ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 4, December 2014

# Robust Fault Detection for VLIW Processors Using Software Based Self Testing

D.Jeyamani Latha[1], P.Hannah Lincy Merlin[2]

Associate Professor, ECE, SCAD College of Engineering and Technology, Tirunelveli, India[1]
Student, M.E. VLSI Design, SCAD College of Engineering and Technology, Tirunelveli, India [2]

***Abstract***— VLIW (Very Long Instruction Word) processors are adopted in several products especially for embedded applications. So the problem of testing the processor is important whether it contain any faults or not. So the efficient and optimal test techniques are needed for detecting the faults in functional units of a VLIW processor. Structural Software Based Self Test (SBST) is an effective solution to detect the faults in a processor during the operational life. In structural based SBST the post silicon test validation are used to detect the fault in memory, stuck at faults in ALU (Arithmetic and Logical Unit) of VLIW processor. The automatic generations of efficient test patterns are given to the memory unit and the fault is detected. By introducing the input test into the ALU, the soft errors such as stuck at 0 and stuck at 1 fault are detected. Also the intermittent fault like glitches is detected.

**Keywords***:* VLIW processor, test patterns, software based self test, stuck at fault.

## I. INTRODUCTION

Mainly due to the continuous scaling in the semi-conductor manufacturing process and to the increasingly high operational frequency of integrated circuits, processor chips face growing testability problems. Moreover, since the production processes are highly stressed, phenomena like metal migration or aging of the circuit may increase the occurrence of permanent faults in the systems, even during the circuit operational phase. For these reasons, new test solutions are being investigated in order to provide high fault coverage with acceptable costs (e.g., in terms of test time, silicon area over-head and required test infrastructure). A promising approach for processors and processor-based systems (e.g., systems on a chip, or SoCs) corresponds to the so-called Software-Based Self-Testing (SBST).The basic idea is to generate test programs to be executed by the processor and able to fully exercise the processor itself or other components in the system and to detect possible faults by looking at the produced results. One of the main advantages of SBST lies in the fact that it does not require any extra hardware; therefore the test cost is reduced and any performance or area penalty is avoided. The SBST approach allows at-speed testing and can be easily used even for on-line testing [5]. For these reasons, SBST is increasingly applied for processors and SoC testing, often in combination with other approaches. Among the various microprocessor architectures, very long instruction word (VLIW) processors were demonstrated to be a viable solution especially for applications demanding high performance while exposing a considerable amount of parallelism, such as several digital signal processing algorithms used in multimedia and communication applications. VLIW processors are currently adopted in several products, in particular for embedded applications, and the problem of testing them is increasingly relevant [2]. In this paper, post silicon test validation is proposed to find the faults occur in the circuit. This scheme helps to identify the faults after manufacturing phase. The paper is organized as follows: Section 2 explains about the previous works on software-based self testing. Section 3 deals with the main design. Results are discussed in Section 4.

## II. RELATED WORK

Self testing is an important method in the wireless sensor system or any other digital system may be victimized to various faults like structural, environmental, etc. Therefore, it is very important that the system is tested and diagnosed and tested often during the lifetime of the system. It is also very important that the test and diagnosis is fast and fault coverage should be high. Therefore BIST is specified as one of the system's functions. The test functions are localized to the circuit which facilitates high speed testing and also it reduces the test application time. It also provides an easy access to the system components and interconnections. Faults can be defined as physical or logical defects in either the design or implementation of a particular device. Faults may induce errors. Under an error condition, the states of the system may

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 4, December 2014

be incorrect. Errors may induce failures. The method proposed in [6] aims at automatically generating the test program for a given VLIW processor starting from the VLIW manifest and from a library of existing SBST programs. The generation of the test program is automatically performed on the basis of the VLIW configuration, and is therefore autonomously tuned depending on the VLIW manifest features. Monitoring the system for the detection of faults may be either internal or external. Internal monitoring is done with the help of software and external monitoring is done with the help of hardware. In [3],focused on a specific issue which must be faced when testing a VLIW processor: the register file characteristics are different than in other processors, since it must be accessed from different domains. In [1], Identify the testability hotspots in the pipeline logic using two fully pipelined RISC. A systematic SBST methodology enhances existing SBST program so that test the pipeline logic is performed. This result forms the input to the methodology and retrieves the test development efforts behind pre-existing SBST programs. Generic solutions that can be applied to test the pipelined logic of any processor. The proposed solution is integrated in a systematic SBST methodology which adapts available SBST programs and enhances them to achieve high fault coverage for the pipelined logic. In [4], a software methodology is proposed for VLIW processors for detecting faults based on the execution of each operation twice on two different FUs exploiting the idle computational resources and checking the redundant results through control instructions. This approach is a valid solution for detecting both permanent and temporary faults exploiting only the ISA of the VLIW processor, but the performance degradation, mainly due to the checking instructions, and the code growth (more than 100% on the considered benchmarks)are significant.

*A. Overview Of Vliw Processor*

Reconfigurable processors are increasingly used in different domains. The key characteristic lies in the fact that they can be easily configured to match the specific requirements of the target application e.g., in terms of performance, size, and power consumption, thus possibly making them more convenient than traditional processors. Whereas conventional processors mostly only allow programs that specify instructions to be executed one after another, a VLIW processor allows programs that can explicitly specify instructions to be executed at the same time (i.e. in parallel).

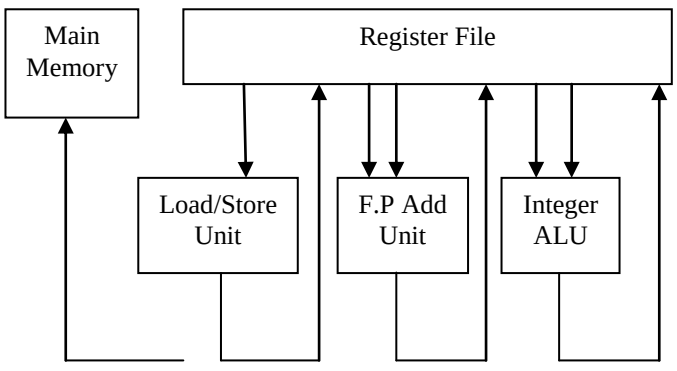


**Fig. 1 VLIW processor**

Traditional approaches to improving performance in processor architectures include breaking up instructions into sub-steps so that instructions can be executed partially at the same time (known as pipelining), dispatching individual instructions to be executed completely independently in different parts of the processor like superscalar architectures and even executing instructions in an order different from the program. These approaches all involve increased hardware complexity. This type of processor architecture is intended to allow higher performance without the inherent complexity of some other approaches to work. The VLIW approach, by contrast, depends on the programs themselves providing all the decisions regarding which instructions are to be executed simultaneously and how conflicts are to be resolved. As a practical matter this means that the compiler (software used to create the final programs) becomes much more complex, but the hardware is simpler than many other approaches to parallelism. Before executing any operations in parallel the processor must verify that the instructions do not have interdependencies. For example, if a first instruction's result is used as a second instruction's input then they cannot execute at the same time and the second instruction can't be executed before the first. Modern out-of-order processors have increased the hardware resources which do the scheduling of instructions and determining of inter dependencies. The VLIW approach, on the other hand, executes operations in parallel based on a fixed schedule determined when programs are compiled. Since determining the order of execution of operations (including which operations can execute simultaneously) is handled by the compiler, the processor does not need the scheduling hardware that the three techniques described above require. As a result, VLIW CPUs offer significant computational power with less hardware complexity than is associated with most superscalar CPUs.

## III. MAIN DESIGN

In main design, how the patterns are generated and identify the faults in the circuit. It is explained with fig.2

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 4, December 2014

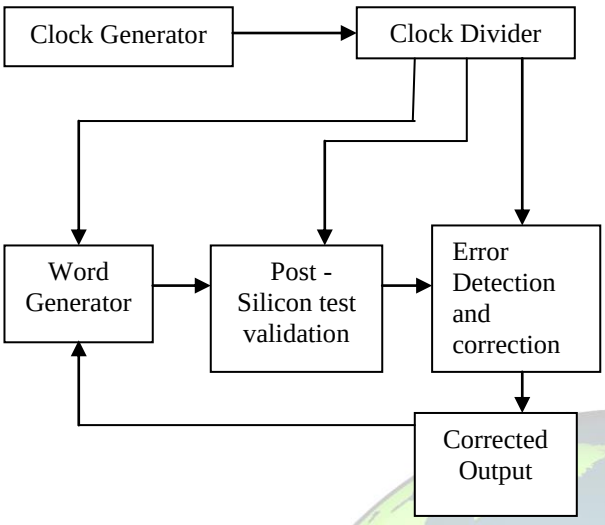


**Fig. 2 Memory Testing**

A failure is said to have occurred in a circuit or system if it deviates from its specified behavior. Faults in a circuit occur due to defective components, breaks in signal lines, shortened to ground or power supply, short circuiting of signal lines, excessive delays. A fault is characterized by its nature, value, extent and duration. The nature of a fault can be classified as logical or non-logical. A logical fault causes the logic value at a point in a circuit to become opposite to the specified value. Non-logical faults such as malfunction of the clock signal, power failure and so on. The duration of a fault refers to whether the fault is permanent or temporary.

*(A) Software Based Memory Testing*

The purpose of a memory testing is to confirm that each storage location in a memory device is working. Problems with the electrical connections to the processor will cause the memory device to behave incorrectly. Data may be stored incorrectly, stored at the wrong address or not stored at all. It can be explained by wiring problems on the data address and control lines. If the problem is with a data line, several data bits may appear to be "stuck together" (i.e., two or more bits always contain the same value, regardless of the data transmitted). Similarly, a data bit may be either "stuck high" (always 1) or "stuck low" (always 0). These problems can be detected by writing a sequence of data values designed to test that each data pin can be set to 0 and 1, independently of all the others. If an address line has a wiring problem, the contents of two memory locations may appear to overlap. In other words, data written to one address will actually overwrite the contents of another address instead. This happens because an address bit that is shorted or open will cause the memory device to see a different address than the one selected by the processor. Another possibility is that one of the control lines is shorted or open. The operation of many control signals is specific to the processor or memory architecture.

The basic idea behind any memory test is to write some set of data to each address in the memory device and verify the data by reading it back. If all the values read back are the same as those that were written, then the memory device is said to pass the test. As a result there is no error if there is any mismatch of data while reading data from memory then the error availability is possible. A clock generator is a circuit that produces a timing signal (known as a clock signal) for use in synchronizing a circuit's operation. Figure 2 shows how the memory is tested.

*(B)Design of word generator*

This module is used to generate a pattern of words. The Linear Feedback Shift Register (LFSR) consists of EXOR gates and D-flip flops. It is a shift register that, when clocked; the outputs of two or more flip flops are combined in XOR configuration and feeding those outputs back into input of one of the flip flop. A linear LFSR is widely used as test pattern generator because of its small circuit area and excellent random characteristics. When the outputs of the flip-flops are loaded with a seed value (anything except all 0s, which would cause the LFSR to produce all 0 patterns) and when the LFSR is clocked, it will generate a pseudorandom pattern of 1s and 0s. Only signal necessary to generate the test pattern is the clock.

***Steps for post silicon validation***

1. The patterns (64 bits) are generated in word generator by using LFSR.
2. The generated patterns are stored in the memory unit.
3. Then read the patterns from the memory.
4. If all the values read back are the same as those that were written, then there is no error in memory unit.
5. If there is any deviation in the data stored then the error is occurred.
6. So error correction logic is to be performed i.e. shifting operation is occurred till all the bits changed to 1.
7. Finally the corrected output is feed back again.

*(C) Generation of Stuck at faults*

The post silicon test is used to test the Stuck at fault such as Stuck at zero(S-A-0) and Stuck at one (S-A-1). It is referred to as permanent faults. The manufacturing defects (such as a shorted transistor or an open metal line) that cause an input or an output of logic function to be permanently stuck at a high or low logic level.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 4, December 2014

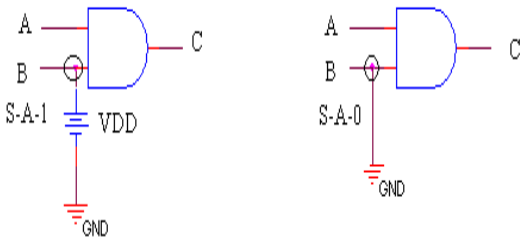


**Fig. 3 Stuck at faults**

Figure 3 illustrates the two possible physical defects, namely a short to the power shown on the left and a short to the ground on the right, causing a permanent logic value at line A. Structural software-based self-test strategies are characterized by a component oriented test development approach, fine-tuned to the low, gate level details of the processor core. Pseudorandom patterns or patterns generated by an automatic test pattern generator (ATPG) are produced for each of the processor components. The patterns are generated by using Linear Feedback Shift Register. The patterns are stored in a register file of the VLIW processor. Then the testing process is involved to detect the fault in the memory unit. It considers the processor structure so the structural fault coverage should be increased. It doesn't require manual intervention in some steps of the test generation so automation is limited. So the SBST detect the permanent faults and temporary faults in efficiently.

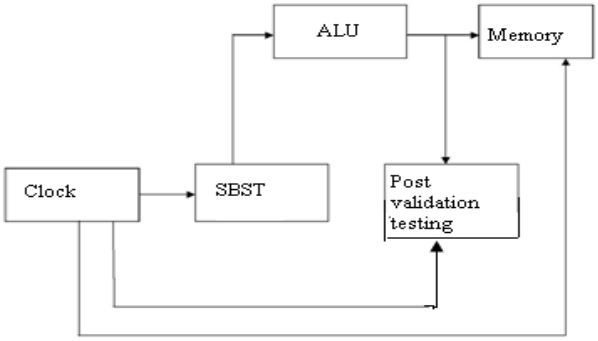


**Fig. 4 Stuck at fault detection**

Post silicon validation is used to detect the functional and electrical errors that have escaped the pre-silicon validation phase. Electrical faults are soft errors and crosstalk faults. Soft errors are specified as stuck at faults and crosstalk faults are specified as glitches. Both affect the correct functionality of the chip. So this post silicon validation is used to capture the bugs. The test vectors from the SBST unit is given to the ALU and ALU checker verify the circuitry in it and the fault is detected and it was shown in the figure 4.

***Steps for detecting the stuck at faults***

1. The 4 bit input signals are given to a0, a1, a2, a3 shown in figure 5
2. Determine the output of the circuit for the specified input.
3. Modify the Boolean functionally of the gate whose input has the fault i.e. give complementary signals to any of the input signals.
4. If the output of normal circuit varies from the one with fault, then the fault is detected.

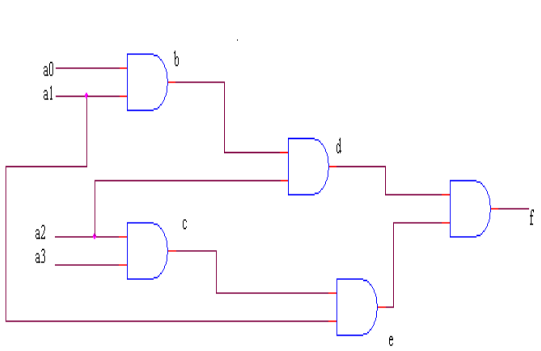


**Fig. 5 Logic gate diagram for fault free circuit**

5. Steps 2 to 5 are repeated for another fault in the circuit. This continues till all faults are considered in the circuit.

Table I(A)
TEST PATTERN TO TEST MULTIPLE FAULTS

| Test inputs | a0,a1,a2,a3 | Faults Detected |
|---|---|---|
| 1 | 1 0 0 0 | S-A-1 at output of all nets |
| 2 | 1 1 1 1 | S-A-0 faults in all the nets of the circuit |

Glitches are the spurious transitions which occur due to difference in arrival times of signals at the gate inputs. There have been a number of attempts made in the past to eliminate these spurious transitions. These are not needed for the correct functioning of the logic circuit. Power consumed

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 4, December 2014

by glitches is called as glitch power. The glitches producing circuit is shown in the figure 6. A glitch happens sometimes as signals propagate through combinational logic. Before reaching the final steady state value, a wrong intermediate value produced in the output during the settling phase

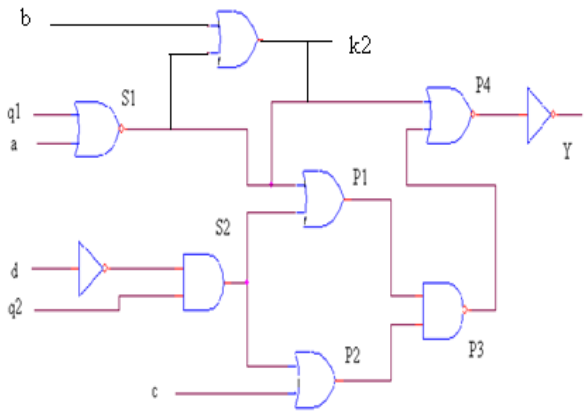


**Fig. 6 Glitch producing circuit**

By giving inputs to a,b,c,d,q1,q2 the output y is calculated using the above logical gate diagram. Error signal is created by the AND combination of buffer control signal and output. Glitches will be occurred in the circuit.

## IV. Simulation Results

Post silicon test validation is used to identify the faults. 'a' is a 4 bit value (a0,a1,a2,a3) then using proper logic function mentioned in the algorithm calculate the value for b, c, d, e, f. In the next step take the complementary values for any one output signals i.e. change the boolean function of any one gate producing outputs. The values changed from 0 to 1 means stuck at 1 fault occur and from 1 to 0 means stuck at 0 fault occur. Figure 7 shows the stuck at fault generated in the circuit.

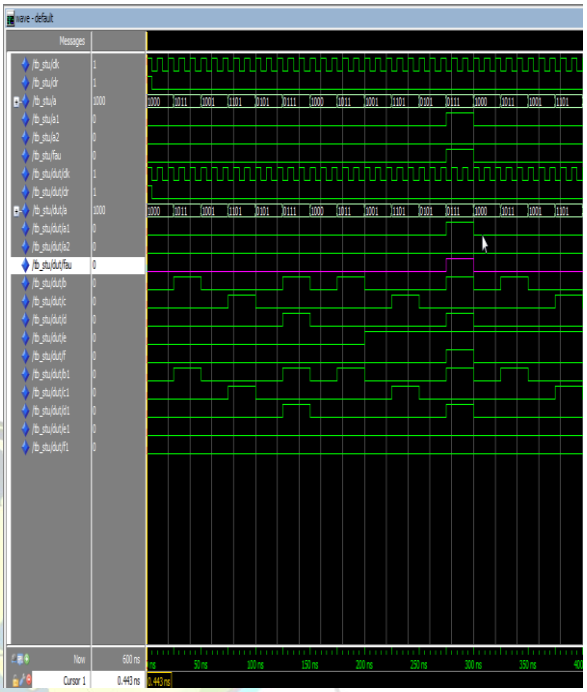

**Fig. 7 Simulation result for stuck-at-fault**

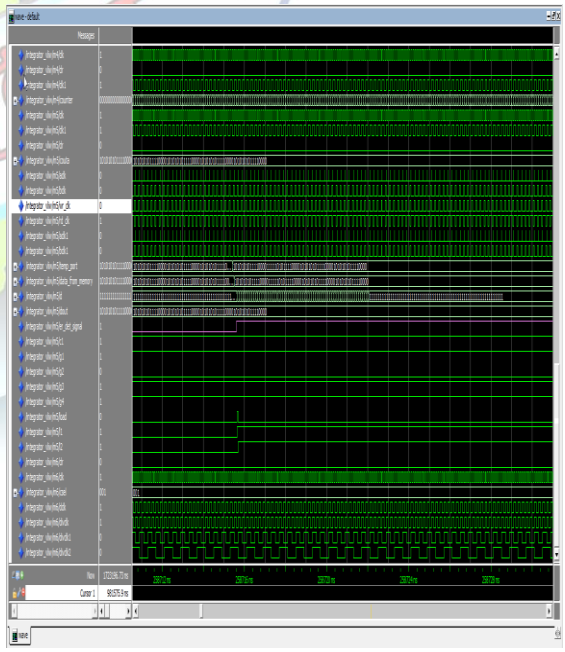

**Fig. 8 Simulation result for error detection and correction**

Figure 8 shows the simulated output for error detection and correction. Give the input bits to input count i.e. 64bits [1010….1111]. If the data from memory also have

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 4, December 2014

same input (i.e.) 1010….1111 means there is no error so the error detecting signal is low. For simulation we have to generate an error so assign for (e.g.) 1111….10111 to temp port (count/cout & data from memory). The values in the temp port are different from the input patterns. Then the error detecting signal is high. Finally shifting takes place until all bits changed into 1.

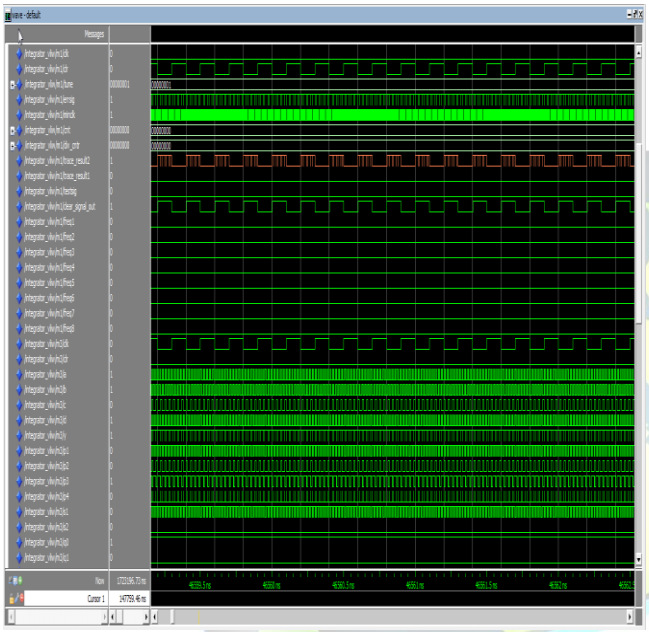

**Fig. 9 Simulated output for glitch**

By giving inputs to a,b,c,d,q1,q2 the output y is calculated using logical gates. Error signal is created by AND combination of buffer control signal and output. Providing period and duty cycles to the clock input. Because of the change in the duty cycles this fault will occur and it is shown in the figure 9.

## V. CONCLUSION

The proposed Software Based Self Test (SBST) of processor cores in Very Long Instruction Word (VLIW) processor can be very effective, low cost strategy for off-line testing. Faults can be detected by generating patterns.The proposed system effectively achieve permanent faults as well as temporary faults in Arithmetic and Logical Unit (ALU) and register file of the VLIW processor with reduced power consumption and delay as compared to Built In Self Test.

## ACKNOWLEDGEMENT

I would like to articulate my profound gratitude and indebtedness to Assoc. Prof. D.Jeyamani Latha, for guiding and encouraging me in all aspects.

## REFERENCES

[1] D. Gizopoulos, M. Psarakis, M. Hatzimihail, M. Maniatakos,A. Paschalis, A. Raghunathan, and S. Ravi, "Systematic software based self-test for pipelined processors," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 11, pp. 1441–1453,Nov. 2008.

[2] M. Psarakis, D. Gizopoulos, E. Sanchez, and M. Sonza Reorda,"Microprocessor software-based self-testing," *IEEE Design Test Comput.*,vol. 2, no. 3, pp. 4–19, May–Jun. 2010.

[3] D. Sabena, M. Sonza Reorda, and L. Sterpone, "A new SBST algorithm for testing the register file of VLIW processors," in *Proc. IEEE Int.Conf. Design, Autom. Test Eur.*, Mar. 2012, pp. 412–417.

[4] M. Ulbricht, M. Scholzel, T. Koal, and H. T. Vierhaus, "A new hierarchical built-in self-test with on-chip diagnosis for VLIW processors,"in *Proc. IEEE Symp. Design Diag. Electron. Circuits Syst.*, Apr. 2011,pp. 143–146.

[5] E. Sanchez, M. Sonza Reorda, and G. Squillero, "On the transformation of manufacturing test sets into on-line test sets for microprocessor," in*Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, Oct. 2005,pp. 494–502.

[6] Davide Sabena, Matteo Sonza Reorda, (2014) 'On the Automatic Generation of Optimized Software-Based Self-Test Programs for VLIW processors' *IEEE Trans. Very Large Scale Integr,(VLSI)* vol. 22, no. 4, pp.813-823.

[7] G.Xenoulis, D.Gizopoulos, N.Kranitis, and A. Paschalis, "Low-cost online software-based self-testing for embedded processor cores," in *Proc.IEEE Int. On-Line Testing Symp.*, 2003, pp. 149–154.

[8] Tai-Hua Lu, Chung-Ho Chen, and Kuen-Jong Lee, "Effective hybrid self test program development for software based self-testing of pipeline processor cores." in *Proc.IEEE Int VLSI. vol. 19, no. 3, March 2011.*