ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

A Comparative Study on Low Energy Optimization Techniques in WSN

Vinitha Ann Regi[1], Mujeebudheen Khan[2]

Department of Information Technology,

Rajagiri School of Engineering & Technology, Rajagiri Valley P O, Kochi-39 [1, 2]

Abstract: Sensor networks are mostly distributed over a large area which may sometimes be harsh and uneven. Therefor the system must be tolerant to faults raised by the environment and also other intermittent problems. Therefore it is very important that the sensed values and readings of each and every sensor node must be accurate and reliable. Therefore, for the purpose of validation, various online and offline methods are used. ATE is a hardware test used to test the readings. But ATE capabilities can't match the test requirements in terms of volume and speed. BIST is a test embedded within the system itself. But it has certain disadvantages like hardware and area overhead and also elevated power dissipation. Therefore SBST plays an important role in testing the sensor node. The SBST routines are used for this purpose. They can either be downloaded to the nodes or they may be available in the flash memory also. Various optimization methodologies have been discussed in the paper**.**

Keywords: WSN, optimization, SBST

I. INTRODUCTION

A Wireless Sensor Network consists of a number of wireless sensor nodes which is distributed over a geographical location. It may consist of a few to a large number of nodes. They are bound to measure variables like temperature, pressure, sound, etc. It has extensive applications in military applications, surveillance, etc. The sensor hardware consists of a radio transceiver, microcontroller, battery. The transceiver consists of an internal antenna or connection to an external antenna. The various applications are area monitoring, environmental monitoring, air quality monitoring, air pollution monitoring.

The main features of a WSN are:

- Power consumption constrains for nodes using batteries or energy harvesting
- Ability to cope with node failures
- Mobility of nodes
- Communication failures
- Heterogeneity of nodes
- Scalability to large scale of deployment
- Ability to withstand harsh environmental conditions

A sensor network consists of a base station or a sink. All the nodes collect the specified data on an individual basis, and then they are sent to the base station. A base station appropriately processes the data and obtains the corresponding results. The nodes assemble together and form a network topology. If one of the nodes fails, then all other nodes rearrange themselves to form a new topology.

One of the major constraints of a sensor node is the power. Therefore, every node must try to optimise the power so that the lifetime of the system can be increased. The basic architecture of a wireless sensor node consists of various layers which are physical layer, data link layer, network layer, transport layer, application layer. It is depicted below in Fig 1.

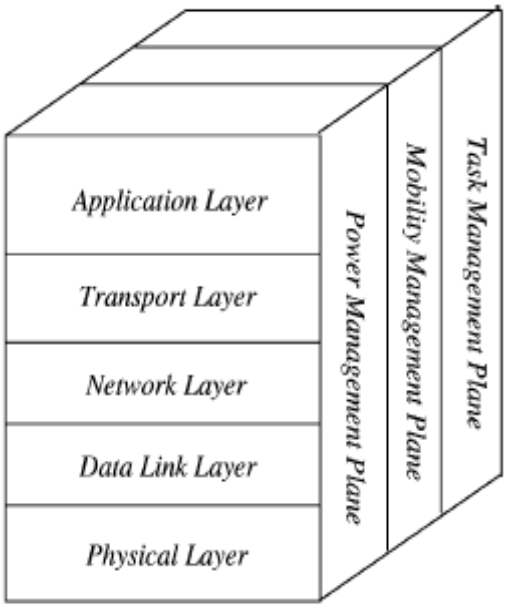

Fig .1. WSN Architecture

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

The paper deals with various optimization methodologies to optimize the power consumption and also the accuracy of the sensed readings and values by a sensor node.
The rest of this paper is organized as follows. Section 2 provides an idea on the existing solutions. Section 3 elaborates the current methodology. Conclusion and future work is in section 5.

II. EXISTING SOLUTIONS

1. AUTOMATIC TEST EQUIPMENT (ATE)

ATE is one of the methods for testing the electronic circuits. It can be termed as an external method. The circuit-under-test can be either the entire chip or just a part of the chip. For the purpose of testing, there should be inputs and processing of the inputs it should produce outputs. The input can be termed as input test vectors. They are binary patterns which can be applied to the CUT and the associated output responses are the values observed on the outputs of the CUT. Then the output responses passed through a comparator where they are compared against the expected responses. If all the output responses match the corresponding expected responses, then the system is marked as fault-free. When ATE is used, the input test vectors and the correct response data are stored in ATE memory. As the technology advances and the CUT grow more complex, the ATE capabilities can't match the test requirements in terms of volume and speed. The limitations of ATE paved way for Built-In-Self-Test (BIST).

2. BUILT-IN-SELF-TEST (BIST)

The main testing method incorporated is the hardware based test is termed as Built-In-Self-Test (BIST). BIST is the embedded hardware tester. Self testing is an important method as 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. There are mainly two types of BIST methodology. They are online and offline methodologies. 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 be incorrect. Errors may induce failures. Under a failure situation, the behaviour of the system may deviate from the expected behaviour. The failure may induce a hazard. Faults can be categorized into three: Design, Fabrication, and Operational. Design faults are those faults in the system's hardware. Fabrication faults are those faults due to the defects in the manufacturing process. Operational faults occur during the normal operation of the system and the factors may be environmental or some related problems. 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.

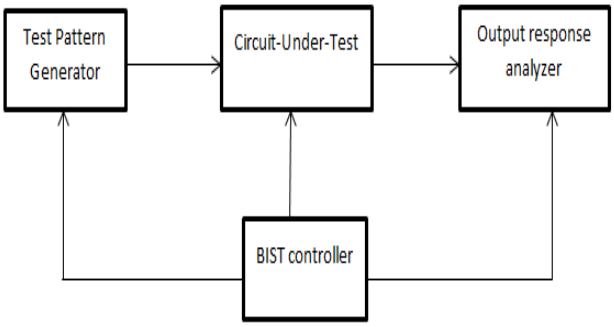

Fig .2. BIST

BIST can be either online or offline.

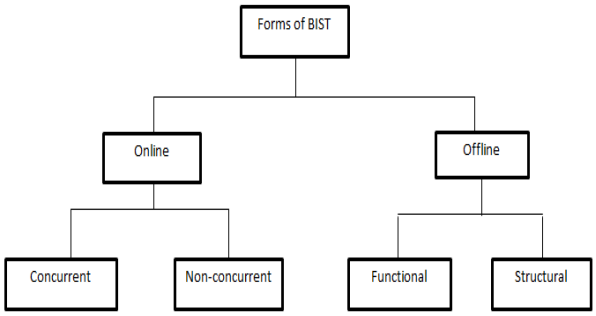

Fig .3. Forms of BIST

A. *Online testing*

Online testing is done during the normal operation of the system. It detects operational faults. The main aim of the online test is to diagnose errors and if diagnosed, it should take appropriate corrective actions. Online tests can be divided into two: Concurrent and non-Concurrent online testing. The concurrent testing works simultaneously with the other normal operations of the system whereas the non-Concurrent testing is idle during the normal operations of the system.

a) Non-concurrent testing:

It can be triggered either by event or by time. During the operation of the system, changes can happen to state of the system or key events may occur. This triggers non-concurrent testing which detects permanent faults.

b) Concurrent testing:

The major drawback of non-concurrent testing is that transient or intermittent faults whose effects disappear quickly cannot be diagnosed and hence it may affect the system adversely. Concurrent testing rectifies this drawback as it can diagnose such faults which during the operational life cycle of the system. For more efficiency, it

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

can be associated with non-concurrent testing or diagnostic software.

B. *Offline testing:*

Offline testing is done when the function of the system is suspended. It can be divided into two: Structural and Functional.

1) Structural testing:

It is based on the structure of the circuit-under-test (CUT).

2) Functional testing:

It is based on the functional description of the circuit-under-test (CUT).

BASIC ARCHITECTURE OF BIST

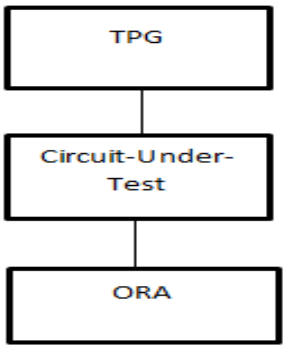

Fig .4. Architecture of BIST

BIST Test Structures are

a) TPG – test pattern generator
b) PRPG – pseudorandom PG
c) LFSR – Linear feedback shift register- commonly used PRPG
d) SRSG – shift register sequence generator- single output PRPG
e) ORA – (generic) output response analyser
f) SISR – single input signature register
g) MISR – multiple input signature register
h) BILBO – Built in Logic Block Observer- non-simultaneous PG & SA
i) CBILBO – Concurrent BILBO- simultaneous PG & SA

The advantages of BIST are:

a) fast, efficient and hierarchical
b) BIST is a cheap test solution
c) No need of expensive ATE
d) Testing during operation and maintenance
e) Uniform technique for production, system and maintenance tests.
f) Dynamic properties of the circuit can be tested at speed
g) Support concurrent testing
h) Can be used for delay testing

The disadvantages of BIST are:

a) Additional BIST hardware and area overhead
b) Performance degradation, timing problems
c) Additional delay and elongated critical path
d) Elevated power dissipation

III. THE CURRENT METHODOLOGY

The software based self test (SBST) has turned as an effective way for processor manufacturing testing which is more cost effective than BIST. In Software-Based Self-Testing, the execution of a program takes place on the embedded microprocessor. Thus, structural faults can be identified using functional internal resources. The basic idea behind the SBST is to effectively make use the Instruction Set Architecture (ISA) and the various resources embedded on the chip. It does not require any hardware modification of the processor. Initially the self-test program would be downloaded into the internal instruction memory. The self-test program is initially downloaded into the internal instruction memory and then to a low-cost external equipment. Then, it is executed by the microprocessor and then computes and stores self-test responses in the data memory. Finally, the external equipment brings back those self-test responses for evaluation.

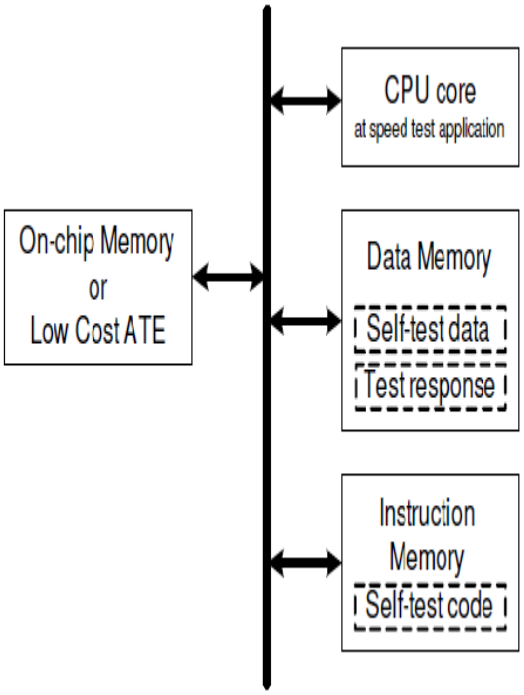

Fig .5. Architecture of SBST

It effectively utilizes the power as it operates in the functional model of processors. Thus various intermittent problems like performance degradation, quality degradation can be avoided thus resulting in high test quality. Also the fault coverage is high in the case of SBST.

Different methodologies have been proposed for the development of self-test programs. In [1], [2] and [3] the authors suggested methodologies based on Instruction Set Architecture (ISA). In [4], the authors proposed structural component-oriented approach. In [5] the authors proposed specific self test routines based on loop of instructions.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

The wireless sensor nodes may be deployed in a harsh or a hostile environment. Thus testing the accuracy of the sensor node readings is very important in order to validate the same. In such cases online SBST can be used for improving the reliability of the system. For that purpose various optimization methodologies have been proposed.

Energy Optimisation Methodologies

In [MT SBST], the author proposed multithreaded (MT) SBST methodology. It generates an efficient multithreaded version of the test program and schedules the resulting test threads into the hardware threads of the processor to reduce the overall test execution time and on the same time to increase the overall fault coverage. In [10.1], the author proposed an SBST scheme that utilizes existing CPU instructions efficiently by taking the least amount of cycles and selecting operands with least Hamming distance and weight to minimize the overall energy consumption of the test code. SBST approaches applied to the microprocessor can be classified into two. They are structural and functional methodologies. The structural methodology is structural in nature and the functional methodology is functional in nature. In [12], the authors described methods for optimising the energy of the sensor nodes. The methodology described consists of four steps. Various SBST routines are available to test the validity of the sensed data. Here it is assumed that the SBST routines are not available at the memory. But the various cases considered are

1) The SBST routines are located in a specific position in the flash memory of the WSN node and therefore routines cannot be updated and thus downloading is irrelevant.
2) The WSN node has some free space in its flash memory where the SBST routines reside. The routines can be updated a number of times during the lifetime of the application. The new routines should also fit in the available space in the flash memory. The number of updates is considerably smaller than the number of executions of the routines.
3) The WSN node does not have any space in the flash memory and every time the routines are executed they are downloaded from a relay node.

IV. CONCLUSION

WSN nodes are deployed over a geographical area for monitoring various parameters like temperature, pressure, etc. For their energy needs, they primarily depend on battery cells. But once the power dies out, there exists no option other to dispose the sensor node. Therefore, it is very important that the sensor nodes utilize the energy optimally. Various methodologies have been proposed in order to optimize the energy dissipation. ATE, BIST, SBST methods can be used to check the reliability and the accuracy of the system. Among them, the SBST methods are the most reliable. The SBST routines may be initially available in the flash memory. If not, then they can be downloaded to the sensor node from the memory. SBST optimally utilizes the energy of the system. As a future work, the fault coverage of the SBST can further be improved so that the system's inherent and intermittent faults can further be diagnosed and the system can further be improved.

REFERENCES

[1] J. Shen and J. A. Abraham, "Native Mode Functional Test Generation For Processors with Applications to Self-Test and Design Validation". In *Proceedings IEEE International Test Conference (ITC)*, Washington, DC, October 1998.
[2] Ken Batcher and Christos Papachristou, "Instruction randomization Self Test for Processor Cores". In *ProceedingsIEEE VLSI Test Symposium (VTS)*, pages 34.40, Dana Point, CA, April 1999.
[3] F. Corno, M. Sonza Reorda, G. Squillero, and M. Violante, "On the Test of Microprocessor IP Cores". In *Proceedings Design, Automation, and Test in Europe (DATE)*, Munich, Germany, March 2001.
[4] L. Chen and S. Dey. Software-Based Self Testing Methodology for Processor Cores. *IEEE Transactions on Computer-Aided Design*, March 2001.
[5] N. Kranitis, G. Xenoulis, D. Gizopoulos, A. Paschalis, and Y. Zorian. Low-Cost Software-Based Self-Testing of RISC Processor Cores. In *Proceedings Design, Automation, and Test in Europe (DATE)*, Munich, Germany, March 2003.
[6] N. Foutris, M. Psarakis, D. Gizopoulos, A. Apostolakis, X. Vera, A. Gonzalez "MT-SBST: Self-Test Optimization in Multithreaded Multicore Architectures"
[7] M.Psarakis, D.Gizopulos, E.Sanchez, M.Sonza Reorda, Microprocessor Software-Based Self-Testing", *IEEE Design & Test of Computers*, vol. 27, no. 3, pp. 4-18 May/June 2010.
[8] H. Al-Asaad, B. T. Murray, and J. P. Hayes, "Online BIST for embedded systems", *IEEE Design and Test of Computers*, Vol. 15, No. 4, pp. 17-24, November 1998.
[9] P. T. Gonciari, B. M. Al-Hashimi, and N. Nicolici. Test Data Compression: the System Integrator's Perspective. In Proc. of the Design, Automation and Test in Europe Conference, pages 726–731, 2003.
[10] Ch. Scherrer, A. Steininger, *How does Resource Utilization Affect Fault Tolerance?*, IEEE Int. Symp. on Defect and Fault Tolerance in VLSI-Systems, Mt. Fuji, Yamanashi, Japan, October 2000.
[11] Ch. Scherrer, A. Steininger, *How does Resource Utilization Affect Fault Tolerance?*, IEEE Int. Symp. on Defect and Fault Tolerance in VLSI-Systems, Mt. Fuji, Yamanashi, Japan, October 2000.