International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 03 Issue: 12 | Dec -2019 ISSN: 2582-3930

Design and Implementation of an Advanced Algorithm for Detection of Permanent Faults of NoC Routers

Syed Mohammed Arfath Ali, Dr R Madhu
Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University, Kakinada

***Abstract*— This paper presents the concept of an on-line Transparent SOA-MATS++ Test generation technique for detecting the permanent faults (hard latent faults) which occurred in the SRAM based memory of FIFO (First In First Out) Buffers during the field operation of NoC (Network on Chip) Routers. The NoC is a packet based network communication which deals the sending and receiving the data in the form of packets. The proposed technique, transparent SOA-MATS++ test algorithm helps to prevent the accumulation of faults by the repetition of test process. A test circuit is added to the NoC router and an on-line test is performed. After adding the test circuit to the NoC router, the performance of the NoC router has been investigated in-terms of throughput, and studied by the synthesis process of the test hardware. The synthesis and simulation results of the NoC router are in form of area, power consumption, delay. In this the utilization of area is reduced from 10.4% to 8% by online test-technique using Xilinx ISE Design suite.**

***Keywords*— FIFO buffers, NoC, SoC, permanent fault, in-field test, transparent test.**

I. INTRODUCTION

From the last few years, the technology of NoC (Network on Chip) is widely increased and demanded due to its better communication designing techniques and infrastructure over the bus-based communication network for complex chip designing. This design technique overcomes the challenges which are related to signal integrity, power dissipation, and bandwidth [1]. Similarly, all other SoC (System on Chip) and NoC based SoCs are tested for the defects. The process of testing the components of NoC infrastructure involves in testing the inter router links and the routers. The NoC architecture provides the communication infrastructure for the SoC design [2]. The NoC is a SoC which having multiple processors, usually targeted for the specific embedded applications [1]. The area of NoC transport medium is filled with the routers, which are predominately occupied by the FIFO buffer and routing logic in large amount. The occurrence of the run-time faults in FIFO buffer and routing logic are high, when compared to the other components of NoC. Hence, the test for the NoC architecture design should start with the testing of buffers and then routing logics of the routers. Similarly the test should be done periodically to conform that no faults are get accumulated.

The occurrence of run-time faults are the one of the major concern during the test of CMOS-based memories. Thus, these faults occurred due to the physical effects such as low supply voltage, aging, and environmental susceptibility forms the intermittent (non-permanent) faults in nature [3]. Thus, the frequently occurrence of these intermittent faults tends to become hard or permanent [3]. Similarly, the wear-out and decrease the life-span of memory also cause for the intermittent faults to become a latent hard which are classified as permanent faults. Thus, an on-line test technique is required for the detection of these run-time faults which are intermittent in nature and later become permanent.

For this process we introduced an on-line test called as "Transparent test algorithm" for the testing of FIFO buffers and routing logic which presents within the routers of the NoC architecture design. An online Transparent SOA MATS++ test generation algorithm is introduced for targeting the in-field permanent faults occurred in SRAM - based FIFO memories. Thus, the test is performed when the FIFO memory passes within the routers of the NoC infrastructure. Similarly, another on-line test technique is introduced for the routing logic purpose which performs simultaneously with the test buffers. The introduced technique undergoes into two ways of utilization of the unused part of the header flits. The first, one is deterministic test pattern generated by Tetra-max for routing logic. These test patterns are substitute in the place of unused fields of the header flits and transferred during the normal cycle. The second, one is the Pseudo-random patterns in the synthetic data traffic is used while normal operation is performed and occurs at the routing logic are treated as test pattern [4]. However, the estimation of fault coverage is calculated for both.

A. Types of Faults

Generally, in FIFO buffers the run-time faults are considered as intermittent faults. The primary factors that causes to aging effects, are such as Electro-migration, Time-Dependent Dielectric Breakdown (TDDB), Hot Carrier Injection (HCI) and Negative Bias Temperature Instability (NBTI) [5]. TDDB is a phenomenon where the oxide meets the gate material of an MOSFET degrades over time resulting in a short circuit, which are modelled as stuck-at-faults [6]. Electro-migration reduces interconnect conductivity with passages of time and leads to open circuit, and the circuit is modelled as stuck-open-faults [6], [7]. HCI and NBTI increases the threshold voltage value of the transistors and leads to mobility decrease. Thus, the performance of the memory core layer decreases and forms write and read failures. The write failures are modelled as transition faults, while the read failures are modelled as read disturb faults [8].

II. PREVIOUS METHODS

Over the decades, DFT and BIST are the techniques used for the testing of the routers and as well as the NoC interconnect. The DFT (Design For Testability) technique is used for testing of NoC infrastructure and NoC based core [9].

Where the BIST (Built In Self-Test) technique is used for testing the routers and NoC interconnect [10], [11], [12]. Some of the traditional algorithms are developed to detect the faults. The traditional algorithms are, Checkerboard, Zero-One, GALPAT, Butterfly, etc,.

In NoC infrastructure the FIFO Buffer are large in number and spread all over the chip [12]. Due to this reason the faults are high for FIFO buffers when compared to the other components of the router. For this, two techniques are introduced. One is online and another is offline testing techniques for the test of FIFO buffers in NoC router. The offline test technique is a Distributed BIST approach used to detect manufacture faults of NoC routers [13]. The online test technique is used for detecting the faults in the FIFO buffers of the NoC routers [14]. In both, the offline test technique is failed to detect the permanent faults.

III. PROPOSED TECHNIQUES

Previously, many algorithms has been developed for testing of faults in SRAM or DRAM memories. In that the most and advantageous is March test [15], [16]. This March test contains a sequence of March elements which is performed by write/read operation in each and every cell of the memory. The operations that can be performed in the cells are; read zero (rd0), write zero (wr0), read one (rd1), write one (wr1). It is not used directly for SRAM FIFOs, because due to its address restriction [17], [18]. To overcome this difficulty we introduced a Modified Algorithmic Test Sequence ++ (MATS++) Test technique with Single Order Address [19]. The SOA-MATS++ test is used for the detection of faults in SRAM type FIFOs. Therefore, the technique SOA-MATS++ test is represented as,

$$\{ \quad (wr0); \uparrow (rd0,wr1); \downarrow (rd1,wr0); \quad (rd0) \ \}$$

Where 0 - data,

1 - complement of data, rd0 - read a cell whose value is zero (0), wr0 - write zero,

rd1 - read a cell whose value is one (1),

wr1 - write one,

↑ - increasing the memory address from 0 to n-1, ↓ - decreasing the memory address from n-1 to 0, - memory address can be increased or decreased.

The March Test elements are; M0, M1, M2, as shown below, where X – memory location,

M0: [March element (wr0)] / * Invert Phase* / for cell = 0 to n - 1 (or any other order) do write 0 to X [cell];

M1: [March element ↑ (rd0, wr1)] /* Restore Phase */ for cell = 0 to n - 1 do
read X [cell]; (Expected value = 0)
write 1 to X [cell];

M2: [March element ↓ (rd1, wr0)] /* Read Phase */ for cell = n – 1 down to 0 do
read X [cell]; (Expected value = 1)
write 0 to X [cell];

The use of SOA-MATS++ test for the FIFO Buffer involves in writing the patterns into the memory location and reading them back, due to this the memory data is erased. But, online memory test techniques requires for the restoration of the memory content after the test. For this we modified the March test into Transparent March Test [20]. So that, the test helps to restore the memory content without using the external memory after the test. Thus, we modified the SOA-MATS++ to the transparent SO-MATS++ (TSOAMATS++) test. This can be applied for the online testing of FIFO buffers and the TSOA-MATS++ test is generally represented as,

$$\{ \uparrow(ry, w\ddot{y}, r\ddot{y} wy, ry) \}$$

The proposed technique undergoes three phases of operation during the test. The first one is invert phase where the first two operations form a read write pair (ry, wy) and undergoes invert phase. In this phase the initial data of FIFO buffer LUT (location under test) is read and the complement for this is write back to the same location. After invert phase it undergoes restore phase in this it involves the operation ($r\ddot{y}$, wy) where the data of LUT are read and re-inverted. Here the data is get backed to the original state by flipping the data twice. The last read phase involves in reading the data of LUT (ry) without any write operation.

A. TSOA-MATS++ TEST ALGORITHM

This algorithm describes step by step procedure and undergoes into three phases for each and every location of the FIFO memory. The target location of the memory for the test is represented by the loop index „p" and which varies from 0 to N-1, where N - the number of locations in the memory. For each location the test is performed into three phases during the iterations of loop index „q" [21]. The step by step procedureof the algorithm is shown in the following Fig.1.

Algorithm: Transparent SOA-MATS++ Test Algorithm

```
Require: N = number of rows of the FIFO memory
1. p ← 0;          /* memory address pointer */
2. while (p = N-1) do
3.       q ← 0;    /* test cycle counter*/
4.       While ( q = 2) do
5.       temp ← read (p);
6.                If (q = 0) then
7.                   Original ← temp;
8.                   write ( p, !temp);
9.                else
10.                  If ( q = 1) then
11.                   result ← compare (temp, original);
12.                   Write ( p, !temp );
13.                  end if
14.               else
15.                  result ← compare ( temp, original);
16.               end if
17.               q = q+1;
18.      end while
19.      p = p+1;
20. end while
```

Fig.1. Algorithmic steps

The first iteration, the address of q (q=2) performs the invert phase where the data is inverted form FIFO location. The invert phase undergoes reading the data from LUT into a temporary variable (temp) and getting back to the original variable. Then the data in temporary variable is inverted and written back to the LUT, and resembles inversion of original data.

In the second iteration, the restore phase operation is performed for another address of q (q=1). The data of LUT is reread into temp variable and compared with original data. The comparison result should be in all 1"s pattern, if any bit is represented with "0" in the result it indicates that the fault is occurred at particular bit. After that, the inverted data of temp variable is written back to the LUT. Thus, the data of LUT if restored after the second iteration which is inverted after the first iteration.

In third iteration, the index „q" performs only read operation of LUT, where the data of LUT is read into temp and compared it with original data. In this iteration the result obtained is in all 0"s pattern indicates that there is no faults, if there is any bit change (instead of 0), it means fault is occurred at that particular bit position. At last the read operation detects the fault which is not detected in previous two test runs. At the end of the third iteration of y the loop index „p" is incremented by „1" and starts the test for next location.

B. DETECTION OF FAULTS

The faults like stuck-at-fault, stuck-open fault, transition fault, and read disturb fault are occurred during the field operation of FIFO memories. Let us assume that the size of data is 16-bit, and the data is 1100 1100 1100 1100 which is present in Location Under Test (LUT).

The data in the memory is tested by the proposed Algorithm. Let us consider the data is "110 1100 1100 1100" (16-bit) is stored in LUT. The test cycles start with invert phase(i.e, form where the memory address „j") during this phase the location address of data is read into temp variable and it is stored and come back to the original location, where the data is complemented with 1"s complemented and stored in LUT. Thus, at the end of cycle the data present in both original data and temp data are same (1100 1100 1100 1100), where the LUT contains the complemented data (0011 0011 0011 0011). Assume that stuck-at-1 fault is occurred at 8^{th} bit in the complemented data. Thus, instead of storing 0011 0011 0011 0011 it stores 0011 0011 1011 0011, as a result the stuck-at-fault at position 8 from LSB.

In second iteration of „y", the LUT is readdressed and the data is read into the temp variable is 0011 0011 1011 0011. After that the comparison is done between temp and original by bitwise „XOR" operation. The obtained result should be in 1"s pattern, if any bit changed instead of 1 (i.e, 0) is occurred at any position in the data, it means there is a fault at that bit. The fault bit position in the data is shown in above Fig.2. The faults occurred at different positions for different data.

The operation of the test algorithm is shown in the following Fig.1. In this figure the data is read from input and performs the operation for each and every memory location.

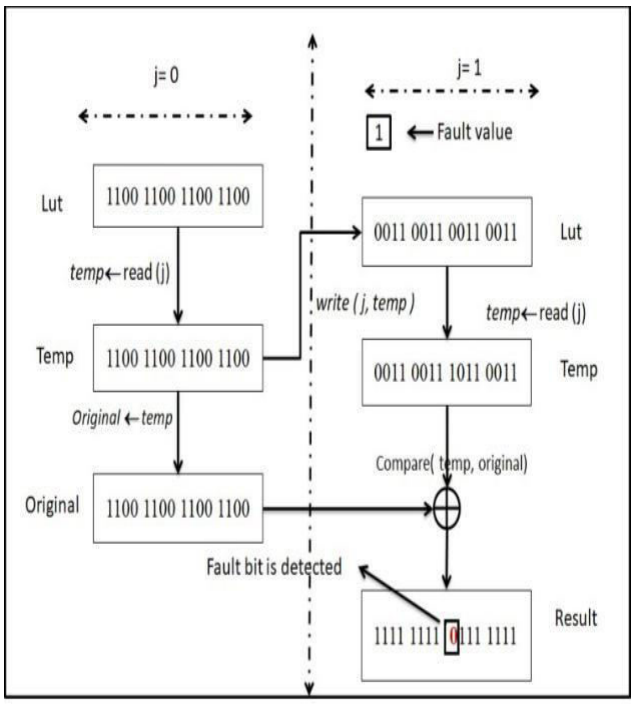

Fig. 2. Fault detection of the transparent SOA-MATS during invert phase and restore phase operation

The above Fig.1 represents how the data is read from a memory location and write into a temporary variable and compare with original data by complementing of temporary data. After that performing the XOR operation between original and complement data, and fault is detected.

IV. HARDWARE ARCHITECTURE

As, previously discussed that the NoC is a mesh type infrastructure. It is a packet based communication process, which sends and receives the data in the form of packets. Here the data packets are classified into flow control units (flits) and transmitted through pipe line architecture [1]. In this the flit movement is assumed for buffering only at the input channels of the routers. The proposed online test is performed only for a data traffic movement from one core to another core to the input channel of a FIFO buffers. The buffers are operated in two modes one is „Normal mode" and another is „Test mode". These two modes of operation of a FIFO buffer are synchronized with two different types of clocks. The clock used for test mode is „test_clk" and the used for normal mode is „router_clk".

A. Architecture

In NoC router each channel input of a FIFO buffer consists of a SRAM based FIFO memory with certain depth. During, the normal mode operation the data flits occur through a data_in line of the buffer which are subsequently stored in the different locations of the FIFO memory. After the requesting of neighbour router the stored flits of TSOA-MATS++ test on the FIFO buffer a "Test-Circuit" is added to the existing hardware with few multiplexers and logic gates as shown in the following Fig. 3(a) the read operation and write operation of the FIFO buffer are controlled by the read enable (rd_en) and write enable (wr_en) lines.

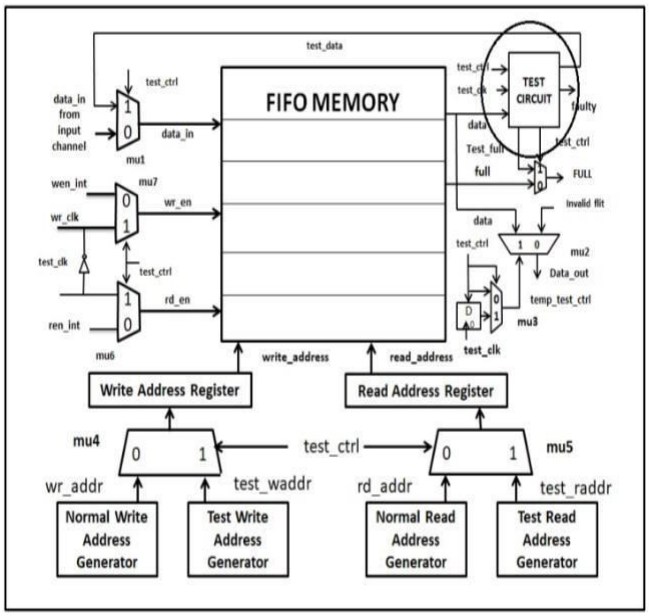

Fig.3(a). Hardware Diagram of the test process for the FIFO buffers.

The multiplexer „mu6" and multiplexer „mu7" which selects the read enable (rd_en) and write enable (wr_en) during the operations of normal mode and test mode. During the operation of normal mode the signal test_ctrl is low, the internal write enable (wen_int) and internal read enable (ren_int) are synchronized with the router clock, which provide the write enable and read enable lines. Similarly, during the test mode operation the write enable and read enable are synchronized with test clock.

The internal architecture of the test circuit is shown in the following Fig.3.(b). As discussed earlier, the read and write operations during the test are performed alternate edges of a test clock, where the read operation is synchronized with positive edges and where write operation is synchronized with negative edges of a test clock.

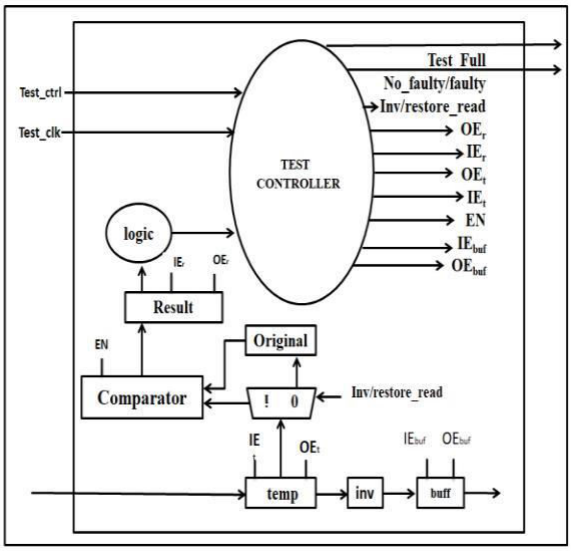

Fig.3.(b). Internal operation of Test Circuit

In test mode, test control (Test_ctrl) is high the test read address and write address are generated by the test address generators. Multiplexers mu4 and mu5 are used to select between the normal addresses and test addresses.

V. RESULTS

A . Summary Reports

The RTL schematic of the FIFO buffer of NoC router is shown in the following Fig.4. Here the schematic diagram represents the 16-bit input and output of NoC FIFO buffer with clock, merror, reset signal.

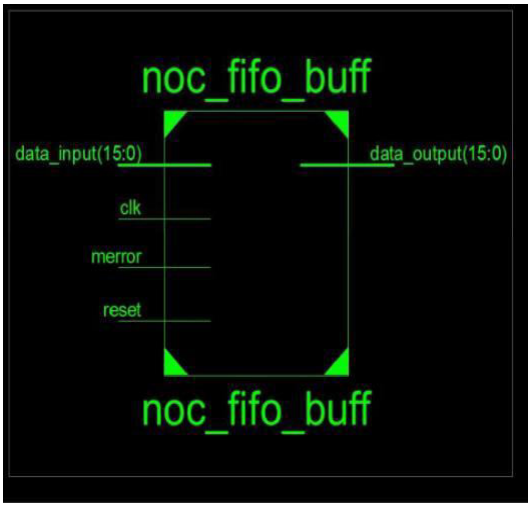

Fig. 4. RTL Schematic of FIFO Buffer

The internal architecture of rtl schematic of the FIFO buffer, is shown in the following Fig. 5. It illustrates the connections, and data flow between the registers, multiplexers, test circuit, and fifo memory.

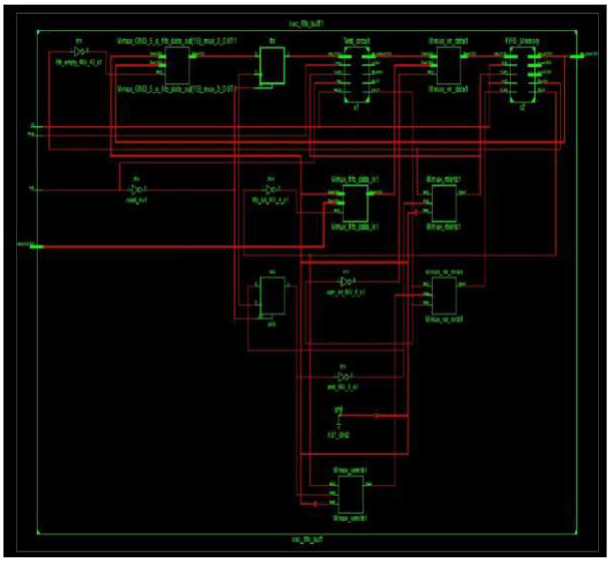

Fig. 5. Internal RTL schematic of FIFO buffer

The summary report for the previously used methods like MARCH C, MARCH X, SOA-MATS++, and the comparison between them for the FIFO Depth is 6, and data width is 16 bit

as shown in the table 1. The table illustrates with the no. of registers used, LUT‟s, power consumption, delays etc.

Table.1. Comparison between previous methods

	MARCH C	MARCH X	SOA-MATS++
Number of Slice Registers	403	337	337
Number of Slice LUT	260	191	179
Number of IOB	69	69	57
Total Power (mW)	81	81	81
Dynamic Power (mW)	1	1	0
Delay(ns)	6.506	5.824	5.824

The proposed TSOA-MATS++ test algorithm is applied to the FIFO buffer of NoC router is synthesised, and the synthesis results like power, time delay, registers used, slices used, flip flops used etc., are shown in following Table.2. It illustrates the comparison between the FIFO depths 6 and 32.

Table.2. Summary report for FIFO 6 and 32

Transparent SOA-MATS++ Test		
	FIFO(depth = 6)	FIFO(depth = 32)
No. of Slices registers	316	180
No. of Slices LUT	264	118
Slices	107	37
Supply Power(mW)	46	42
Dynamic power (mW)	0	0
Delay(ns)	3.999	2.338

Power report:

The power consumption report of the test is shown in the following table.3. The below table explains about the power consumption in the form of dynamic and quiescent (static) power of the FIFO buffer during the test. Here the dynamic power is zero (0) because, it utilizes the power only when it is on-chip implementation.

Table.3. Utilization of power

	Total	Dynamic Power	Static Power
Supply Power (mW)	42	0	42

Area report:

The utilization of the number of bonded IOB‟s are considered as the area utilization. In previous methods the utilization of bonded IOB‟s are 10.47% and considered as area utilization. In this algorithm the utilization of bonded IOB‟s are 8% as shown in the below table 3. Thus, the area is reduced by 2% by using this algorithm.

Table.4. Utilization of Area

Slice logic utilization	used	Available	utilization
Number of bonded IOB‟s	24	285	8%

Timing report:

The total timing report is shown in the following table 5. In this table it gives detail information of the timing constraints like fanout, gate delay, net dealy and considered as overall delay in nano seconds (ns).

Table.5. Timing constraints of the FIFO buffer

Cell: in->out	fanout	Gate delay	Net delay	Logic name
IBUF: I->0	1	0.318	0.574	Merror_IBUF
INV: I->0	1	0.245	0.389	c1/result_INV_0 (no_fault_OBUF)
OBUF: I->0		0.812		No_fault_OBUF (no_fault)
Total		1.375	0.963	

Total delay is 2.338 ns (1.375 ns logic (59%), 0.963 ns route (41%)).

The overall summary report of the TSOA-MATS++ test Algorithm for the FIFO buffer of a NoC router is as shown in following Fig.6.

Device Utilization Summary [-]

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	180	126,800	1%	
Number used as Flip Flops	180			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	118	63,400	1%	
Number used as logic	118	63,400	1%	
Number using O6 output only	1			
Number using O5 output only	0			
Number using O5 and O6	0			
Number used as ROM	0			
Number used as Memory	0	19,000	0%	
Number used exclusively as route-thrus	0			
Number of occupied Slices	37	15,850	1%	
Number of LUT Flip Flop pairs used	1			
Number with an unused Flip Flop	0	1	0%	
Number with an unused LUT	0	1	0%	
Number of fully used LUT-FF pairs	1	1	100%	
Number of unique control sets	1			
Number of slice register sites lost to control set restrictions	6	126,800	1%	
Number of bonded IOBs	24	285	8%	
Number of RAMB36E1/FIFO36E1s	0	135	0%	

Fig. 6. Summary report of FIFO Buffer of NoC router

In this report it illustrates about the usage of the each and every reister, slice, LUT, memory, timing constraints, delays, fanout etc,.

B. Simulation report

The FIFO buffer of NoC is simulated for obtaining the output in the form waveform signals. Here the waveforms represents the output for the 16-bit and detects for the faults. If there is any fault in the data the no_fault signal is represented with high(i.e, 1), otherwise it is low(i.e, 0). The timing waveforms are shown in the below Fig.7.

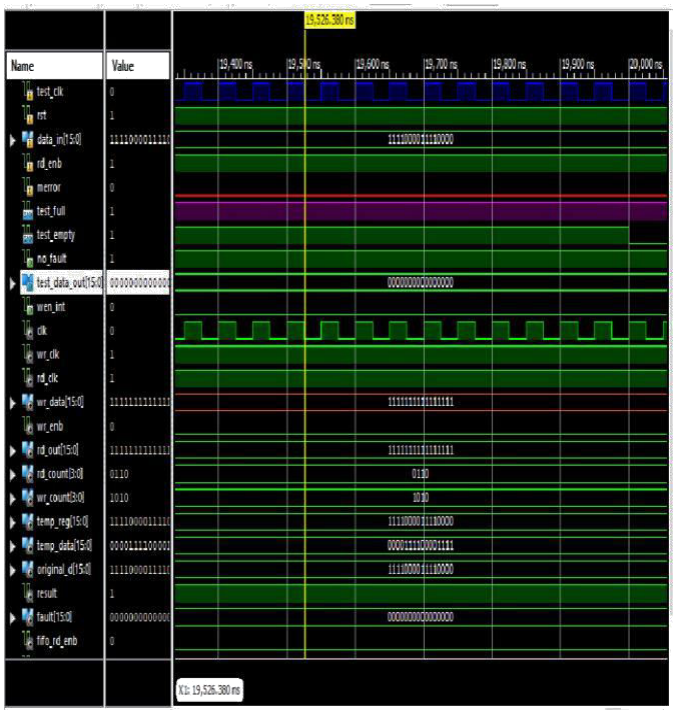

Fig. 7. Timing waveforms of the FIFO Buffer

VI. CONCLUSION

In this, the introduced Transparent SOA-MATS++ test generation algorithm which is used for detecting the run-time faults which are temporary at initial later it became permanent in SRAM based FIFO memories. The FFIO buffer memory present within the router of NoC are tested through on-line and periodic test by proposed algorithm. In periodic test testing of the buffers prevents the accumulation of faults, and also test for each location of the buffer. Along with this algorithm, another test technique is proposed for the routing logic which is operated simultaneously with the TSOA-MATS++ test. It helps in the utilization of unused fields of the header flits of the incoming data packets for the test pattern. The faults are detected and modified automatically and the exact output data is obtained without any faults. This algorithm overcomes the difficulties occur in previous methods and helps to reduction in area (optimized) [22], less power consumption, increases the speed.

REFERENCES

[1] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proc. 38th Annu. Design Autom. Conf.*, 2001,pp. 684–689.

[2] Mohammad Ayoub Khan, Abdul Quaiyum Ansari. "n-Bit multiple read and write FIFO memory model for network-on-chip", 2011 World Congress on Information and Communication Technologies, 2011.

[3] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and F. Grandoni, "Threshold-based mechanisms to discriminate transient from intermittent faults," *IEEE Trans. Comput.*, vol. 49, no. 3, pp. 230–245, Mar. 2000.

[4] BihasGhoshal, KanchanManna, SantanuChattopadhyay, and IndranilSengupta "In-Field Test for Permanent Faults in FIFO Buffers of NoC Routers", IEEE Transactions on VLSI Systems, vol.24, No.1, Jan 2016.

[5] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: New design paradigm for the nanoscaleera," *Proc. IEEE*, vol. 98, no. 10, pp. 1718–1751, Oct. 2010.

[6] S. Borri, M. Hage-Hassan, L. Dilillo, P. Girard, S. Pravossoudovitch, and A. Virazel, "Analysis of dynamic faults in embedded-SRAMs: "Implications for memory test," *J. Electron. Test.*, vol. 21, no. 2, pp. 169–179, Apr. 2005.

[7] Bibhas Ghoshal, Chittaranjan Mandal, Indranil Sengupta. "Refresh re-use based transparent test for detection of in-field permanent faults in DRAMs", Integration, the VLSI Journal, 2017.

[8] M. Radetzki, C. Feng, X. Zhao, and A. Jantsch, "Methods for faulttolerance in networks-on-chip," *ACM Comput.Surv.*, vol. 46, no. 1, pp. 1–38, Jul. 2013, Art. ID 8.

[9] D. Xiang and Y. Zhang, "Cost-effective power-aware core testing in NoCs based on a new unicast-based multicast scheme," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 1,pp. 135–147, Jan. 2011.

[10] K. Petersen and J. Oberg, "Toward a scalable test methodology for 2D-mesh network-on-chips," in *Proc. Design, Autom., Test Eur. Conf.Exhibit.*, Apr. 2007, pp. 1–6.

[11] T. W. Tsens, C.H. Wu, Y. J Huang, J-F.Li,A. Pao, K. Chiu, and E. Chen, "A Built-in-self repairscheme for multiport RAMs", in proc. IEEE VLSI Test System (VTS), Brekeley, May 2007, pp. 355-360.

[12] S. Ram Suja, M. Deivakani. "Testing of FIFO buffer of NoC router using BIST", 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE), 2017.

[13] C. Grecu, P. Pande, B. Wang, A. Ivanov, and R. Saleh, "Methodologies and algorithms for testing switch-based NoC interconnects," in *Proc.20th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, Oct. 2005, pp. 238–246.

[14] M. R. Kakoee, V. Bertacco, and L. Benini, "A distributed and topologyagnostic approach for on-line NoCtesting," in *Proc. 5th ACM/IEEE Int.Symp. Netw. Chip*, May 2011, pp. 113–120.

[15] S. M. A. Habi, Sk. Gupta, "Generating Complete and Optimal March Test for Linked faults in Memories", VTS-2003, 21th IEEE VLSI Test Symposium, 2003, pp. 254-261.

[16] A. J. van de Goor and Y. Zorian, "Functional tests for arbitration SRAM-type FIFOs," in *Proc. 1st Asian Test Symp. (ATS)*, Nov. 1992, pp. 96–101.

[17] Ting-Ju Chen, Jin-Fu Li, and Tsu-Wei Tseng, "A New Method for March Test Algorithm Generation and its Application for Fault Detection in RAMs", in IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, June-2012.

[18] M. Mamatha, M. Muralidhar, "Memory Testing using March C-Algorithm", in IJVDCS, vol.02, issue.07, Oct. 2014, pp 512-517.

[19] A. J. Van de Goor and Y. Zorian, "Effective March Algorithm Testing Single- Order Addressed memories", Journal of Electronic Testing, Theory and Application (JETTA), vol. 5, Nov-1994.

[20] M. Nicolaidis, "Theory of transparent BIST for RAMs," *IEEE Trans. Comput.*, vol. 45, no. 10, pp. 1141–1156, Oct. 1996.

[21] Biswajit Bhowmik, Santosh Biswas, Jatindra Kumar Deka, Bhargab B. Bhattacharya, "Reliability-Aware Test Methodology for Detecting Short-Channel Faults in On-Chip Networks", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2018.

[22] N. Mohanapriya, C. Priya "A fault tolerant router with optimized an colony algorithm", International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2016.