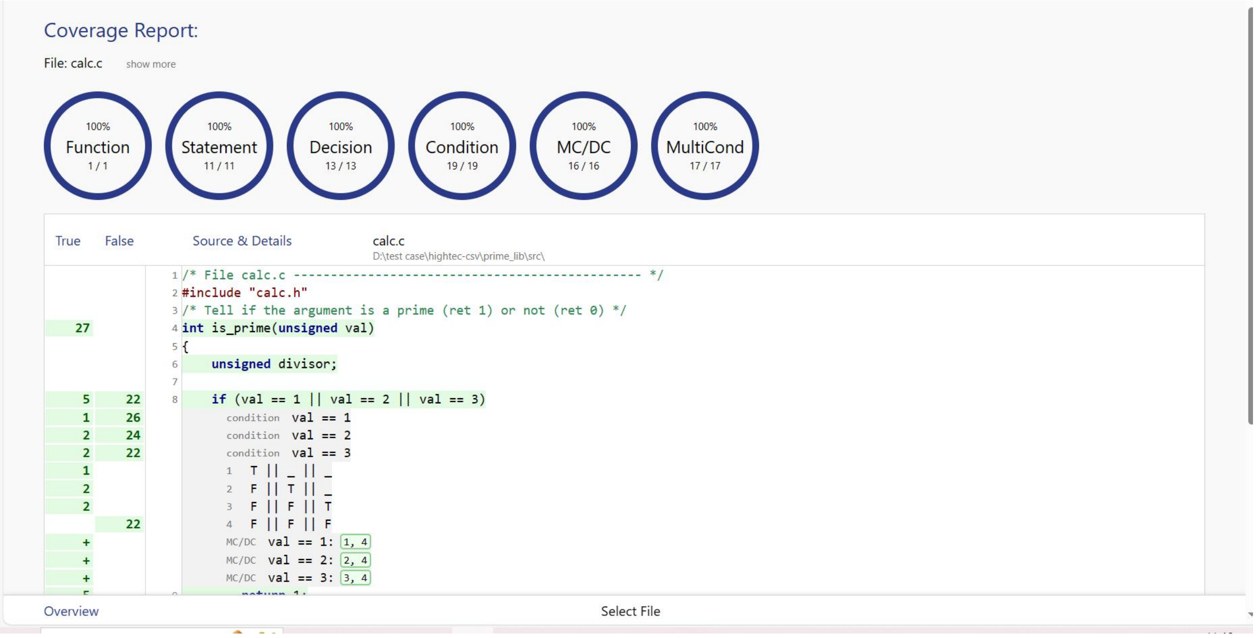

Figure 12. Coverage report

4. Conclusion

This study addresses issues such as performance gaps and quality fluctuations in domestically produced chips by conducting research on key quality control technologies for PDCU production and developing relevant testing equipment. The FCT and EOL testing equipment meet the requirements for functional coverage and software coverage. Additionally, automated testing ensures high efficiency to align with production cycle requirements. These efforts effectively ensure the production quality of PDCU based on domestic chip solutions, promoting the mass production and automotive-grade application of domestically produced chips.

References

[1] Zhao Ziyao. Bridging the "chip" gap for autonomous and controllable development [J]. Automotive Industry Review, 2022(8): 60-63.

[2] Liu Runqing. A brief discussion on the development status of automotive chips in China [J]. Information System Engineering, 2022(10): 62-65.

[3] Wang Xiaodong. Agilent online testing system and functional testing system for automotive electronic product production lines [J]. Foreign Electronic Measurement Technology, 2014(05): 4-9.

[4] Xiao Jie, Chen Zhu, Shen Donghai, et al. Development of a functional testing platform for new energy vehicle electronic control systems [J]. Bus Technology and Research, 2021(01): 60-62.

[5] Li Xiang. Design and development of an automotive EOL diagnostic detection system based on CAN and DoIP [D]. Nanjing: Southeast University, 2021.