Implementation of Concurrent Online MBIST for RFID Memories using March SS Algorithm

M. Jahnavi, P. S. Indrani, M. J. C. Prasad

***Abstract*— This paper presents the implementation of online test scheme for RFID memories based on Memory Built in Self Test (MBIST) architecture. This paper also presents the, Symmetric transparent version of March SS algorithm, implementation of Memory BIST. The comparison between the different march algorithms and the advantage of the March SS algorithm over all other is also presented. The solution was implemented using Verilog HDL and was, in turn, verified on Xilinx ISE 13.2 simulator, and synthesized.**

***Index Terms*—Memory testing, RFID memories, Transponder.**

I. INTRODUCTION

Radio frequency identification (RFID) is a generic term that is used to describe a system that transmits the identity (in the form of a unique serial number) of an object or person wirelessly, using radio waves. It's grouped under the broad category of automatic identification technologies. A signal is sent to a transponder, which wakes up and either reflects back a signal (passive system) or broadcasts a signal (active system). A radio-frequency identification system uses tags, or labels attached to the objects to be identified. Two-way radio transmitter-receivers called interrogators or readers send a signal to the tag and read its response. Read-only transponders represent the low-end, low-cost segment of the range of RFID data carriers. As soon as such transponder enters the interrogation zone of a reader, a scheme to access its identification number is deployed. The tag's unique identification number is hardwired into the transponder during chip manufacture. Therefore, the user cannot alter this serial number, or any data on the chip. Recently RFID (Radio Frequency Identification) attracts attention as an alternative to the bar code in the distribution industry, supply chain and banking sector. This is because RFID system that has advantages of contact-less type and can hold more data than the bar code. Nevertheless, RFID has disadvantages about the problem of identified data clearness, the slow progress of RFID standardization. A new RFID architecture and access scheme is proposed that allows concurrent online tests of the transponder memory. A built-in self-test (BIST) controller with appropriate March SS algorithm is carefully exploited to check for memory errors.

Manuscript published on 30 November 2013.
*****Correspondence Author(s)

P.G scholar M. Jahnavi, ECE Department, Mallareddy Engineering College, secunderabad, India.
Associate professor- P. S. Indrani, ECE Department, Mallareddy Engineering college, secunderabad, India.
Dr. M. J. C. Prasad, ECE Department, Mallareddy Engineering College, secunderabad, India.

The following of this paper is organized as follows. In Section II operation of the Transponder and organisation of its memory is given in a form of flow chart. In Section III regular access scheme for testing of the transponder and the modified scheme is and the different stages in both the schemes are discussed. The section IV presents the proposed March-SS algorithm and comparison between the already proposed March algorithms and March SS algorithm. The section V presents the introduction of the MBIST implementation. Section VI presents the simulation and synthesis results. Section VII gives conclusion.

II. OPERATION OF THE TRANSPONDER

Three different layers define the transponder protocols, which are application layer, communication layer and physical layer.

The application layer the commands from an interrogator are received by the valid transponder only when the tags are singled out. These commands generally include wrong, reading or locking the tags internal memory. At this layer the interrogator had the ability to terminate the tag's operation indefinitely by issuing a password protected command in communication layer the interrogator manages the population while embracing an anti-collision protocol [1]. To support access from several interrogators, transponders provide session flags that may be asserted or de-asserted by interrogators [2].Session flags allow interrogators to organize groups of tags and force them to enter a particular inventory round [1].

Transponder memory is organized in a division in banks according to the function of the memory portion which is given as follows:

→ Reserved memory, which includes passwords for accessing special tag functions.
→ Product Identification memory, which is a code used to identify the object containing the tag.
→ Tag Identifier memory, which is the unique identification number of the tag.
→ User memory, which is an application specific bank.

III. ACCESS TESTING SCHEME

The normal operation of interrogator relies on selection of smaller group of tags ND Random assignment of access slots.

A selection command released by the interrogator forces a tag or group of tags to set or unset their internal flags according to a comparison mask. In this way, an interrogator splits a larger group of tags in smallest sets to make accessing of tags easy. An interrogator starts a new inventory pointing towards a previously selected set of tags [2].

*Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.*

Once the Transponders match the interrogator's flags selection, it must generate an internal random Queue Position Number (QPN) which represents its assigned slot in the DFSA algorithm. The maximum QPN available for the transponders is determined by the interrogator each time an inventory starts. In order to establish a direct link between interrogator-transponder, the interrogator issues a command which is answered only by transponders which QPN is equal to zero. Meanwhile, the other transponders involved in the inventory should decrement their own QPN by one, until their turn to answer the interrogator comes [1]. The success of the anti-collision scheme relies in the effectiveness of the interrogator to select an appropriate. Maximum value for the QPN which avoids picking the same time slot by more than one transponder.

A. Selection stage

Every transponder works in one of four sessions and has separate inventoried flag for each [1]. Whether the transponder may respond or not to the interrogator or not within an inventory round is determined by these flags. A Selected flag (SL) also exists which purpose is to ensure a greater accuracy during management of large transponder populations [2].the scheme also include the test flag whose purpose is to force the transponder to testing state while being access. This test flag is sent by the interrogator. With this scheme the interrogator chooses the population of tags to be tested by asserting its Test flag with the Select command.

B. Testing stage

Figure 1. Briefly describes the testing stage and figure 2 gives the FSM of the transponder access scheme [1]. When the transponder reaches in the range of an interrogator, it reaches the Ready state. Energized transponders that are not participating in an inventory round, ready state acts as a holding state. select commands issued by the interrogator that is accepted by the transponder which is in ready state and forces transponder to set or unset session flag when the interrogator broadcasts a query command with a session flag the transition state changes from ready to the Arbitrate state.

This transition happen for the transponder matching the session flag and other stay in ready state and do not participate in inventory round every transponder going to arbitrate state chooses randomly a QPN. Let t_i be the transponder and QPN_i be the randomly chosen QPN where I is the no. of transponder in an inventory round.

The access scheme allows the interrogator to adaptively choose an adequate interval of QPN in order to consider the number of transponders available in the inventory round or the time needed to finish the memory test [1].

Consequently, by issuing commands to transponders, the interrogator forces them to pass from Arbitrate to Ready back and forward until the QPN interval is appropriate for the current inventory round[2]. QPNi's valid values are defined as: QPNi E [0, 2^Q - 1]; with Q being chosen by the interrogator for each inventory round [1].

A. Memory testing

The proposed testing approach includes a new state for testing, MemTest, which sends a signal to a BIST controller to start the test of a given memory block and keeps track of its result. To prevent unwanted behaviour, a transponder ti in the MemTest state reacts only to the QueryRep command which forces the decrement of QPNi, i.e., changes to the next time slot as given in [1].

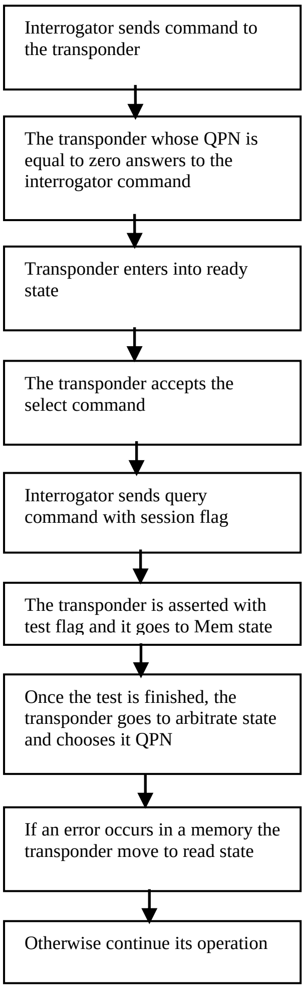

Interrogator sends command to the transponder
The transponder whose QPN is equal to zero answers to the interrogator command
Transponder enters into ready state
The transponder accepts the select command
Interrogator sends query command with session flag
The transponder is asserted with test flag and it goes to Mem state
Once the test is finished, the transponder goes to arbitrate state and chooses it QPN
If an error occurs in a memory the transponder move to read state
Otherwise continue its operation

Fig 1: Process Flow of Testing Stage

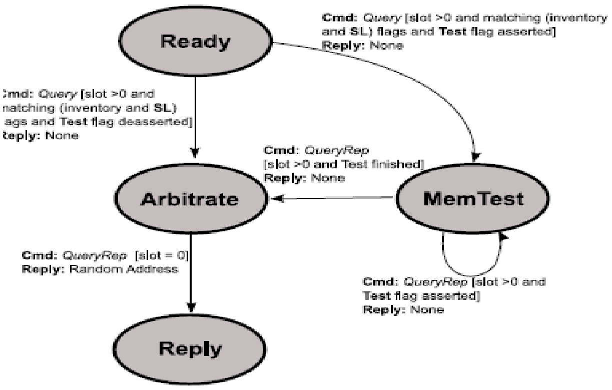

Ready
Cmd: Query [slot >0 and matching (inventory and SL) flags and Test flag asserted]
Reply: None
:md: Query [slot >0 and
natching (inventory and SL)
ags and Test flag deasserted]
Reply: None
Cmd: QueryRep
[slot >0 and Test finished]
Reply: None
Arbitrate
MemTest
Cmd: QueryRep [slot = 0]
Reply: Random Address
Cmd: QueryRep [slot >0 and
Test flag asserted]
Reply: None
Reply

Fig 2: Transponder Access Scheme

In Figure 2 there is another state called Mem State .This signals the BIST controller to start memory test of a given memory block and keeps eye on its result. A Query Rep command is issued and the transponder in the MemTest only reacts to this command which forces the decrement of $QPN_{i.}$

A memory block counter register is 32-bit reg implemented in the transponder used as a counter, during test process. The information of the memory block to test is sent to BIST through data lines.

If a Query command is received by a transponder in ready state, it should go to MemTest state and shall compute its QPN.QPN value is randomly chosen must be selected in such a way that the whole memory test is done .The QPN value so chosen is increased by a fixed offset equal to the number of blocks to test. Firstly a memory counter is load with the number of first memory blocks. Once the test is finished the transponder enters into Arbitrate state and continues to access its information .If error detection information is to be informed to an interrogator, the transponder transit to reply state along with sending a temporary random identifier accompanying an error code. This error code consists of nature and place of error.

IV. MARCH ALGORITHM

March algorithms are used for the testing for faults in semiconductor elements. A March test consists of a March elements sequence, comprising of read/write operation that have to be performed to every cell of the memory. In this paper the testing of RFID memories is done by using March SS-algorithm, which is one of the March algorithms. March algorithm detects faults such as Faults (SAF), Address Faults (AF) and some Coupling Faults (CF) and many more. The table 1 shows the March algorithm brief description.

Table I: March-SS algorithm

A	B	C	D

M0	↑(w0);	↑(ra,ra,ra);
		↑(rac,rac,rac);
		↓(ra,ra,ra);
		↓(rac,rac,rac); ↓ (ra);
			↑((ra)c);

M1	↑(r0,r0,w0,r0 ,w1);	↑(ra,ra,wa,ra,wac);	↑(ra,ra,wa,ra,wac);
M2	↑(r1,r1,w1,r1 ,w0);	↑(rac,rac,wac,rac,wa);	↑(rac,rac,wac,rac,wa);
M3	↓(r0,r0,w0,r0 ,w1);	↓(ra,ra,wa,ra,wac);	↓(ra,ra,wa,ra,wac);
M4	↓(r1,r1,w1,r1 ,w0);	↓(rac,rac,wac,rac,wa);	↓(rac,rac,wac,rac,wa);
M5	↓ (r0);	↓ (ra);	↕(ra);

A) shows March elements denotation M0 to M5;
B) shows original march SS algorithm;
C) shows transparent version
D) shows symmetric transparent version which is used in the project

General March algorithm consists mainly of four operations viz., w0 denotes write zero ; r0 denotes read zero. w1 denotes write one and r1 denotes read one. During modifying the general algorithm to symmetric transparent test if you have 0 then replace it with a; If you have 1 then replace it with ac ; where ‘c’ means complement

The table II shows the comparison between different March test available and their fault coverage. By close observation we can say that March-SS algorithm gives more fault coverage [3].

Table II: Comparison of March-test available

Algorithm	Test Length	Fault Coverage
MATS +	5n	SF, RDF, IRF
MARCH C-	10n	SF, TF, RDF, IRF, CFst, CFds, CFtr, CFrd, CFir
MARCH B	17n	SF, TF, RDF, IRF, CFds
MARCH SS	22n	SF, TF, WDF, RDF,DRDF, IRF, CFst, CFds, CFtr, CFwd, CFrd, CFdrd, CFir

1. State Fault (SF)
2. Transition Fault (TF)
3. Write Disturb Faults (WDF)
4. Read Destructive Fault (RDF)
5. Deceptive Read Destructive Fault (DRDF)
6. Incorrect Read Fault (IRF)
7. State coupling fault (CFst)
8. Disturb coupling fault (CFds)
9. Transition coupling fault (CFtr)
10. Write Destructive coupling fault (CFwd)
11. Read Destructive coupling fault (CFrd)
12. Deceptive Read Destructive coupling fault (CFdrd)
13. Incorrect Read coupling fault (CFir)

A. Symmetric Transparent Test

Regular march tests produce the erases the contents in the ROM. In order to prevent data loss a transparent approach is introduced. The transparent method avoids traditional comparison and, instead, uses a signature analysis mechanism based on a feedback shift register [2]. March tests can be easily extended to transparent versions by replacing values 0 and 1, in the read and write operations, by (r_a^c, r_a, w_a^c, w_a) respectively, where ‘$_a$’ refers to original content and $_a^c$ to its complement.

V. MEMORY BIST

A typical embedded memory BIST (MBIST) approach comprises an MBIST, an MBIST controller and interconnects between them. The MBIST further includes an address generator to provide complete memory address sequences (i.e., for n address lines all the 2n locations are visited in a complete sequence); a data generator to produce data patterns when testing word-oriented memories. A comparator to check the memory output against the expected correct data and a finite state machine (FSM) to generate proper test control signals based on the commands received from the MBIST controller. The MBIST controller pre-processes the commands received from upper-level controller (either on-chip microprocessor or off-chip ATE) and then sends them to the MBIST. BIST addresses most of the challenges faced by testing embedded memories in an SOC. However, the increase in the size and number of embedded memories and the rapid development in VLSI process technologies lead to unique requirements for Embedded MBIST.

A. Implementation of memory BIST

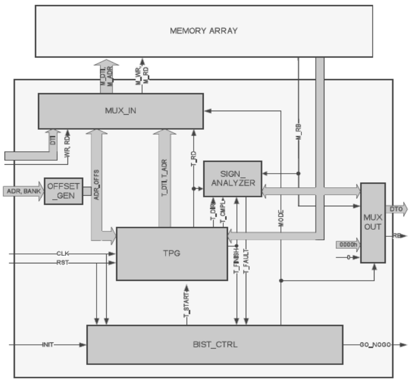


Fig 4: Memory BIST Architecture

The memory BIST architecture consists of six different modules. They are memory input multiplexer, output multiplexer, BIST controller, offset generator, signal analyser and test pattern generator. The input multiplexer selects the input signals to be sent to the memory as claimed by the BIST mode the output multiplexer provides constant values and the ready/busy (RB) signal is set to zero throughout all the test period [1]. Offset generator generates valid address sequences to the memory under test according to bank selected for the memory. The test procedure starts once the *init* signal from the transponder captured by BIST controller. The test pattern generator generates test patterns to be applied to the inputs of the memory. The role of a signature analyser is to compact the sequence of test responses coming out from the CUT into a single word. The so obtained signature is compared to the expected one. If they differ, this means that at least one erroneous response has been catch into the signature analyser during the test procedure.

VI. SIMULATION AND SYNTHESIS RESULTS

The Simulation is done by Xilinx simulator and Synthesis is done by Xilinx Synthesis tool. The Simulation results are shown in below figures. Fig 5 shows the testing of a faulty memory where a fault is induced hence there is no data output on the data signal. Fig 6 there is a testing of a fault less memory hence the final data output is given on the output signal. We can see the complete March algorithm testing in both the figures where the ROM data is being modified.

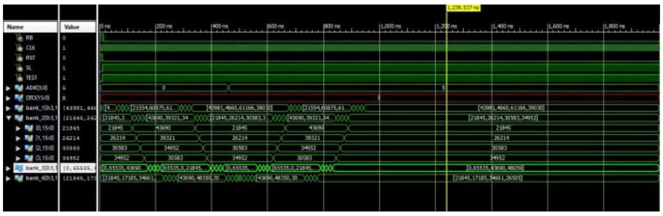


Fig 5: RFID memory with fault

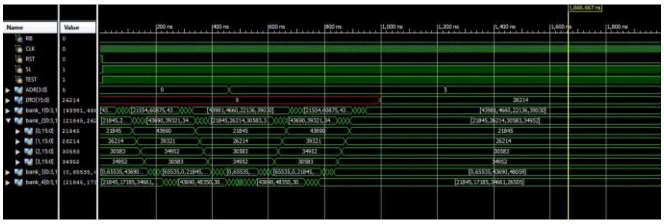
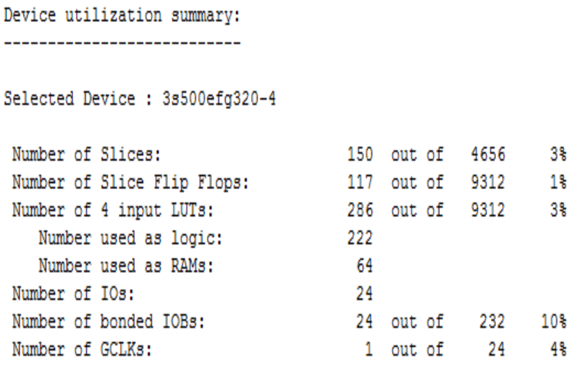


Fig 6: RFID memory without fault

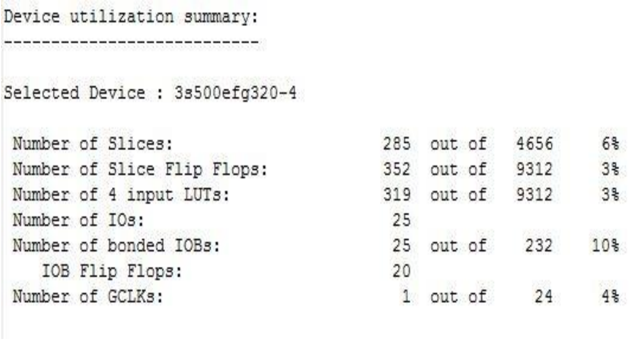


```
Device utilization summary:
---------------------------

Selected Device : 3s500efg320-4

 Number of Slices:                      150  out of   4656     3%
 Number of Slice Flip Flops:            117  out of   9312     1%
 Number of 4 input LUTs:                286  out of   9312     3%
    Number used as logic:               222
    Number used as RAMs:                 64
 Number of IOs:                          24
 Number of bonded IOBs:                  24  out of    232    10%
 Number of GCLKs:                         1  out of     24     4%
```

Fig 7: Cell usage for March C algorithm



```
Device utilization summary:
---------------------------

Selected Device : 3s500efg320-4

 Number of Slices:                 285  out of   4656     6%
 Number of Slice Flip Flops:       352  out of   9312     3%
 Number of 4 input LUTs:           319  out of   9312     3%
 Number of IOs:                     25
 Number of bonded IOBs:             25  out of    232    10%
    IOB Flip Flops:                 20
 Number of GCLKs:                    1  out of     24     4%
```

Fig 8: cell usage with March-SS algorithm

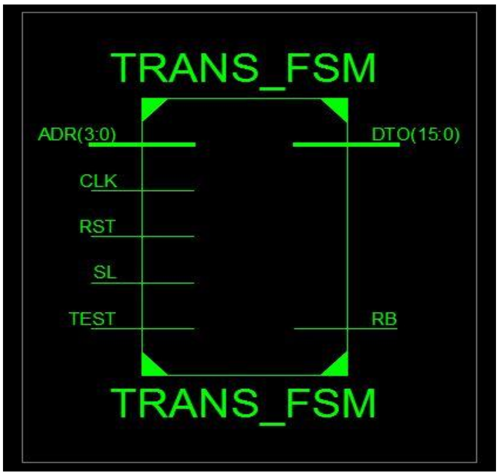


Fig 9: Top module implemented using March-ss algorithm

VII. CONCLUSION

The proposed scheme using March SS algorithm has been implemented successfully and simulated and synthesized Xilinx 13.2 version. The advantages of using the March SS algorithm than the previously proposed March C algorithm are more fault coverage which eventually increases the fault detection.

VIII. ACKNOWLEDGMENT

I M. Jahnavi would like to thank P. S. Indrani Associate professor, who guided me through out to complete my work successfully. I would like to thank my HOD (ECE Dept.) Dr. M. J. C. Prasad for providing us constant support and providing us the resources needed.

REFERENCES

1. Erwing R. Sanchez and Maurizio Rebaudengo “*A Novel Access Scheme for Online Test in RFID Memories”*.

2. Sunil kumar chalamacherlab, Dr. K. Padmapriya, Department of ECE, JNTUACEA, Anantapur, 515002, India *“Implementation of Concurrent Online MBIST for RFID Memories”* Sunil Kumar Chalamacherlab. ,Int.J.Computer Technology & Applications,Vol 3 (4), 1587-1592.

3. Said Hamdioui1;2 Ad J. van de Goor2 Mike Rodger *“March SS: A Test for All Static Simple RAM Faults”* Proceedings of the 2002 IEEE International Workshop on Memory Technology, Design and Testing (MTDT 2002)

4. MironAbramovici, Melvin a. Breuer, D Arthur Friedman, Digital Systems Testing and Testable Design, **ISBN0-7803-1062-4.**

AUTHOR PROFILE

M. Jahnavi is presently pursuing final semester M.Tech in Digital Systems & Computer Electronics at Mallareddy Engineering College, Secunderabad. She received degree B.E in Electronics and communication From Mallareddy College of Engineering and Technology. Her areas of interest are Digital system designs, VLSI.

P. S. Indrani is presently working as an Associate Professor in the department of Electronics and Communication Engineering, MREC, Secunderabad, Andhra Pradesh, India. She is having 8 years of teaching experience. Her areas of interest are VLSI, Image Processing, Embedded systems.

Dr. M. J. C. Prasad is presently working as a Head of the department of Electronics and Communication Engineering, MREC, Secunderabad, Andhra Pradesh, India. He is having 15 years of teaching experience. His areas of interest are Communication systems, Digital Systems, Image Processing. Digital signal processing, Advance DSP Systems.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)