On-Line Test Based on SBST of LVDT Demodulation Circuit for Avionics Controller

Zhang Ying, Zhang Xiaokai, Jin Zhengfei and Chen Xin

Abstract—**In recent years, deep sub-micron technology has promoted the booming of the semiconductor industry, but it also makes it difficult to test the chips. The integration of chips is becoming much more complicated, and traditional online test methods are incapable of testing high-reliability systems efficiently. Against this background, we propose a novel test structure based on Software Based Self-Test (SBST) to assist in testing aviation system. This structure employs a low-overhead microprocessor to control the entire test flow, including testing of the processor itself and testing of the target circuits. In order to achieve ideal test effect, we focus on applying optimal test methods to different parts of the circuit such as DFT and ATPG. The on-line test results of a LVDT module for avionics control system proves that the test structure is efficient and flexible.**

Index Terms—**SBST, On-line Test, CPU Self-Test, ATPG**

I. Introduction

WITH the continuous development of semiconductor technology, greater demands are being placed on the overall performance of electronic systems. In order to improve the reliability and stability of electronics systems, especially for secure-sensitive application such as avionics control system, efficient test methods are urgently needed for maintenance. The traditional off-line test methods are liable to detect the permanent faults statically, while hardly can detect transient faults or errors that may occur on the function mode. This scenario will be fatal for fault-sensitive electronic system. Another test method utilizes automated test equipment (ATE) based on computer information processing and built-in self-test (BIST) test structure [1]. However, this method needs to modify the original structure of the circuit and increases the area of the circuit, which may affect the performance of the whole system. Moreover, the cost of ATE is quite high.

Meanwhile, embedded processor is adopted by most designers in the design of integrated circuits. As the technology of chip processing advances, delay faults of wires and the incompleteness of signals are more likely to occur. Some of the faults will only be activated when the circuit is running at full speed, for typical ATE and BIST scheme, more expensive and faster test equipment is required. These technical challenges of test have aroused the research on new test methods to test processors or controllers over the past decade. The method called Software Based Self-Test (SBST) adopts the processor's instruction set to test the processor. In recent years, functional testing based on SBST has been presented as a suitable solution to the problem of test generation for complex VLSI systems, such as microprocessors [2]. Shen and Abraham [3] developed a tool called Vertic which is able to generate test programs. They applied it on GL85 processor and proved the high efficiency of SBST for the first time. Then Parvathala, et al [4] proposed an automatic function self-testing method called Frits which is able to generate random instruction sequences. Gurumurthy, et al [5] developed an instruction generation technique for faults that are hard to be detected. They also utilized ATPG tools to help generate test vectors.

These proposed SBST schemes focus on the self-test of the processor. In addition, on-line testing is also an effective method to capture operational faults, detect the transistor aging, and improve the system reliability [6]. So we apply SBST to the on-line test of Linear Variable Differential Transformer (LVDT) demodulation module circuit besides processor for an avionics controller. We propose a test structure based embedded MIPS processor with high fault courage and low overhead. When the processor functions normally, it will implement on-line testing for circuits under test by executing prepared test programs stored in embedded RAM. Also the processor is able to analyze and store the test responses on-line. Test vectors can be generated by LFSR circuits or ATPG tools. Considering the modular construction of proposed test scheme, it will be adaptable for other fault-sensitive system on-line test.

The rest of the paper is organized as follows. Section II introduces the fault models and test methods of integrated circuit. Section III describes the architecture of on-line test system we designed. We proposed the implement of on-line test for avionics controller in Section IV. Section V provides the experiments results. Finally, the paper is concluded.

II. Integrated Circuit Testing Method

This section introduces the models of faults in integrated circuits and the test methods currently employed. Emphasis is laid on the SBST method adopted in the paper. The general workflow and the advantages of SBST are also introduced.

Manuscript received January 12, 2019; revised January 16, 2019. This work was supported in part by the National Natural Science Foundation of China (No. 61701228 and 61106029), the Aviation Science Foundation of China (No. 20162852028 and 20152052025) and the Fundamental Research Funds for the Central Universities (No. NS2015045 and NS2016046).

Zhang Ying is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, 211106 CHINA (phone: 86-25-84892381; fax: 86-25-84892452; e-mail: tracy403@nuaa.edu.cn).

Zhang Xiaokai is with the Aviation Key Laboratory of Science and Technology on Aero Electromechanical System Integration, Nanjing, Jiangsu Province, 211100 CHINA (e-mail: xiaokai989@hotmail.com).

Jin Zhengfei is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, 211106 CHINA (e-mail:609204625@qq.com).

Chen Xin is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, 211106 CHINA (e-mail:xin_chen@nuaa.edu.cn).

In order to improve the accuracy of fault identification, the logic level fault model of the circuit is mapped to the processor instruction level by simulation. The key points involved in mapping are circuit fault model, processor model and instruction level fault model. Based on the processor model, the simulation results of faults given by software self-test should fully reflect the relation between the underlying fault and the instruction-level fault.

A. Circuit Fault Model

Circuit fault models mainly refer to stuck-at fault, flip-over fault, bridging fault, delay fault and intermittent fault.

1) Stuck-at Fault

The effect of stuck-at fault is that the output value of a signal line in a circuit or system (such as the input line of a gate, the connecting wire, etc.) cannot be changed, and its value is always 0 or 1 [7]. In digital system, according to the fixed value, stuck-at fault can be divided into stuck-at-1 fault and stuck-at-0 fault. Stuck-at-1 fault refers to the situation that level remains at 1 while stuck-at-0 fault refers to remains at 0. Stuck-at fault is not simply a physical defect of short circuit caused by connecting wire to ground or connecting wire to power supply. It is a kind of abstraction of circuit fault at the level of logic function, which means that the logic value of a connection is fixed to 0 or 1 and cannot be controlled.

According to the number of stuck-at fault in the circuit, it can be divided into another two categories: single-stuck-at fault which has only one stuck-at fault in the circuit and multiple-stuck-at fault which contains two or more stuck-at faults in the circuit.

2) Flip-over Fault

Unlike stuck-at fault, flip-over fault usually is caused by environmental disturbances and it is not a permanent fault existing from the beginning of circuit manufacturing. Because dynamic logic units and memories are sensitive to the external environment, they are most prone to flip failures in the system. There are two types of overturn and transition failures. One is permanent flip. The signal on a certain line is always in the flip state and cannot be restored to normal state. One is temporary flip, which causes the signal on a line to flip, and after a period of time, it can automatically restore to the normal state of the circuit. Due to this kind of fault which is not caused by the damage of hardware, it cannot be solved by the ordinary hardware testing methods. However, the impact of flip-over fault on the circuit is enormous and it is difficult to detect, so IC designers have to consider this aspect when they design the chips. Nowadays, flip-over fault detection is also regarded as an important research direction. The on-line testing is one of the promising solutions to this problem.

3) Bridging Fault

Bridging fault means the unexpected connection of logic gates or signal lines in a circuit, which is generally referred to as short circuit phenomenon. Expression < a, b > is usually employed to indicate that there is a bridging fault between node A and node B. Bridging fault can also be classified according to the nodes and node locations. Nodes can be divided into internal nodes (gate, source, drain and base) and signal lines (input lines and output lines). Node locations can be divided according to whether they are in the same transistor or in the same gate. In different circumstances, bridge fault may be handled as stuck-at fault or flip-over fault.

4) Delay Fault

Delay fault mainly considers the dynamic failures of signals in the circuit and it focuses on the time-delay changes of various components and the changes of edge parameters of the pulse signal. This kind of fault has a great impact on sequential circuits, because it will cause disorder in the timing coordination of circuits. Transmission delay fault and path delay fault are common delay faults. Transmission delay fault refers to the fault resulted from signal propagating through components or wires. The transmission delay fault in the circuit includes gate transmission delay fault and line transmission delay fault. In fact, when the system works at low clock frequency, the gate transmission delay in the circuit is much larger than the line transmission delay. A physical path P in a combinational circuit is an alternating sequence consisting of line and gate (or fan-out point). Line l connects gate g to the input of the next gate. These lines and gates constitute a physical path P which can be divided into ascending path and descending path according to the jump of input. Each logical path in the circuit corresponds to a possible single path delay fault.

5) Intermittent Fault

A fault that lasts for a limited period of time and then recovers the required functional capacity by itself without being repaired is called an intermittent fault. Intermittent fault is a common fault in avionics controller. Like flip-over fault, it is usually caused by external environmental factors. It is difficult to detect intermittent faults by common test methods because such faults do not have repeatability. There are some similarities between intermittent fault and flip fault, but also differences between them [8]. Flip-over fault may result in wrong circuit behavior in a short time and then it will disappear and no longer occur. However, intermittent fault can lead to the alternation of wrong and correct circuit behavior, and there is no regular. It is hard for integrated circuits to deal with the damage caused by flip and intermittent faults, unless a fault happens and is detected by the circuit during testing. Therefore, on-line testing will be an effective solution.

B. Test Methods

The digital modules of avionics controller are vulnerable to all the above fault models and faults are more difficult to detect as the scale of the circuits increases. There is a pressing need to implement the effective testing for digital modules in avionics controller. The current mostly applied test methods are scan-based test, build-in self test and software-based self test.

1) Scan-based Test

Scan design is a widely used structured testability design method. It is proposed for complex sequential circuit testing, mainly to increase the controllability and observability of sequential logic units. The idea of scan has long been adopted in system testing, but it was first proposed to be applied to hardware test by Williams and Angell of Stanford University in 1973 [9]. Nowadays, the traditional scan design is one of the most widely used methods in design for Testability (DFT).

With the development of digital integrated circuits, sequential circuits have been adopted by more and more designers. So the test of sequential circuits starts to attract the attention of the industry. However, sequential circuits are more difficult to be tested than combinational circuits in the following aspects.

i. The internal sequential unit of the sequential circuit cannot be directly controlled, so it is difficult to set the internal nodes to certain logical values. Besides, the state of the internal sequential unit of the circuit is difficult to be observed as well.

ii. The test generation of sequential circuits is very complex and the amount of test vectors is large, so it costs quite much time to generate and apply test vectors.

Experts and scholars have made breakthroughs in the research on testing of sequential circuits. The basic idea of test method based on scan design is to turn the original registers into shift registers and connect them together. The input signals are transferred to the internal register by shift register, and the output signals can be shifted out by the same way. By these means, the controllability and observability of the circuit are improved obviously.

There are two kinds of scan test methods which are full scan and partial scan. Full scan, as its name implies, means that the shift register chain contains all the registers of the circuit, while partial scan means that the shift register chain only connects a part of the internal registers leaving other registers unchanged. The two implementations cause the difference in fault coverage and hardware overhead. For the specific application which needs to choose between full or partial scans, fault coverage and hardware overhead should be taken into account together.

However, the hardware overhead of scan design is still high. The extra overhead increased by adding scan path accounts for about 30% of the total production cost, so the application of this method in VLSI chip testing and batch production is restricted.

2) Build-In Self-Test

Build-in Self-Test is a technology that inserts relevant functional modules into the circuit during design phase to make the circuit able to test itself and reduce the dependence of ATE. BIST is a DFT technology which can be applied to almost all circuits, so it is widely adopted in the semiconductor industry. For example, BIST is employed to embedded test pattern generate circuit, sequential circuit, mode selection circuit and debugging circuit in DRAM testing.

Nowadays, highly integrated circuits are widely adopted. At the same time, testing these circuits requires high-speed mixed-signal equipment. The rapid development of BIST technology is due to the circuit complexity and high ATE cost. BIST is capable of solving many problems for circuits which cannot be tested directly due to having no external pin and it also helps reduce the cost of testing.

BIST designs a debugging circuit inside the chip to make testing easier and it is in charge of generating test sequences and analyzing response data. Therefore, three hardware parts need to be added which are test sequence generator, output response analyzer and test control module. There are several methods to generate test sequences such as exhaustive test, pseudo-random test, weighted test, adaptive test and pseudo-exhaustive test.

BIST circuit includes test generation circuit, data compression circuit, analysis circuit, memory circuit for ideal response (ROM) and test control circuit.

In the non-test mode, the circuit under test works normally with input and output connected to external circuits.

In the test mode, the test flow is scheduled by the test control circuit. Under test control, the input channels of the circuit are selected as test inputs. The test sequence is generated by the test generation circuit and applied to the circuit under test. Then, the test response of the circuit’s output channels is compressed by data compression circuit and sent to the analysis circuit. The analysis circuit compares the output response with the expected response and stores the results for further usage.

Compared with scan-based test, BIST has lower hardware overhead. However, BIST is applied in test mode and not capable for on-line testing.

3) Software-based Self-Test

The market demand for high-performance chips with low power consumption is increasing day by day. This trend has prompted processor manufacturers to develop new processor manufacturing methods. However, the improvement of manufacturing technology cause more faults as the decrease of device size, which increases the need for on-line testing. At the same time, the improvement of processor’s frequency also requires more expensive external test equipment to test the chip.

These testing challenges have leaded the semiconductor industry to consider new testing methods over the past decade. The purpose of new methods is to achieve ideal fault coverage per million gates without increasing excessive testing costs or affecting the design of the depth optimized high-performance processors. Therefore, the software-based self-test method appears and becomes one of the most dominate solutions for on-line chip testing.

Traditional integrated circuit testing methods are ATE and BIST. ATE is expensive and it is unable to support full-speed testing. BIST costs low and supports full-speed testing, but it increases the area of on-chip circuits. By adopting SBST, we are able to implement full-speed testing with low cost and less area overhead. In brief, the SBST is an efficient method for processor testing.

The emphasis of SBST is to test the processor and its peripheral modules by employing the resources on chip. The processor takes advantage of its own instruction set to generate and load test vectors and eliminates the requirement for additional hardware overhead (such as scan chains). The test flow of SBST consists of three parts [10].

First of all, the developers will design test programs and generate test vectors for test and diagnosis according to the circuit under test.

Next, the test programs and test data will be downloaded to the processor’s memory.

And then, the processor executes the test programs at full speed. These programs execute appropriate instructions to achieve the same effect as activating faults by test vectors.

Finally, the test response is analyzed by processor online or uploaded to the storage module.

Compared with other test methods, the advantages of SBST are as follows.

TABLE I
INSTRUCTIONS CATEGORY

Category	Instructions
Memory Access	LD, ST
Operation	ADD, ADDI, SUB, XOR…
Branch Jump	BZ

i. SBST is non-invasive. It does not require any additional hardware (which is unacceptable for deep optimized circuits, such as processors), and it has no additional power consumption compared with the normal mode of processors.

ii. SBST supports online testing. It is adopted at normal frequency of processor which makes it possible to detect delay faults that are not visible at low frequency.

iii. SBST avoids over-testing. It only detects the fault that occurs during the normal function phase of the processor. So the yield of products can be improved by adopting SBST.

iv. SBST is flexible and easy to reconstruct. After the test program is optimized, the test flow can be easily changed by loading new test program and test data.

v. SBST can be applied to the whole life cycle of processors. Testing at the manufacturing stage can be reused throughout the product life cycle.

Considering the requirement of module testing for avionics controller, such intermittent faults sensitive, real-time fault detection, low area overhead, etc., SBST is adopted as the test method in this design. Although SBST is mainly employed for processor testing, we draw on its ideas and apply it to test the LVDT module.

III. THE ARCHITECTURE OF ON-LINE TEST SYSTEM

The circuit under test in this paper is a part of an LVDT module for avionics controller. The LVDT is a linear displacement transducer which has many advantages such as high accuracy, wide linear range and excellent repeatability. So it is widely applied to measure displacement, speed and acceleration [11]. As a terminal to acquire information for aero engine controller, the LVDT demodulation circuit is a prerequisite for stable, accurate and fast processing, decision and feedback of the control system. It is composed of four modules including an IIR filter, two 32-bit multipliers and a high-speed dual-port RAM with a width of 16 bits and a depth of 256 bits. The testing process of these circuits will be described in detail in next section.

In order to implement the on-line LVDT module test structure proposed by this paper, a microprocessor is needed as a control module for the whole test procedure. MIPS (Microprocessor without Interlocked Piped Stages) is based on the reduced instruction set architecture [12]. It has relatively simple structure and supports pipeline execution, so it is capable of fulfilling parallel assignments. Due to the above features, we employ the 16-bit MIPS microprocessor as the CPU in proposed SBST scheme.

The upper part of Fig. 1 shows the structure of MIPS microprocessor we apply. It can be seen that the MIPS processor is based on Harvard architecture and it has a five-stage pipeline. The five units are fetch unit, decoder unit, execution unit, memory accessing unit and write back unit.

The function of each unit is explained as follows:

TABLE II
THE CIRCUITS REPRESENTED BY THE FIRST 8 BITS ADDRESS

Address	Target Circuit
8’b00000001	Memory
8’b00000011	Multiplier
8’b00000111	IIR Filter
8’b00001111	Controller

● Fetch unit: This unit reads instructions from ROM which storing instructions.

● Decoder unit: This unit parses the instructions transmitted from the fetch unit.

● Execution unit: This unit executes operations according to the instruction.

● Memory accessing unit: This unit is responsible for reading or writing RAM.

● Write back unit: This unit is used to write data back to register.

This microprocessor has a total of 8 16-bit general registers named as R0~R7. In particular, register R0 has a special characteristic that no matter what data is written to register R0, the data read from register R0 is always 0.

The length of each instruction is 16 bits and the instructions executed in on-line test structure can be divided into the following categories as Table I shows.

Fig. 1 shows the architecture of the on-line test system based on SBST. It can be seen that the system consists of three parts which are the MIPS microprocessor, the bus controller and the target circuit. The MIPS microprocessor is responsible for generating test vectors, applying test vectors on target circuit, collecting test response and online analysis of the test response. The key design is to set up a data path and a control path between the MIPS microprocessor and target circuit, so that the MIPS microprocessor can accomplish the whole test process of the circuit under test.

When the MIPS microprocessor is accessing memory, the address width is 16 bits. But in fact RAM only needs 8-bit address width and the first 8 bits are 0. Considering that we have different modules to be tested, we can make use of the first 8 bits to decide which module to be tested. Table II lists the example of decoded test targets by the first 8 bits address.

According to the instructions, the bus controller chooses the modules and transmits data and signals between the

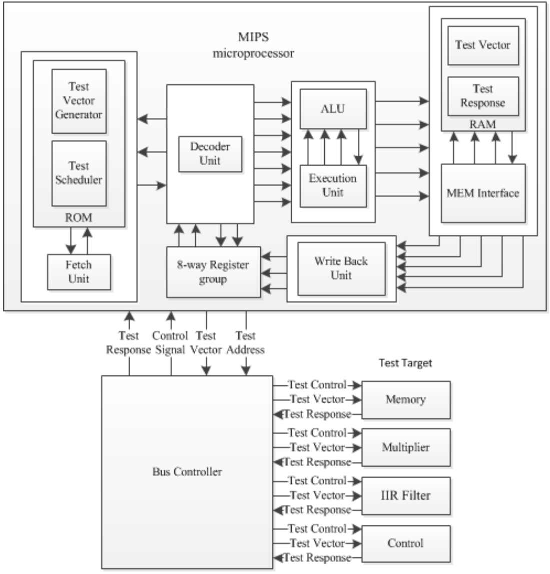


Fig. 1. Structure of Test System

TABLE III
TEST FOR OPERATION ADD

Operation	Test Incentive *a*	Test Incentive *b*	Ideal Response *r*
ADD	1010101010101010	0101010101010101	1111111111111111
ADD	0101010101010101	1010101010101010	1111111111111111
ADD	0000000000000000	0000000000000000	0000000000000000

microprocessor and the module. On the basis of the signal width required by the circuit under test, the bus controller configures the data transferred from microprocessor according to the circuit under test. For example, if we have to configure 32 bits of data to execute a test operation while the data width of the microprocessor is 16 bits, then we need to configure 2 instructions at a time. So we design an identification number for the circuit to be tested. The identification number is generated by microprocessor and only the correct number can activate the corresponding test application.

When the test process is activated, the identification number may be 0 or 1. If the number is 0, the bus controller receives the first 16-bit data. On the contrary, the bus controller receives the last 16-bit data. After two times of configuration, the bus controller will send the whole 32-bit data to the circuit. Similarly, the response date will be stored in the registers of bus controller temporarily and wait for the microprocessor to fetch the data. The logic to fetch data can also be designed according to the requirement of data width. The configuration process can be adjusted through the modification of processor program, which contributes the flexibility of proposed test structure.

IV. IMPLEMENT OF ON-LINE TEST

The whole testing process based on SBST is divided into two parts: the first part is the self-test of MIPS microprocessor, and the second part is the test of other modules in LVDT circuit. The bus controller is responsible for data interaction and control signal transfer [13].

A. Test for MIPS

The MIPS microprocessor controls the process of the test system and is the core module of SBST. So it is necessary to test the microprocessor to guarantee the completeness and correctness of the test system.

Above all, we divide the MIPS microprocessor into 4 parts: memory, ALU, register group and pipeline. According to the functions and characteristics of each module, we adopt relevant test programs for different modules based on the instruction set of the MIPS microprocessor. Then, we will introduce the test program for each module to realize the self-test of microprocessor.

1) Memory

We employ the March C+ algorithm to test RAM. The March C+ algorithm is based on March C algorithm and it adds a read operation to the original design. Compared with March C algorithm, the March C+ algorithm can be used to test stuck-open fault and the fault coverage of the March C+ algorithm for RAM is over 95% [14]. In addition, the March C+ algorithm can detect stuck-at faults, transition faults, address decoder faults, coupling faults and neighborhood pattern sensitive faults of decoder circuit, memory data register and memory address register.

The March C+ algorithm is represented by the element symbol as *b(w0) ↑(r0,w1,r1) ↑(r1,w0,r0) ↓(r0,w1,r1) ↓(r1,w0,r0) ↓(r0)* [15]. The character *b* means choosing ascending or descending order randomly, *w* means writing, *r* means reading, ↑ means ascending order, and ↓ means descending order.

When executing test program, the MIPS microprocessor will read and write all memory cells in ascending order or descending order according to the test program based on March C+. By comparing the data written to RAM and read from RAM, the MIPS microprocessor is able to judge which memory cell has a fault.

2) ALU

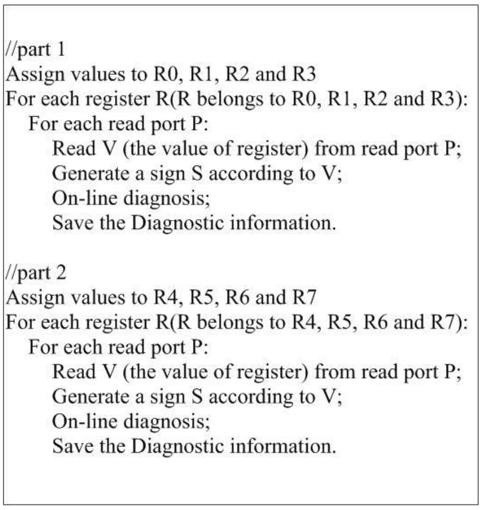
The ALU occupies a considerable proportion of the hardware space of the microprocessor and has a higher contribution to the fault test coverage. Also, the SBST method needs the ALU to execute right operations to guarantee the correctness of test. Therefore, the test of ALU is an important part of the self-test of the microprocessor.

The ALU of the MIPS microprocessor has 8 kinds of operation which are ADD, SUB, AND, OR, XOR, SL, SR and SRU. The inputs of ALU are two16-bit numbers named *a* and *b*. The output is a 16-bit number named *r*.

We take operation ADD as an example. The assembly code designed to test operation ADD is "ADD R3, R1, R2, 0". 3 groups of test stimulus are needed and Table III shows the test stimulus and ideal response. During the process of testing, the microprocessor will compare the value of register R3 with the ideal response to judge whether the circuit is working normally.

3) Register Group

The test data of SBST are stored in registers and the test program designed to test registers is divided into 2 parts. In each part, half of the registers are in the test mode while another 8 registers are used to control the test execution. By taking such measures, we can use all the 8 registers of the microprocessor during the test without adding additional hardware resources. The test program presented in the form of pseudocode is showed in Fig.2.



```
//part 1
Assign values to R0, R1, R2 and R3
For each register R(R belongs to R0, R1, R2 and R3):
    For each read port P:
        Read V (the value of register) from read port P;
        Generate a sign S according to V;
        On-line diagnosis;
        Save the Diagnostic information.

//part 2
Assign values to R4, R5, R6 and R7
For each register R(R belongs to R4, R5, R6 and R7):
    For each read port P:
        Read V (the value of register) from read port P;
        Generate a sign S according to V;
        On-line diagnosis;
        Save the Diagnostic information.
```

Fig. 2. Test Program for Register Group

In the initialization stage, we need to assign different values to registers with one encoding range, so we can detect stuck-at faults when executing read or write operation. After the initialization, the microprocessor should execute some instructions to activate all the data path of register group. Part of the assembler commands are as follows:

```
[1]XOR   R5 = R0, R1
   XOR   R6 = R2, R3
[2]XOR   R5 = R1, R0
   XOR   R6 = R3, R2
```

R0 is the first operand in the first part, but in the second part R0 is the second operand. The two instructions seem similar, but in fact the value of R0 is read from two different read ports of the register. So that, all the data paths accessible to register group will be tested. By comparing the values of R5 and R6 with ideal response, the microprocessor can verify whether there are faults in the circuit.

4) Pipeline

The traditional development of SBST method focuses on the function modules of the microprocessor and ignores the test of pipeline unit. Actually, the average fault coverage of function modules can be more than 90%, while the fault coverage of pipeline unit is lower than 80%. Although the logic of pipeline occupies relatively small hardware size, it is critical to guarantee the microprocessor function. So we design a test program specially to test the processor's pipeline.

The fault of pipeline is related to data dependency of the overlap instructions. Providing the assumption that instruction *b* is executed after instruction *a*, instruction *b* should read the value of certain register only after *a* finishes the write operation on it. Otherwise, it will cause the function error and this data dependency is defined as the read-after-write hazard. Some hazards can be solved by sending the address information to the decoder unit in advance (forward path method). For other unsolved hazards, the following instruction execution will suspend until the write operation finishes.

To verify the function of pipeline, we should firstly find out all the hazards that can be solved or cannot be solved. Secondly, we should activate all the conditions which may suspend the pipeline. Thirdly, we should activate all the different forward paths to the pipeline level, and confirm whether the design of pipeline functions normally. An example is provided for demonstration as follows.

If there are two instructions *Ia* and *Ib*. *Ia* generates results at the pipeline level *p* and saves the results at the pipeline level *s* ($1 \leqslant p \leqslant s \leqslant n$). *Ib* reads data at the pipeline level *d*, but it uses the data in pipeline level *f* ($1 \leqslant d \leqslant f \leqslant n$).

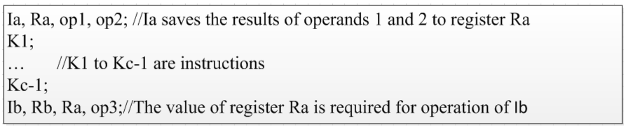
Then, the following operations are illustrated in Fig.3.

Auxiliary Theorem I:

(a) If $s\text{-}d < c$, then there is no risk of data.

(b) If $s\text{-}d \geqslant c$, then there is risk of data.

i. If $p\text{-}f \geqslant c$, then the risk cannot be eliminated and the pipeline suspends.



```
Ia, Ra, op1, op2; //Ia saves the results of operands 1 and 2 to register Ra
K1;
...      //K1 to Kc-1 are instructions
Kc-1;
Ib, Rb, Ra, op3;//The value of register Ra is required for operation of Ib
```

Fig. 3. Instructions

ii. If $p\text{-}f < c$, then the risk is eliminated through the forward path.

Based on Auxiliary Theorem I, we propose a method to test pipeline hazard and forward path. As the Fig. 4 shows, we take the five-level pipeline processor as an example.

It can be seen from Fig. 4 that the instruction LD gains the result at the memory-access level (*p*=4) which is the third clock, and then saves the result to the register R1 at the write-back level (*s*=5) which is the fourth clock. The instruction ADDI reads the register's value at the decode level (*d*=2) which is the second clock cycle, but it executes the operation of ADD at the execute level (*f*=3) which is the third clock. The second instruction needs the data of R1 in the third clock but the value of R1 is updated in the next clock. Therefore, the calculation result will be different from the ideal result. At this moment, the hazard detection module of the processor will be activated to eliminate this problem by suspending the pipeline operation.

B. Test for LVDT Modules

As the test of microprocessor, we divide the circuits into two parts. One is memory which belongs to general module like RAM and the others are special modules such as IIR Filter and LVDT control module.

1) Memory

We test the RAM of target circuit still by using the March C+ algorithm. The process of test is the same as the test of MIPS's RAM.

2) IIR Filter and LVDT Control Module

Like the other modules, the SBST applies test programs to detect faults of IIR Filter and LVDT control model. However, the IIR Filter and LVDT control module are not general logic circuits, and it is relatively difficult to find an efficient algorithm to design test programs for those circuit, so we determine to test them by inputting test vector. The test for IIR Filter and LVDT control module are the same, so we take the IIR Filter as an example. There are mainly two ways to generate test vectors for integrated circuit. One adopts linear shift register to generate pseudorandom test vectors and the other applies ATPG (Automatic Test Pattern Generation) algorithms to generate test vectors. The number of pseudorandom test vectors increases exponentially with the expansion of the test scale while the ATPG algorithm can reach high fault coverage with less test vectors, so we employ the software called TetraMAX to generate test vectors for special modules. The MIPS microprocessor will be in charge of inputting the test vectors.

The IIR Filter is tested by inserting scan chains and applies Synopsys's DFT Compiler and TetraMAX for testing development. DFT Compiler replaces the ordinary register in the circuit with the shift register which supports the function

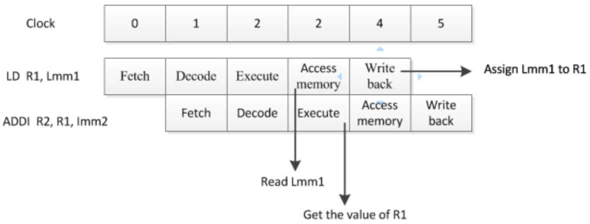


Fig. 4. Process of Instruction Execution

TABLE IV
COMPARISON OF 1 SCAN CHAIN AND N SCAN CHAINS

Test Module (IIR Filter)	Test Time /clock cycle
1 scan chain	560
n scan chains (n=22)	140

of scanning, and these shift registers increase the controllability and observability inside of the circuit.

To design test program for the special module under test, we adopt the following test process:

First of all, we should import the circuit's Verilog file into DFT Compiler. Then DFT Compiler will analyze the internal structure of the circuit. After that, the software will set test constraints and replace ordinary registers with shift registers. At last, it will output the netlist file and test constraint file of the circuit which has been inserted scan chains.

Secondly, we should import the netlist file and test constraint file generated by DFT Compiler into TetraMAX, then set up the ATPG model of the circuit. Next, we should set up the type and format of the test vector and the ATPG tool will generate the test vector file.

Thirdly, the test vectors generated by the TetraMAX will be applied on the VCS software platform for simulation to ensure the reliability of the test vectors.

Finally, we will design test programs according to the tested vectors which are validated.

Scan design is a testability method which is widely applied, and it is proposed for complex sequential circuit testing. The general scan design only inserts one or several scan chains, so each scan chain has several scan registers, and the input and output of data need to be transmitted in series. In order to simplify the design of the bus controller and make full use of the parallelism of microprocessor, we set up 22 scan chains for IIR filter which has 22 scan registers in all. So each scan chain has only 1 register and all the inputs and outputs of registers will be parallel, which decreases the testing time and simplifies the test control logic.

The parameter of two means to insert scan chains is shown in Table IV, which proves that inserting *n* scan chains greatly shortens the time of register configuration. As a result, the total test time is shortened.

C. Test System Scheduling

As a complete system, the scheduling of the system modules also plays an important role. In order to guarantee the correctness of the test results, it must be ensured that the test data can be transmitted correctly and orderly between modules. As we can see that the bus controller is responsible for data interaction and control signal transfer between the microprocessor and the LVDT demodulation circuit.

The test process is divided into 3 steps and the first step is to input test vectors. The MIPS microprocessor reads the test program and utilizes Memory Write Instruction (ST) to access the bus interface. After receiving the test vectors, the bus interface inputs the test vectors to the circuit according to the test order. The second step of testing is to retrieve response data. The response data is stored in the registers of the bus interface and the microprocessor applies the instruction LD to fetch the data. And the last step of the test is to compare the response data retrieved from the module with ideal response online and determine whether the circuit has fault.

V. EXPERIMENTAL RESULTS

The hardware simulation platform adopted in our scheme is Xilinx Kintex-7 development board. The software platforms are ModelSim, Xilinx's ISE, Synopsys's Design Complier and TetraMAX. We employ ModelSim and ISE to emulate Verilog codes and Synopsys's EDA software to insert scan chain (DFT Compiler), generate test vectors (TetraMAX) and verify simulation results (VCS).

A. Self-test of MIPS Microprocessor's RAM

According to the March C+ algorithm, we design the test program for RAM and implement hardware simulation on FPGA. Fig 5 illustrates the simulation waveforms of March C+ algorithm and it describes the whole process of testing which mentioned in Section IV. The MIPS completes the entire test by executing the instructions in ROM. The signals captured by Chipscope from FPGA are showed in Fig 6. This picture shows one operation (read 0, write 1 and read 1) of March C+ algorithm for one memory cell. Signal *mem_write_en* controls the reading and writing of RAM. Signal *ex_alu_result* is responsible for memory address. Signal *mem_read_data* shows data which are read from RAM. Signal *mem_write_data* provides data to be written to RAM. Initially, *ex_alu_result* sends read address BA46h. In the next clock cycle, the microprocessor reads data 0000h from the address. Then, the microprocessor pulls up enable signal *mem_write_en*. Signal *mem_write_data* sends the data FFFFh and *ex_alu_result* sends the write address BA46h. The data is written to RAM. Finally, signal *ex_alu_result* sends read address BA46h and the data read from the address is FFFFh as the signal *mem_read_data* shows. The waveform of Fig 6 is consistent with the algorithm flow and verifies the validity of the simulation. So the test program of March C+ is proved to be correct.

B. Self-test of MIPS Microprocessor's Register Group

The simulation waveform of register group from ModelSim and FPGA are separately showed in Fig 7 and Fig 8. We assign values to *reg_array0*, *reg_array1*, *reg_array2* and *reg_array3*. The microprocessor will firstly calculate the value of *reg_array0* XOR *reg_array1* and store the value to *reg_array6*. Then the microprocessor will calculate the value of *reg_array2* XOR *reg_array3* and store the value to *reg_array7*. Finally, the microprocessor will compare the value of *reg_array6* and *reg_array7* by XOR gate. If these two values are equal, the circuit being tested has no fault. Because the simulation waveform and real waveform are the same, the test program realizes the proposed function.

C. Self-test of MIPS Microprocessor's Pipeline

Under normal circumstances, the signal *pipeline_stall_n* remains at 1. When the microprocessor's hazard_detection module is activated, the *pipeline_stall_n* will be pulled down. And the signal *pc* represents the instruction that the microprocessor is executing. In order to test pipeline, we must activate the function of the pipeline logic. Fig 9 shows the waveforms of *pipeline_stall_n* and *pc* captured by Chipscope from FPGA. As shown in the Fig 8 that when *pc* is

02h, 06h, 08h and 0Bh, signal *pipeline_stall_n* is 0 and the microprocessor will delay certain time to execute the instruction. In other cases, *pipeline_stall_n* is 1 and instructions are executed normally according to the clock frequency. The waveforms show that in the former case, the hazard_detection module is activated correctly and functions normally. So the pipeline test program is able to achieve the goal.

D. Test Results of Target Circuit's

Fig 10 shows the test report of IIR Filter generated by TetraMAX. It can be seen that the ATPG tool generates 20 test vectors for a total of 630 faults and the test coverage is 100%. Fig 11 shows the test report of LVDT control module generated by TetraMAX. The ATPG tool generates 74 test vectors for a total of 1416 faults and the test coverage is also 100%.

Figure 12 demonstrates the process of input of a vector which is applied to activate the reset function of the IIR Filter module. It is consist of two stages. The first stage is assigning values to the scan registers. When signal *test_se* is high, signal *EXP:ai* assigns the scan register a value of 22'h1278fc on the rising edge of *clk*. We observe that the value of output signal *EXP:so* also changes to 22'h1278fc. The second stage is inputting a set of combinatorial logic signals to observe the change of the circuit. Signal *test_se* is pulled down to prevent interference of the scan design. After this vector activates the reset function of the circuit, the microprocessor inputs another set of combinatorial logic signals through *EXP:ai* on the rising edge of the signal *rst_n*. It is illustrated that the data of signal *EXP:so* changes from 22'h1278fc to 22'h000000 and the circuit is reset successfully. By inputting other test vectors generated by TetraMAX in the same way, the circuit can be tested.

E. Joint Simulation of Test System

Figure 13 illustrates that the workflow of the test system which is divided into four stages. Above all, the MIPS microprocessor sends the test vectors to the bus controller by two continuous instructions as signal *instruction* shows in the picture. Secondly, the bus controller splices the two sets of test vectors and takes control of the test process. In the third stage, the enable signal *test_se* is raised, which means that the internal registers can be configured by the input of scan chain. At the same time, the configuration data is imported through one rising edge of *iir_clk*. In the fourth stage, signal *test_se* is pulled down so that the circuit is not affected by the scan input. At the same time, the input data is loaded to the circuit port through one rising edge of *iir_clk*, and the output data is saved by the bus controller. The MIPS microprocessor will retrieve the response data back to the processor through two consecutive instructions and determine whether there is a fault in the circuit by comparing the data online.

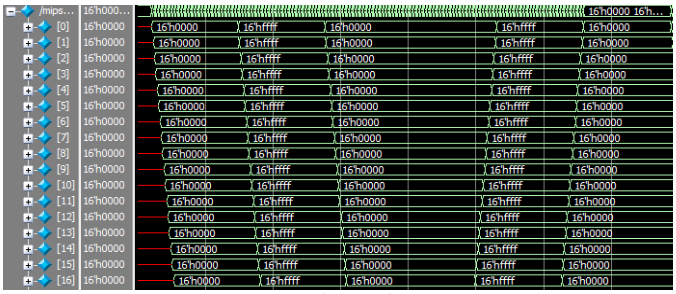


Fig. 5. Simulation waveforms of March C+ algorithm

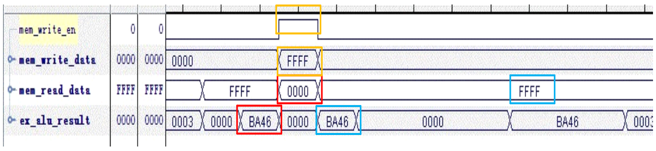


Fig. 6. Waveforms of writing and reading RAM from FPGA

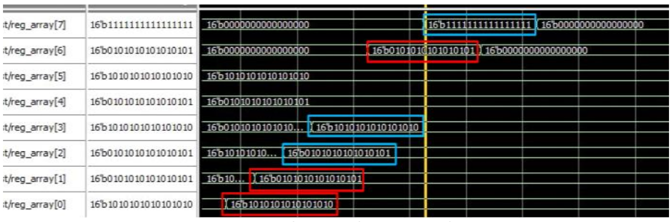


Fig. 7. Simulation waveforms of registers from Modelsim

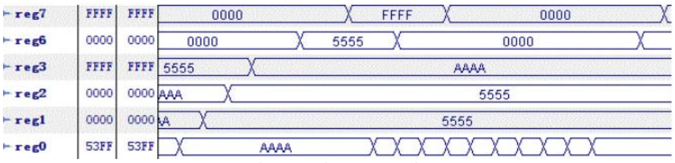


Fig. 8. Waveforms of registers from FPGA

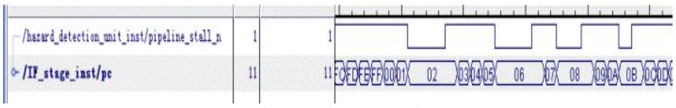
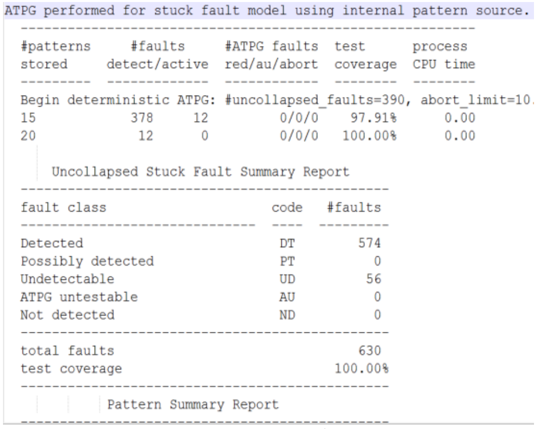


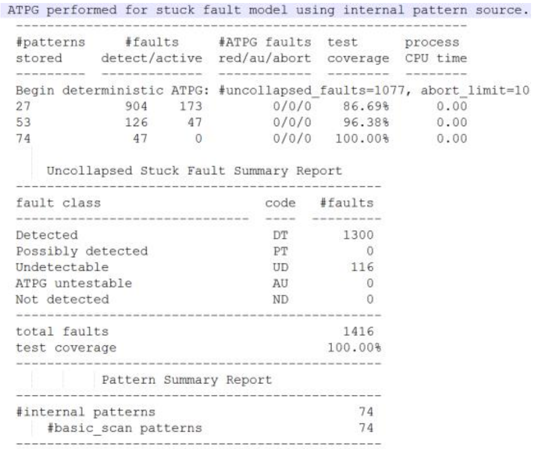
Fig. 9. Waveforms of pipeline from FPGA



```
ATPG performed for stuck fault model using internal pattern source.
 ----------------------------------------------------------------
 #patterns     #faults      #ATPG faults  test      process
 stored      detect/active  red/au/abort  coverage  CPU time
 ---------  -------------  ------------  --------  --------
 Begin deterministic ATPG: #uncollapsed_faults=390, abort_limit=10.
 15             378      12         0/0/0    97.91%      0.00
 20              12       0         0/0/0   100.00%      0.00

    Uncollapsed Stuck Fault Summary Report
 -----------------------------------------------
 fault class                     code   #faults
 ------------------------------  ----  ---------
 Detected                         DT        574
 Possibly detected                PT          0
 Undetectable                     UD         56
 ATPG untestable                  AU          0
 Not detected                     ND          0
 -----------------------------------------------
 total faults                               630
 test coverage                          100.00%
 -----------------------------------------------
           Pattern Summary Report
 -----------------------------------------------
```

Fig. 10. Test Report of IIR Filter



```
ATPG performed for stuck fault model using internal pattern source.
 ----------------------------------------------------------
 #patterns     #faults     #ATPG faults  test      process
 stored      detect/active  red/au/abort  coverage  CPU time
 ---------  ------------  ------------  --------  --------
 Begin deterministic ATPG: #uncollapsed_faults=1077, abort_limit=10
 27            904    173        0/0/0    86.69%      0.00
 53            126     47        0/0/0    96.38%      0.00
 74             47      0        0/0/0   100.00%      0.00

     Uncollapsed Stuck Fault Summary Report
 -----------------------------------------------
 fault class                     code   #faults
 ------------------------------  ----  ---------
 Detected                         DT       1300
 Possibly detected                PT          0
 Undetectable                     UD        116
 ATPG untestable                  AU          0
 Not detected                     ND          0
 -----------------------------------------------
 total faults                              1416
 test coverage                          100.00%
 -----------------------------------------------
            Pattern Summary Report
 -----------------------------------------------
 #internal patterns                          74
     #basic_scan patterns                    74
 -----------------------------------------------
```

Fig. 11. Test Report of LVDT Control Module

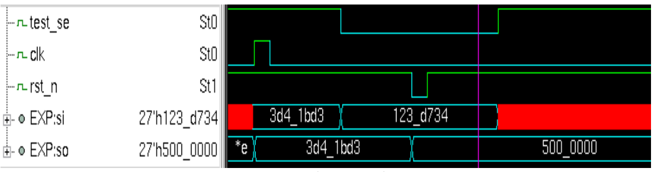


Fig. 12. Simulation waveforms of applying test vector

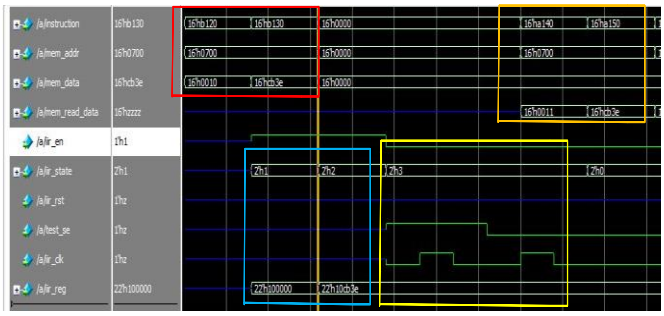


Fig. 13. Joint simulation of the test system

VI. Conclusion

With the increase of the circuit complexity and the need of at-speed testing, the SBST method will become a promising solution to integrated circuit on-line testing. The test structure we present in this paper is a typical example of an SBST application. We can choose the optimal test method or test algorithm for each module such as March C+ and DFT mentioned before, so that each module is able to achieve ideal test coverage. Microprocessor-based online testing is also more convenient and efficient. By writing a test program, the system can be tested at-speed and each test module can be scheduled and managed. Compared with the traditional test methods, the test structure proposed in this paper is capable of achieving higher efficiency and flexibility, which are extremely important for circuits calling for high reliability. However, it may increase resource overhead as well. We hope to further improve the test structure and achieve better balance on efficiency and overhead.

References

[1] Jervan G, Eles P, Zebo Peng, et al. "Hybrid BIST time minimization for core-based systems with STUMPS architecture". *Proc. of IEEE International Symposium on Defect and Fault Tolerancein VLSI Systems*, 2003, pp. 225–232.

[2] J án Hudec. "An Efficient Adaptive Method of Software-Based Self Test Generation for RISC Processors". *4th Eastern European Regional Conference on the Engineering of Computer Based Systems*, 2015, pp. 119-121.

[3] J. Shen, J. A. Abraham. "Native Mode Functional Test Generation for Processors with Applications to Self-Test and Design Validation". *Proceedings International Test Conference*, 1998, pp. 990–990.

[4] P. Parvathala, K. Maneparambil, W. Lindsay. "FRITS: A Microprocessor Functional BIST Method". *Proceedings International Test Conference*, 2002, pp. 590-598.

[5] S. Gurumurthy, S. Vasudevan, J. A. Abraham. "Automatic Generation of Instruction Sequences Targeting Hard-to-Detect Structural Faults in a Processor". *Proceedings International Test Conference*, 2006, pp. 776–784.

[6] Ching-Wen Lin, Chung-Ho Chen. "Processor shield for L1 data cache software-based on-line self-testing". *22nd Asia and South Pacific Design Automation Conference*, 2017, pp. 420-425.

[7] R. K. K. R. Sandireddy, V. D. "Agrawal. Diagnostic and detection fault collapsing for multiple output circuits". *Design, Automation and Test in Europe, 2005*, pp. 1014-1019.

[8] Hassan Ebrahimi, Alireza Rohani, Hans G. Kerkhoff. "Detecting intermittent resistive faults in digital CMOS circuits". *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2016 IEEE International Symposium on*, 2016, pp. 87-90.

[9] M. J. Y. Willaims, and J. B. Angell. "Enhancing Testability of Large-Scale Integrated Circuit via Test Points and Additional Logic". *IEEE Transactions on Computers*, 1973, pp. 46-60.

[10] Mihalis Psarakis, Dimitris Gizopoulos, Ernesto Sanchez, et al. "Microprocessor Software-Based Self-Testing". *IEEE Design & Test of Computers*, 2011, pp. 4-19.

[11] K V Santhosh, Akanksha. "Design of an adaptive soft calibration circuit for LVDT". *International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques*, 2017, pp. 524-528.

[12] Agineti Ashok, V. Ravi. "ASIC Design of MIPS Based RISC Processor for High Performance". *International Conference on Nextgen Electronic Technologies*, 2017, pp.263-269.

[13] Jin Zhengfei, Zhang Ying, Chen Xin, Ge Fen, "A Novel On-Line Test Scheme for Avionics Controller Based on SBST". *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2018*, 4-6 July, 2018, London, U.K, pp. 241-246.

[14] L. Chen. "Software-Based Self-Testing with Multiple-Level Abstractions for Soft Processor Cores". *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol.15, no.5, 2007, pp. 505–517.

[15] Xu Chuanpei, Tao Yi, Wan Chuntingl. "BIST method of SRAM for network-on-chip". *12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, 2015, pp. 558–562.