DIAGNOSTICS SOFTWARE FOR ETHERNET BASED NETWORK INTERFACE CARD

[1]Pruthvi shree E.A, [2] Mrs Meghashree A.C , [3]Dr .G. F. Ali Ahammed

[1]Masters Student, [2]Assistant Professor, [3]Associate Professor and Course Coordinator
[1]Department of Digital Electronics and Communication Systems,
[1]VTU PG Center, Mysuru, India

***Abstract:* In embedded world, NIC (Network Interface Card) is a part of network mainly used for data routing. Diagnostics software is used to qualify the hardware components present in the Ethernet based NIC and detect the manufacturing faults in NIC.NIC is a FPGA based line card, FPGA contains multiple virtual devices. To check the functionalities of virtual devices LINUX based diagnostics software is developed. Communication with the devices can be done by using device drivers and then add the test cases such as access tests, interrupt tests and OHXC bus testing. Diagnostics software tests the newly manufactured system before NIC is used by end user and reduces the complexity of the system and maintenance cost. Diagnostic software increases the reliability of the system.**

***Index Terms* - Diagnostics, Access tests, Interrupt tests.**

I. INTRODUCTION

Distributed applications require rapid and reliable exchange of information across a network to synchronize operations and/or to share data. The performance of these applications depends upon an efficient communication facility. In order to connect computers to a network for providing communication, a network interface card (NIC) is necessary. The NIC contains the electronic circuitry required to communicate using a wired connection (e.g., Ethernet) or a wireless connection (e.g., Wi-Fi).

Fig.1 Network Interface Card

The complexity of the system that we use on a day to day is growing constantly. This trend is strong in the area of embedded systems. So it is better to qualify and detect hardware fault in the network interface card before installing the system in field, basic tests needs to be performed to qualify the NIC. These tests check the basic functionality of hardware components/modules present in the system. If all tests performed on system have successfully passed, the system is ready to use by software. Otherwise, the components/modules for those tests have failed needs to be rectified. The bundle of these tests routine that needs to be perform over system, to qualify it, is called Diagnostics Software. CFPGA is present on the card which contains multiple numbers of virtual devices and communication with the devices can be done using device drivers. A device driver is a software component that lets the operating system and a device communicate with each other.

Diagnostic software packages have numerous test cases to check the functionalities of hardware components. There are two types of tests

1. Control plane test.
2. Data path test.

II. System architecture of diagnostic software

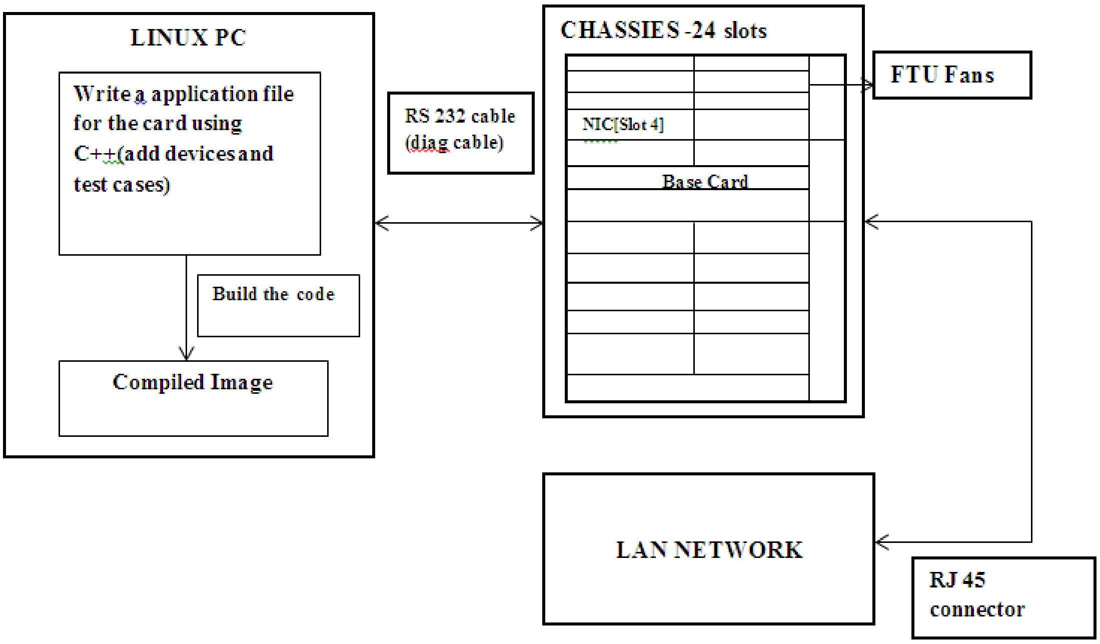


Fig.2 System architecture of Diagnostic software

The above figure shows the system architecture of diagnostic software. Diagnostics software is being executed in two ways that is, U-Boot and LINUX to overcome limitations of U-Boot the Diagnostics Package needs to be ported to LINUX. Write an application file for the card using C++ in LINUX PC , then build the code and get the compiled output this binary file is called as Image.

Connect RS 232 cable one end to the LINUX PC and other end to the chassis. Chassis used in this project contains 24 slots and FTU fans are inserted to slot number 22, 23 and 24.These fans are used for heat absorption in the system. Base card is inserted to slot number 9. Daughter card is a 2 line card means it occupies two slots at a time. Compiled image is loaded on to the card using RS232 cable and shell scripting.

NIC used in this project is an Ethernet based network interface card, therefore RJ45 connector is used. One end of this cable is connected to Ethernet port of the card and another end is connected to LAN. Once the setup is done insert the NIC into particular slot of chassis and load the compiled image onto to card and check the functionalities of hardware components present on the card.

III. Test cases

3.1. Access test

Access test is to qualify the address bus and data bus access to a particular register on a FPGA or any other devices. Each data line going to the device is checked to test whether address bus and data buses are working or not.

Examples: Scratch pad test, FPGA version test, Slot ID test. Toggle LED test and port LED test. Write 0b1111…101…1111 to the scratch pad of the device concerned. Read back the value in the scratch pad and compare, if it is same as written data then the access test is passed. Continue the same procedure for 16 bit and 32 bit scratchpad.

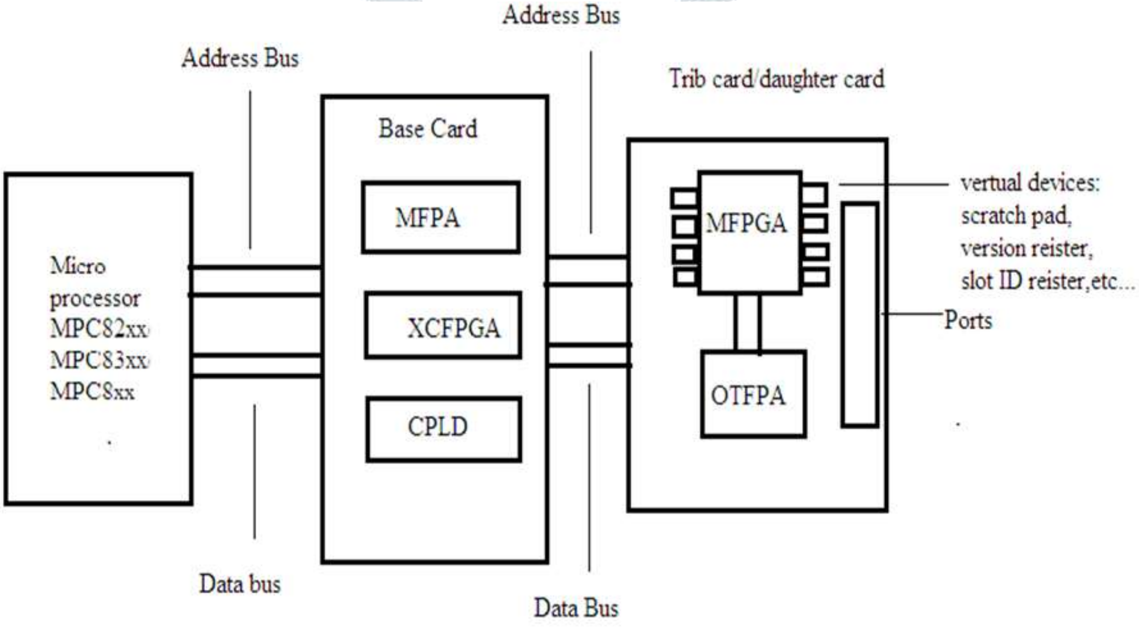

Address Bus
Address Bus
Trib card/daughter card
Base Card
MFPA
XCFPGA
CPLD
Micro processor MPC82xx MPC83xx MPC8xx
MFPGA
OTFPA
vertual devices: scratch pad, version reister, slot ID reister,etc...
Ports
Data bus
Data Bus

Fig.3 Path covered by the access test

3.2 Interrupt test

The main aim of this test is to qualify the IRQ line running from every device on a card chained across FPGA to the processor. This test will be performed by enabling the interrupt from every device and chaining it till the processor end. Finally once the interrupt chaining is proper, following conditions should be checked.

1. Presence of previously reported interrupts. (PENDING 0)
2. Generation of Interrupt. (PENDING 1)
3. Clearing of the interrupt. (PENDING 0)

This chaining will be checked till the processor. Interrupt test for a particular device can be performed in 2 ways i.e, Event generated interrupt test and manual interrupt test

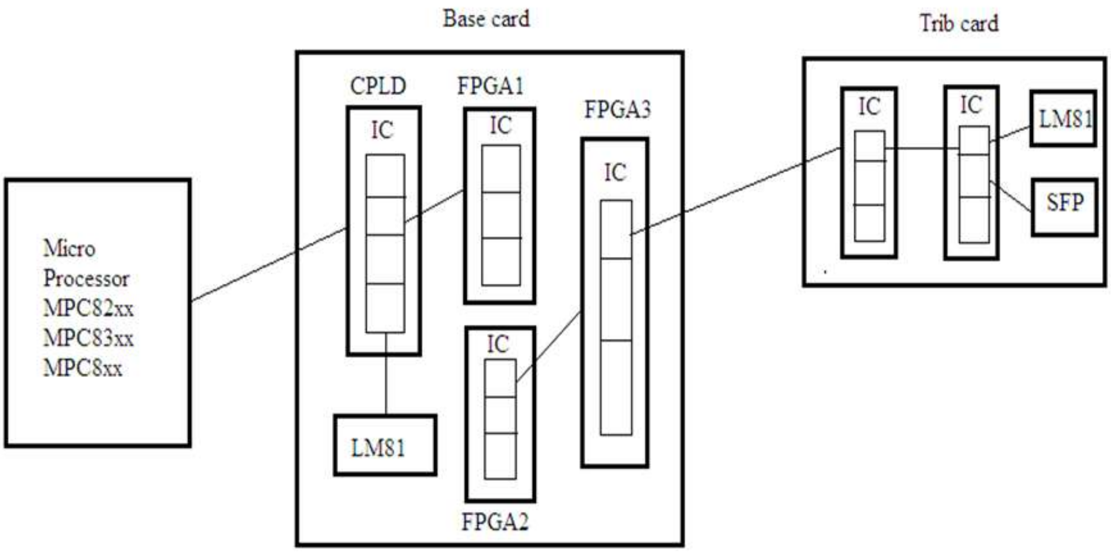
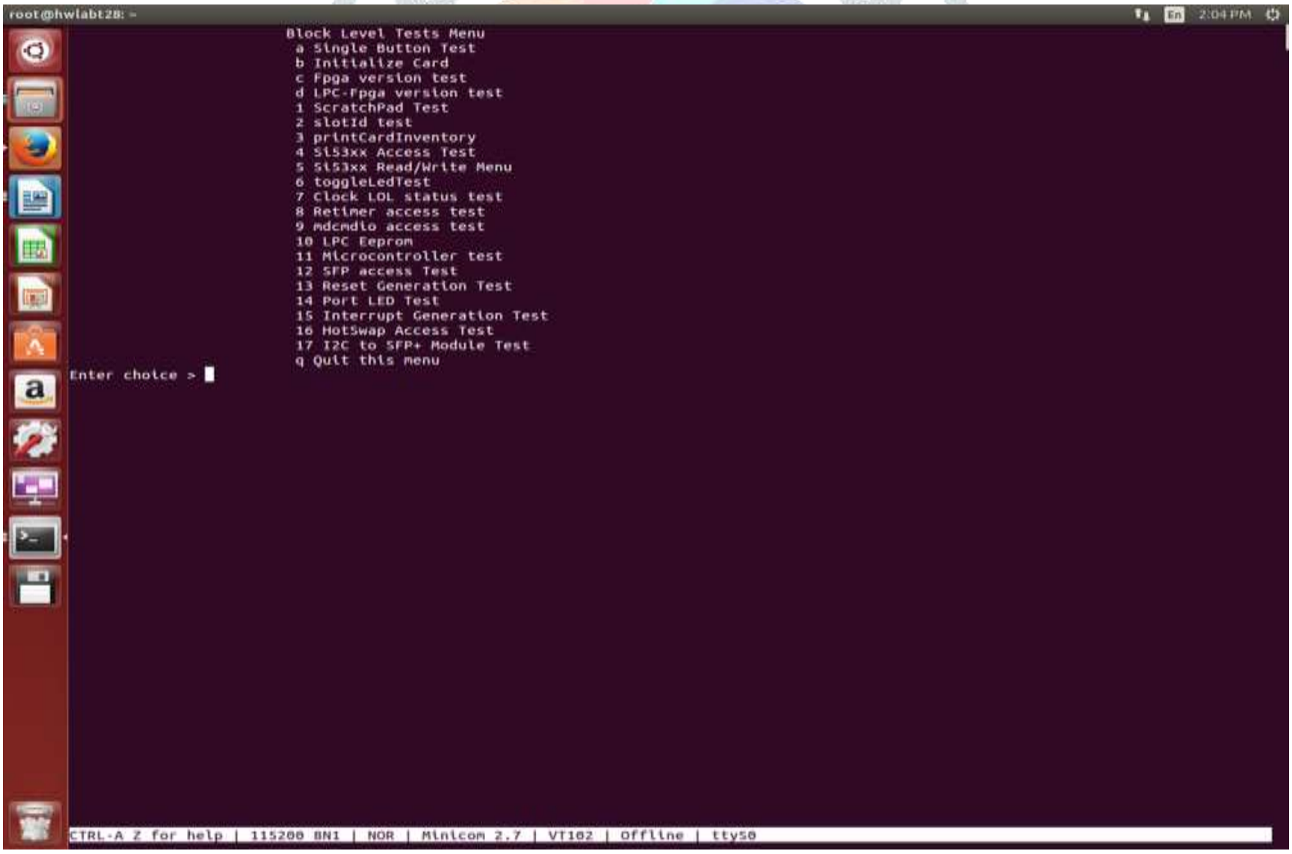


Fig.4 Path covered by the interrupt test

IV. RESULTS

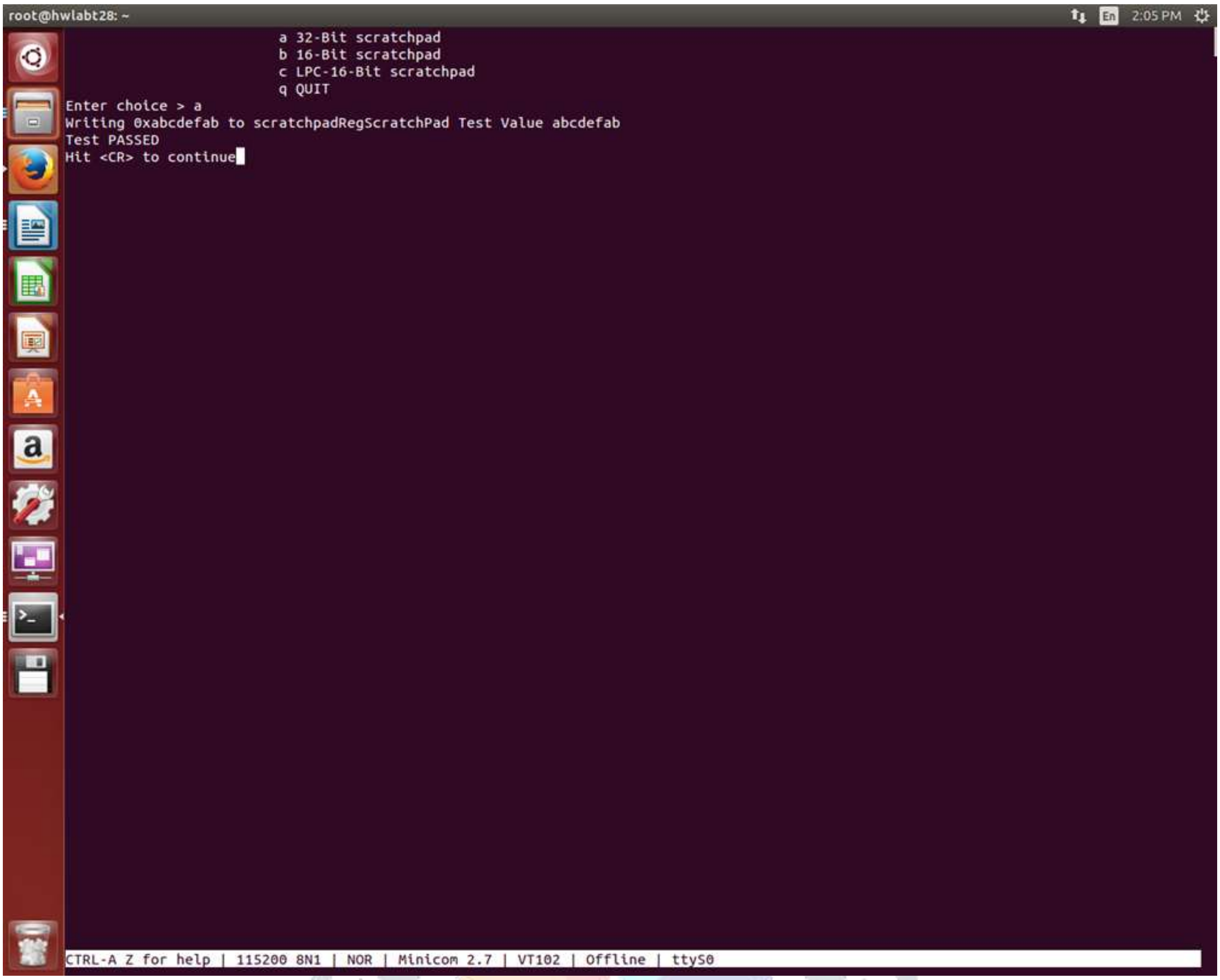

```
root@hwlabt28: ~
```

```
                          Block Level Tests Menu
                           a Single Button Test
                           b Initialize Card
                           c Fpga version test
                           d LPC-Fpga version test
                           1 ScratchPad Test
                           2 slotId test
                           3 printCardInventory
                           4 Si53xx Access Test
                           5 Si53xx Read/Write Menu
                           6 toggleLedTest
                           7 Clock LOL status test
                           8 Retimer access test
                           9 mdcmdio access test
                           10 LPC Eeprom
                           11 Microcontroller test
                           12 SFP access Test
                           13 Reset Generation Test
                           14 Port LED Test
                           15 Interrupt Generation Test
                           16 HotSwap Access Test
                           17 I2C to SFP+ Module Test
                           q Quit this menu
Enter choice > █
```

```
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyS0
```

Fig.5 Screenshot of block level menu

Block level test menu consists of number of test cases which are added in the application file. Select the test cases to check the functionality of the hardware components present in the card.



```
                             a 32-Bit scratchpad
                             b 16-Bit scratchpad
                             c LPC-16-Bit scratchpad
                             q QUIT
Enter choice > a
Writing 0xabcdefab to scratchpadRegScratchPad Test Value abcdefab
Test PASSED
Hit <CR> to continue

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyS0
```

Fig.6 Screenshot of scratch pad test

Scratch pad test is an access test. In 32 bit scratch pad test , write the 32 bit data into the 32 bit scratch pad and, read the scratch pad whether the data sent and data receive is same then the test is successful and scratch pad is working fine.

V. SHARING OF PAST

"Exploring Diagnostic Capabilities of Software-based Self-Tests for production and in-field Applications". In this paper they presented various methods for diagnostic software for manufacturing test and in-field usage in comparison to traditional diagnostic tests with diagnosis on gate level. Further they showed a reduction for the required fault dictionary from the initial manufacturing test approach. This software-based self-tests are available in-field and provide a diagnostic resolution [1].

"Printed circuit board interconnect fault inspection based on eddy current testing". Testing is one of the important stages in the production of electronic products to make sure that system is fully functioning. The reliability of circuit boards such as a Printed Circuit Board (PCB) affects the reliability of a device on which they are mounted. To avoid components from getting damaged and further problems upon power up, the PCB circuit must pass through fault inspection techniques.

In this paper, a single fault (short or open) on a single sided printed circuit board (PCB) interconnect was experimented. The induced magnetic fields of faulty and fault free interconnects were detected by a planar array-coil sensor using eddy current testing (ECT) principle [2].

"Microprocessor Software-Based Self-Testing" This article discusses the potential role of software-based self-testing in the microprocessor test and validation process. The key idea of SBST is to exploit on-chip programmable resources to run normal programs that test the processor itself. In addition, the article proposes taxonomy for different SBST methodologies according to their test program development philosophy, and summarizes research approaches based on SBST techniques for optimizing other key aspects [3].

"Improving yield and reliability of chip multiprocessors". An increasing number of hardware failures can be attributed to device reliability problems that cause partial system failure or shutdown.

In this paper propose a scheme for improving reliability of a homogeneous chip multiprocessor (CMP) that also serves to improve manufacturing yield. Our solution centres on exploiting the natural redundancy that already exists in multi-core systems by using services from other cores for functional units that are defective in a faulty core. A micro-architectural modification allows a core on a CMP to use another core as a coprocessor to service any instruction that the former cannot execute correctly. This service is accessed to improve yield and reliability, but at the cost of some loss of performance [4].

"A built-in self-testing method for embedded multiport memory arrays". An SoC circuit generally contains processors, memories, and peripheral interface devices on a single chip. This induces various testing problems due to the inaccessibility of components, especially in memory modules, all transistors are exposed to the possibility of being attacked. An important part of an SoC is memory arrays which are used in the form of small arrays or buffers (embedded memory arrays) between subsystems with different data consumption rates. Further, we do not like to throw the chip away if there exist a limited number of faulty memory cells. Hence, the need of testing memories on a chip is critical so that further repair can be performed.

In this paper, we develop a powerful test architecture for two-port memories using the serial interfacing technique. Based on the serial testing mechanism, we propose new march algorithms which can prove effective to reduce hardware cost considerably for a chip with many two-port memories [5].

“Embedded Software-Based Self-Test for programmable Core-Based Designs”. In this paper they perform on-chip test generation, measurement, response analysis and even diagnosis . here many of the current problems with external tester-based and hardware BIST techniques for SoCs and they derive the tests that can be delivered by processor instructions [6].

VI. CONCLISION AND FUTURE SCOPE

Diagnostics software helps us to detect the faults in the NIC and it checks the functionalities of hardware components present in the network interface card. The program of testing is specific to each NIC. The diagnostics menu which we provide is user friendly so that users can install the software and check for the hardware functionalities if the card is working without any failure of devices. This software reduces the complexity of the system management and it also reduces the maintenance cost and increases the reliability of the system. Diagnostic software qualifies the NIC.

In future work GUI can be improved in better way. Software can be improved to test the multiple cards at a time in a chassis. Currently adding the card, FPGA devices and making them up is taking minimum of 7 days. With the proper data input it is better to create a tool to make it automatic.

REFERENCES

[1]. “Exploring Diagnostic Capabilities of Software-based Self-Tests for Production and In-Field Applications”.T. Koal, S. Scharoba, H.T. Vierhaus , June 2015 IEEE.

[2]. “Printed circuit board interconnect fault inspection based on eddy current testing”. S. Socheatra, N. B. Zain Ali, M. H. Md Khir, August 2014 IEEE.

[3]. “Microprocessor Software-Based Self-Testing”.Mihalis Psarakis and Dimitris Gizopoulos, Ernesto Sanchez and Matteo Sonza Reorda,2010 IEEE.

[4]. “Improving yield and reliability of chip multiprocessors”. Abhisek Pan, Omer Khan, Sandip Kundu, April 2009 IEEE.

[5]. “A built-in self-testing method for embedded multiport memory arrays”. V. Narayanan, S. Ghosh, W.-B. Jone, S.R. Das, October 2005 IEEE.

[6]. “Embedded Software-Based Self-Test for Programmable Core-Based Designs”,Angela Krstic,Wei-Cheng Lai and Kwang-Ting Cheng,Li Chen,Sujit Dey,2002 IEEE.