Agile development technology of test application program suitable for semiconductor device testing

Tao Yan\textsuperscript{1,*}, MinGuo\textsuperscript{1,2}, XueboZhu\textsuperscript{1}, Yang Yang\textsuperscript{1}

\textsuperscript{1}China Electronics Technology Instruments Co., Ltd, Qingdao 266555, China
\textsuperscript{2}Key Laboratory of Navigation System Simulation and Testing Technology, Qingdao, 266555, China

*Corresponding author’s e-mail: 403294137@qq.com, eqid@ceyear.com

Abstract. The semiconductor devices (especially integrated circuits/chips) test field involves a wide range of device types and test parameters, which leads to low development efficiency, poor stability, weak reuse, long development cycle and repetitive work in application development. Therefore, a test application development method based on reusable component technology is proposed in this paper. Firstly, this paper introduces the basic concepts and characteristics of the reusable component technology. Then, according to the current situation of semiconductor device testing, the design method of component-based test application program is described. Finally, the typical semiconductor device test application program is designed and developed by using LabWindows/CVI integrated development environment. Practice has proved that the component-based test application program design can speed up the R&D and ensure the software quality at the same time. The agile development requirement of application program for semiconductor device test system can be satisfied.

1. Introduction
With the continuous development of electronic information technology and manufacturing technology, the production scale of semiconductor devices has been greatly improved, along with the increasing demand for testing\cite{1-2}. Semiconductor device (especially integrated circuits/chips) testing involves a variety of device types and test parameters, resulting in a huge demand for testing\cite{3-4}. It is urgent to customize and develop a large number of test applications. At the same time, most of the application products in the field of semiconductor device testing are non-standard and non-modular, which makes it difficult to transplant the application program. When users use the test systems of different manufacturers, they will encounter the compatibility problem between hardware and software products, and users need to redevelop the application. Also you have to recompile the application when you change the content of the test project. For different test equipments, even if there is only a small functional difference, it is difficult to achieve reuse. If every test function and test application must be completely redesigned and developed from scratch, problems such as low efficiency, poor stability, weak reusability, long development cycle and repeated work will be caused.

The traditional application program adopts the centralized structure. The components of the application program with this structure belong to the tight coupling relationship, and the structure is compact and complex. The most prominent disadvantage of this software structure is the poor reusability, and it is difficult to inherit the existing application program when developing the new application program. The long-term software engineering practice has shown that reusable component
technology is the key technology to achieve rapid and high-quality software output. The component-based test application is not only flexible and efficient in assembly and integration, but also has the characteristics of high quality, high reliability and good maintainability, which provides an effective solution to the above problems[5].

2. Reusable component technology
Reusable component technology is enriched and evolved on the basis of object-oriented technology. Its basic idea is to divide large and complex software applications into a series of software units that can be implemented in advance, easy to develop, understand and adjust, namely component[6-7]. There are many definitions of components, but there is no generally accepted standard definition. However, practice shows that the so-called "component" is a software unit that can be clearly identified and managed and has certain independent functions; it is also a self-contained software module that can provide a certain service, which encapsulates certain data (attributes) and methods, and provides a specific interface. Developers use this specific interface to use components and make them interact with other components to construct applications. In a word, components are easy to identify and have certain functions.

With the deepening understanding of software reuse, the concept of component has been more widely defined. It is no longer limited to the source code, but extends to the requirements specification of system and software, system and software architecture, documents, test plans, test cases and data, and other useful information for development activities. These information can be called component parts of components and component structure model. The type is shown in Figure 1.

![Component structure model](image)

Figure 1 The component structure model

Reusable component technology has the following characteristics.

1. Highly reusable: a component can be reused in different software. This feature is also an important embodiment of the advantages of components, avoiding the repeated development of the same function.

2. Plug and play: a component is an independent object with specific functions that can cross the boundary of the process and realize the "plug and play" of network, language, application, development tools and operating system. It can be easily integrated into the system without modifying code and recompiling.

3. Interface as the core: the interface and implementation of components are separated. Components interact with other components or systems through interfaces. The specific implementation of components is encapsulated in the interior. Assemblers only need to care about the interface without knowing the implementation details.

4. Language independence: the definition of a component specification does not depend on a specific language, so the language used to write components can be different from that used to write client software.

To sum up, components include both technical and engineering factors. In terms of technology, components can be inherited and reused at the binary level, providing encapsulation of functions, and accessing their services through interfaces externally, so that reusable components do not necessarily need source code. In terms of engineering, components can be independently produced, acquired and
configured, allowing seamless integration into the third software system. In this view, the most ideal way to use components is to build the existing components into an application program[8].

Based on the combination reconfiguration and reuse advantages of reusable component technology, the developed application has high reusability and portability. It can realize the agile development and high reuse of application by using the standard and consistent assembly method to assemble the components.

3. Test application development
The traditional test software usually adopts the "top-down" design, which has certain advantages for the overall design of the software. However, when the overall design of the software or the user's requirements change, it may lead to the original work being pushed over again, resulting in a huge waste of resources. The design of test software based on component technology is distributed and bottom-up, so it can adapt to the change of user requirements by adding or deleting components.

The development of test application based on reusable component technology is divided into two steps. The first step is to develop independent reusable components for general test requirements and form a component library; the second step is to select the reusable components from the component library for specific test tasks, but the components in the component library can’t cover all the requirements, and then develop special components are assembled and debugged to complete the rapid development of test application.

3.1. Reusable component development process
The development steps of reusable components are shown in Figure 2.

(1) Test requirements analysis: determine the functional performance indicators of components. Generally, the functional requirements, performance requirements, control processes, interface requirements and other constraints of components are analyzed one by one to form the requirements specification document of components.

(2) Determine the design scheme: determine the design scheme of components, it is necessary to determine the software structure, data structure and processing process in combination with the requirements specification documents to form the design specification document of components.

(3) Code debugging: according to the nature of the target program and the actual environment, using the appropriate integrated development environment, code conversion and debugging verification of the design scheme. Code debugging usually takes up a lot of development time, so the source code and object program of components should be formed.

![Figure 2 The flow chart of reusable component development](image-url)
(4) Component testing: document review, code check, static analysis and dynamic test of code and documents formed by code debugging. When the component passes the test process, the corresponding test report is formed. If the component fails to pass the test, it is necessary to return to the coding debugging step for modification and improvement, and then conduct the test again.

(5) Component warehousing: add the tested components to the component library for unified management, which can be called at the beginning of the test application. It includes the installation program, target program, source program, development documents, development environment tools and other media materials of the component, which should ensure the normal operation and maintenance and upgrade and update activities in the life cycle of the component.

3.2. Test application development process

Test application is an application software used to complete specific test tasks. Generally, we take the smallest processing unit of each test task as a test step, and all processing units required by each test task as a test program. The development steps of test application based on reusable components are shown in Figure 3.

![Figure 3 The flow chart of test application development](image-url)

(1) Test task analysis: the test task is decomposed into several test steps by combing and analyzing the test task.

(2) Component search assembly: for a single test step, search the matching components from reusable component library for assembly development.

(3) Special component development: when the components in the component library cannot cover all the test requirements, it is necessary to carry out customized development of special components for individual requirements (the specific development process is consistent with reusable components), and the component should be assembled with other components.

(4) Test state setting: set the state parameters corresponding to the test steps.

(5) Test step debugging: debug the test steps. If the test task requirements are not met, adjust the state parameters again.
(6) Comprehensive test: test the application program to make it meet the predetermined requirements. It includes the integration test for the assembly process and the acceptance test for the test task. If it fails, the test task needs to be analyzed again.

(7) Release and delivery: transfer the test application delivery (including application installation program, target program, source program, development documents and development environment tools and other media materials) to the corresponding software product library for unified management and delivery to users for use.

Based on the above development process of reusable components and test applications, it is obvious that in the early stage of building test system, due to the imperfection of component library, more special components may be needed to develop test application for specific test tasks. At this time, we can add the developed special components to the component library as reusable components to enrich the reusable component library scale. With the continuous enrichment and iterative updating of the component library, an open, extensible and reconfigurable test ecological environment will be formed. With the help of components in the component library, the complex test requirements of users can be met without or only a small amount of additional components.

4. An example of semiconductor device test application program

According to the requirements of semiconductor device (especially integrated circuits/chips) test for agile development of application program, a certain type of broadband low noise amplifier is selected as an example. The LabWindows/CVI integrated development environment of NI company and virtual instrument technology are used to develop reusable functional components, and a test application editor is developed to realize the assembly of test application program[9-10].

The test flow of semiconductor device includes test start prompt message, broadband low noise amplifier parameter test, test data collection and storage, test report generation, test end prompt message. The components to be developed include general pop-up message component (as shown in Figure 4), test data upload component (as shown in Figure 5), test report generation component, and parameter test component for a certain type of broadband low noise amplifier. The parameters to be tested include voltage, current, gain, noise figure, input standing wave, output standing wave, etc.

![Figure 4 The pop up message component interface](image1)

![Figure 5 The test data upload component interface](image2)
With the help of the test application editor, the reusable function components and parameter test components are assembled to realize the rapid development of the broadband low noise amplifier test application program, as shown in Figure 6. When other semiconductor devices need to be tested, the development of application program can be completed quickly by redeveloping special test components and assembling components.

5. Conclusion

![Figure 6 The test application editor interface](image)

This paper describes the concept, characteristics and component application programming method of reusable component technology, and its application in semiconductor device (especially integrated circuits/chips) testing field. From the application effect, it can be seen that the development efficiency of test application based on reusable component technology is not only greatly improved, but also has the advantages of easy reuse, easy reconstruction and easy maintenance. It provides a reliable engineering implementation way for the rapid customization development of complex test system software, and can greatly reduce the implementation difficulty of the system in customizing a series of application programs. Through the continuous enrichment and iterative updating of the test application library, it will be able to meet the urgent test requirements in the field of semiconductor device testing.

Acknowledgments

This paper is one of the phased achievements of the program "The integrated development of multi parameter tester for microwave semiconductor devices" (2017YFF0106701), which is a National Key R&D Program of China.

References

[1] Li Na, Zheng Li, Niu Li. Capacity planning in semiconductor assembly and test manufacturing system [J]. Systems Engineering, 2006,24 (8): 107-112
[2] Zhang Zhicong, Zheng Li, Zhang Tao. Investigation on semiconductor test scheduling [J]. Semiconductor Technology, 2008,33 (1): 46-50
[3] Provost B, Sanchez-sinencio E. On-chip ramp generators for mixed-signal BIST and ADC self-test [J].IEEE Journal of Solid-State Circuits,2003, 38(2):263-273
[4] Liu Yuanhua. Trend of ultra large-scale integrated circuit testing technology [J]. Electronics & Packaging, 2014,14 (7): 16-18
[5] Schreiner D, Barany G, Schordan M, et al. Comparison of type-based and alias-based component recognition for embedded systems software[J].International Journal on Software Tools for Technology Transfer, 2013,15(1):41-52
[6] Zhao Haiyuan, Wang Lifang, Jiang Zejun. Design and implementation of general development for measurement and control software based on componentization of ideas [J]. Electronic Design Engineering, 2013,21 (8): 80-83
[7] Huang Ersong, Jiang Huowen, Lu Zhenhui. Package technology---an important technology of software reusing [J]. Journal of Jiangxi Science & Technology Teachers’College, 2003 (6): 93-94

[8] Xu Feng, Liu Ying, Huang Hao, et al. Research of component composition technology based on software architecture connector [J]. Computer Applications, 2006, 26 (4): 836-839

[9] Yuan Dawei. Design of virtual instrument system based on LabWindows/CVI [D]. Master's Dissertation of Harbin Engineering University, 2010

[10] Xie Kang, Sun Xiaojin, Shi Weichun. Application of virtual instrument technology in semiconductor testing [J]. China Computer&Communication, 2018 (5): 36-38