Indian Journal of Science and Technology, *Vol 10(30), DOI: 10.17485/ijst/2017/v10i30/115526, August 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Comprehensive Testing Technique for Embedded System PCBA

Mohammed Naim Khan*, Namita Arya and Amit Prakash Singh

University School of ICT, Guru Gobind Singh Indraprastha University, Dwarka - 110078, Delhi, India;
khan.naim81@gmail.com,aryanamita12@gmail.com,amit@ipu.ac.in

Abstract

Objective: An electronic system developed for specific application with the integration of hardware and software is known as Embedded System. Due to complexity of hardware and software in a single system, it requires specific technique for testing before deployment of the device. This paper proposes the comparative study of testing techniques of embedded system. **Methods/Analysis:** In this paper, a test methodology of embedded system using PCBA has been proposed, that covers all the testing aspects of the embedded system from electrical board level hardware to the embedded system software. Three tests namely power rail test, interconnect test and the infrastructure tests are used for hardware functionality for stuck at and bridging faults. The functionality test has been used for programming validation using port interface. **Findings:** This paper proposed automatic testing of embedded board, which reduces manual errors, increases test coverage and reduces the test time in the production line. The proposed method showed that it can be applied to any board in any form with the same test hardware barring the input-output cables. The software in host machine has been used for testing purpose of different boards. This testing technique is useful for embedded systems implemented in control systems.

Keywords: Boundary Scan, Embedded System PCBA, JTAG, System Testing

1. Introduction

Embedded systems have become integral part of our daily lives being used in the form of appliances such as cell phones, toys, handheld PDAs (personal digital assistant), cameras etc. These systems have set themselves into all fields of our lives be it the kitchen (food processors, microwave ovens, etc.), the living rooms (televisions, air conditioners, etc.) or the work places (fax machines, printer, credit card readers, etc.).The unprecedented growth of embedded systems in our lives have led to increased production of these systems and the competition has forces industries to be cost effective, reliable and with shorter time to market.

Industries involved in development of embedded system are aware that high quality and reliability of their product plays important role for their sustenance in market. These constraints lead to rigorous testing which is expensive and time consuming. On the other hand time to market has become an important parameter for embedded system market hence an intuitive test method is necessary to automate the testing process without a compromise on quality.

Boundary scan testing is widely used in circuit testing of electronic systems. IEEE 1149.1[1] is widely adopted for in-circuit test to validate placement of components, opens between driver and receiver, short between differential pair and short across capacitor[2]. Tests for path delay faults are also achieved using random access scan[3]. Boundary scan tests are also used in backplane testing for multi-drop test architecture[4]. The above boundary scan test techniques are efficient for PCBA level testing for the quality of assembly line but do not have the functionality test of the system. For testing embedded system it is necessary to validate not only the hardware quality but also the software quality.

Embedded systems have to use hardware and software together in order to meet system functionality. Hence hard-

**Author for correspondence*

ware software covalidation techniques evolved. Various hardware software covalidation models like Textual Fault Models, Control-Dataflow Fault Models, State Machine Fault Models, Gate-Level Fault Models and Application-specific Fault Models have been discussed by[5]. Hardware software covalidation techniques for embedded systems includes simulating or emulating a system requirements with a given test input sequence. Covalidation process has three major steps, test generation, cosimulation, and test response evaluation[5]. The covalidation testing techniques are efficient than the boundary scan test in terms of system functionality test. But covalidation test are majorly based on state machine level and fall short of testing the overall system functionality.

For testing of embedded system a hardware software interaction based technique is proposed[6]. This technique focusses on test data selection based on fault injection in hardware and converting it into software fault for selection of test data. A target based and host based testing is proposed[7,8]. It involves inserting application in the device under test and connecting it to host with test scripts to test the services.

In this paper, we present a test methodology for embedded systems that helps reduce test time and covers hardware and software aspects of the system. The paper is structured as follows: In section 2 related works are described and issue of this paper is addressed. In section 3 our approach is presented, followed by conclusion in section 4.

2. Proposed Methodology

The proposed test method for embedded system is for complete PCBA assembly on production line. The test steps are divided into following parts:

1. Voltage Rail Test
2. Programming Processor and Memories
3. Interconnect Tests
4. Infrastructure Tests
5. Functionality Test

The architecture is as shown in Figure 1. The device under test (DUT) is connected to the test board that has a JTAG component via the bed of nails and via connectors for the infrastructure test. Both DUT and test board are connected to test machine or host machine via the JTAG controller that has minimum two ports.

The test system can be adopted for variety of boards just requiring to change the test framework at software level in the host machine. The hardware still remains the same with the interface connection cables adapted as per the DUT requirements.

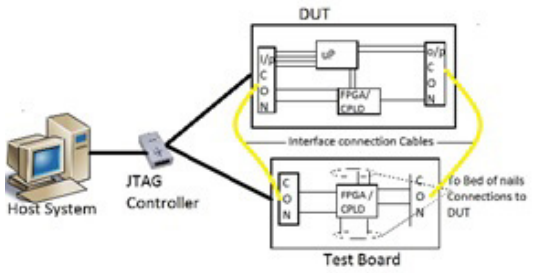

Figure 1. Test Architecture.

The host system runs any of the available boundary scan software (Universal scan, LabVIEW etc...) for JTAG test.

3. Voltage Rail Test

After the assembly of PCBA and random visual inspection and X-ray for soldering quality test the first thing to check is the generation of power supplies on board. This can be checked manually using DMM. As the production quantities of embedded system are large and manual testing is time consuming and increases cost overhead it is effective to automate the voltage Generation test. This is achieved using the bed of nails that are placed on the power rail test pads. These bed of nails are connected to test board via voltage monitors as shown in Figure 2.

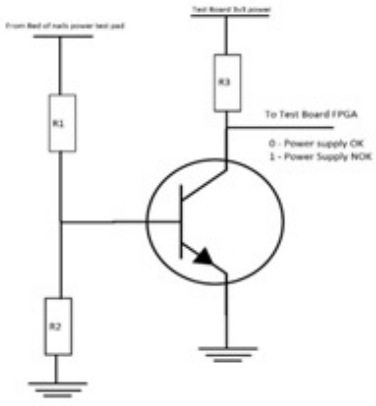

Figure 2. Voltage level monitoring using BST.

The test board has multiple of these voltage level monitoring circuits to test multiple voltage rails. The values of R1 and R2 are adjusted depending on the voltage rail so as to generate 0.6V at 90% of the rail voltage value. If the voltage is within ±10% of the expected value the transistor is ON and a '0' is fed to respective IO pad of test board FPGA indicating the voltage test is passed. Else a '1' is present to indicate failure.

4. Programming Processor and Memories

Embedded system has a microprocessor or microcontroller at the heart of the system. These controlling component requires programming, for microcontroller the chip has inbuilt flash and for microprocessor it requires external memory to be programmed. The external memory can be serial flash like SPI or parallel NAND or NOR flash. The programming of these components is the nest step in testing of embedded system PCBA.

The programming of external Flash and inbuilt Flash is achieved via the boundary scan flash programming if pre-programming of these chips is not feasible. Various tools like universal scan, LabVIEW etc. are available that support the external and Internal Flash programming. Flash programming using universal scan is shown in Figure 3. Alternatively a utility can be developed to emulate the external flash signals and program the device. For Internal flash the device manufacturer specifies the programming procedure using JTAG.

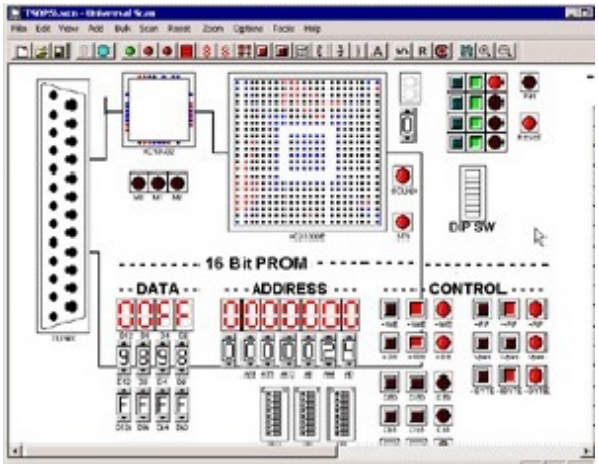

Figure 3. Flash programming using Boundary scan.

The drawback of programming flash with boundary scan is that it is time consuming due to the serial scan chain. So pre-programming the flash is our preferred option. Pre-programming tools from Renesas or Hi-Lo systems can be used. Such tools also offer the gang programming sockets for multiple chip programming thus saving time and effectively cost over the life cycle of product.

5. Interconnect Tests

As the complexity of embedded systems increases the possibility of having more than one boundary scan-able device on the same board increases. As the third step to our testing of embedded system PCBA we test the interconnection between the two boundary scan-able devices on board.

The interconnection test are traditionally performed to check the opens and shorts on the PCBA. These tests provide the stuck at 0 and stuck at 1 hardware faults for the connection between the two devices. The concept is easily extrapolated for more than two devices. Automatic batch scripts can be prepared to run the interconnection test automatically as shown in Figure 4.

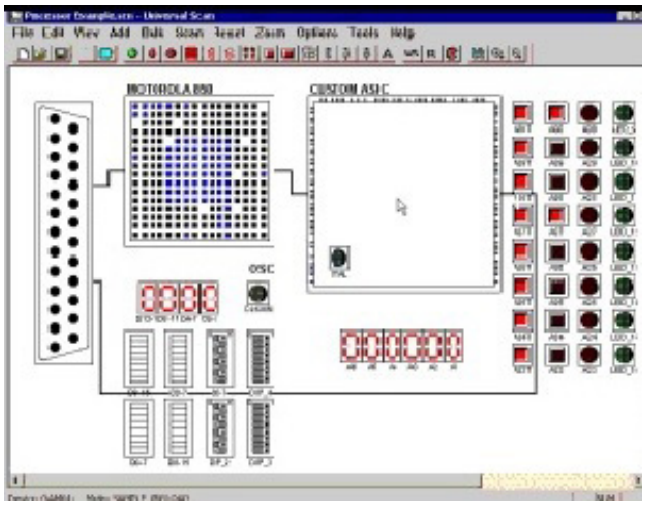

Figure 4. Interconnection test.

Interconnect test can also be performed by applying specific patterns to test the bridge faults between adjacent interconnections.

6. Infrastructure Tests

Infrastructure test are performed by connecting the test board to the input output interfaces of the DUT to the test board. Thus scan chain is established between all the

boundary scan-able devices on DUT and the Boundary scan-able device on test board. The infrastructure test will test all the input output connections that arrive or depart from the boundary scan-able devices on DUT at first level. Other signals that travel through other components and have some Boolean logic before arriving at boundary scan-able device are also incorporated considering the respective combinational logic equations that may have affected the signals. Having the test board and the connector cable error free the infrastructure test will increase the test coverage of the system by testing the external connectors and their connection to on board devices. Here too we test for stuck at '0', stuck at '1' and bridging faults. Both Interconnect test and Infrastructure test are performed in ExTest mode of Boundary scan-able devices.

7. Functionality Test

Embedded systems are covering more and more aspects of our day to day life, this makes the functionality of embedded system complex with tight constraints for time, cost and quality. So the design for test (DFT) constraints needs to be part of the requirements specification itself. As the embedded systems grow in complexity the correctness of the software component has a major role in the overall embedded system quality. We have checked the hardware correctness in previous steps. Now we need to validate the software correctness this is achieved through the Functional testing of embedded system PCBA.

During the development of embedded system itself we have provisions for the debug port that can be used for testing of embedded systems. Traditionally RS232 serial port is used as debug port for embedded systems. With the advent of USB many embedded systems have USB port as debug port due to its compact connector interface. Though USB stack implementation is complex and is preferred to be used only when we have operating system based embedded system. With the widespread use of Internet it is very common to find a RJ45 Ethernet port on embedded system and the Ethernet port can also be used for testing purpose.

Serial port testing is incorporated in the embedded software by developing the software to send messages over UART after BIST during the boot process to the kernel booting. This gives the intuitive details if something goes wrong during the boot process. The embedded software can also include in-line command interface to test various sections of firmware after the boot process is complete. The example commands for UART test for update FMW and verification of FMW are as follows:

```
sudo update -r  /tmp/filename
sudo update -v | grepMrcSpiProj
```

A UART interface boot process is shown in Figure 5.

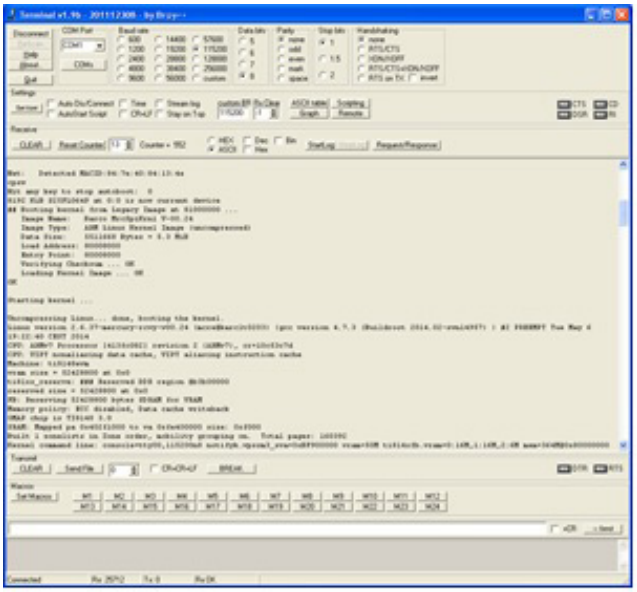

Figure 5. UART Boot messages.

The serial port is used in legacy hardware to test and debug overall system functionality. Thus the DFT requirements of the system need to be incorporated in development phase itself. The in-line commands need to support all the critical functionalities that the embedded system is expected to perform. The serial port testing can be carried out by generic serial terminal like HyperTerminal or Teraterm. Alternatively a GUI based application can be developed for sending the commands that even supports the batch scripts to be run automatically for running the critical test for long term stress testing.

UART interface can also run over RS485 for long distance if the embedded system is expected to run in harsh environment to actually run the test in real time environment or to establish network of embedded systems.

USB serves as alternative to the legacy UART interface port for testing and debugging embedded system. USB has an advantage of being hot plug interface and the host system is able to recognize the embedded system using the vendor ID (VID) and product ID (PID) to invoke proper class driver and its associated application. USB testing

achieves similar results as UART testing. Host machine uses USB interface to program the DUT and to test it with various commands that have been developed to support over USB interface. USB has the advantage of speed and connector form factor over the RS232 connector.

Ethernet connector is used to test the network connectivity of the embedded systems. Similar to UART and USB Ethernet interface is also used to test the critical functionalities of the DUT. The Ethernet commands can be JSON commands or Java script commands. Alternatively the embedded system can publish its webpage over the assigned ip address. The webpage then provides the functionalities that the embedded system is expected to perform. So the functionality test can be performed over the webpage hosted by the embedded system. Ethernet command testing has the advantage of remote testing suitable to test the embedded system in real time harsh environment. The drawback is it requires the complete TCP/IP stack implementation making it feasible only with high memory controllers and it is easily implemented in OS is running in the system.

Alternatively if the debug ports are not available then the DUT figure 1 can be allowed to boot normally with its application program and the Test board in boundary scan mode can be used to send the various verification patterns at the input connector to emulate the real time environment and observe the output from the output connector. The test script development for Verification pattern approach is discussed by[9].

A similar approach of input output domain partitioning and parallel test model development[10] can be applied and test scripts developed in accordance to the FSM states where the test board generates the respective input signals for the DUT testing. Similar boundary scan code coverage of critical embedded system software[11] can be applied.

All the above methods can be applied to the proposed functionality test to increase test coverage and automate the embedded system testing process to reduce test time and effectively cost of testing producing a reliable, high quality embedded system product.

8. Conclusions

Embedded system complexity is increasing continuously in past few years. As these systems become part of our daily life and their application in safety critical applications makes it mandatory for an intuitive test of embedded systems. Also the cost constraint of embedded system due to its commercial market application puts restriction on the test time allocation for the embedded systems.

In this paper we have proposed a test methodology of embedded system PCBA that covers all the testing aspects of the embedded system from electrical board level hardware to the embedded system software. The electrical aspects of the hardware are covered in power rail test, interconnect test and the infrastructure tests. After checking the supplies generated are within the expected limits the interconnection between the devices on board is checked followed by connections to the external world. These tests check the hardware functionality for stuck at and bridging faults.

The programming aspect of embedded system PCBA is also covered along with the functionality test of the embedded software. For the functionality test the UART port, USB port or the Ethernet port can be used. For implementation tests via any of these ports, the design requirement of the embedded software must be adopted to support these test functionality through any of the selected ports. Thus the requirement specification for embedded software must have the testing requirements defined and it should be duly implemented to test the actual functionality of the system else a test image may be required that defeats the purpose of system level testing of embedded system PCBA.

If any of the debug ports are not available, then for functional testing using boundary scan technique the various methods of Input output partitioning and Verification partitioning can be applied using the test board. This also covers the system functionality in real time environment.

The proposed method can be applied to any board in any form with the same test hardware baring the input output cables that need to be adopted as per the board under test. The software in host machine needs to be adopted for the testing purpose of different boards. Thus the test method can be adopted for ideally any embedded system PCBA.

The test methodology proposed is automatic, reduces manual errors, increases test coverage, reduces the test time in the production line proportionally reducing testing cost and increasing the quality of the product delivered.

9. References

1. IEEE 1149.1-2001. Standard Test Access Port and Boundary-Scan Architecture. Available from: http://standards.ieee.org Date accessed: 27/01/2016.

2. John M, Bill E. Embedded Testing in an In-Circuit Test Environment. IEEE International Test Conference Santa Clara CA. 2008. p. 1–6.
3. Le KT, Baik DH, Saluja KK. Test Time Reduction to Test for Path-Delay Faults using Enhanced Random-Access Scan. 20thInternational Conference on VLSI Design (VLSID'07). 2007
4. Tapan C, Chen-Huan C. Bradford G, Van T. A Practical Approach to Comprehensive System Test & Debug Using Boundary Scan Based Test Architecture. IEEE International Test Conference Santa Clara CA. 2007. p. 1–10.
5. Harris IG. Fault Models and Test Generation for Hardware-Software Covalidation. IEE Proceedings - Computers and Digital Techniques. 2003; 20(4):40–7. Crossref
6. Ahyoung S, Byoungju C. An Interaction Testing Technique between Hardware and Software in Embedded Systems. Third International Workshop on Formal Approaches to Testing of Software FATES 2003 Montreal Quebec Canada October 6th. 2003; 192–204.
7. Swapnili K, Anjali M, Goldi J. Development of Software Interface for Testing of Embedded System. 15thInternational Conference on Advanced Computing Technologies. 2013. p. 1–6.
8. Swapnili K, Anjali M. Universal Methodology for Embedded System Testing. 8th International Conference on Computer Science and Education Colombo. 2013. p. 567–72.
9. Wei-Tek T, Lian Y, Feng Z, Ray P. Rapid Embedded System Testing using Verification Patterns. IEEE Software. 2005 Aug; 22 (4):68–75. Crossref
10. Sebastian S. Martin R, Kai-Steffen H. Partitioning the Requirements of Embedded Systems by Input/Output Dependency Analysis for Compositional Creation of Parallel Test Models Annual. IEEE Systems Conference. 2015. p. 96–102.
11. Joao C, Ricardo B, Gilberto R. On the use of Boundary Scan for Code Coverage of Critical Embedded Software. IEEE 23rd International Symposium on Software Reliability Engineering Dallas TX. 2012; 341–50.