An enhanced barrel shifter based BIST scheme for word organized RAMs (EBBSR).

M.leela vinoth krishnan

Depatment of Electronics and Communication, CEG-Anna university, Chennai, INDIA.
Krishnan7_ece@yahoo.co.in.

***Abstract*— This paper proposes a BIST scheme for word organized Rams with March C test algorithm using the barrel shifter of processor without any extra circuit , most of the processors has barrel shifter at the hardware level ,except some intel processor , here a functionally optimized barrel shifter enhanced with all the necessary function is described , however this scheme can be used in processors with barrel shifter supporting all the functions of shift and rotate as described in this paper .Considering barrel shifter as a part of the processor this BIST scheme has Zero hardware Overhead lowest among the literature, less algorithmic time and mean time to detect , no serial fault masking.**

March test algorithm is used to test word organized RAM since march test have been proven to provide high fault coverage, simple, and easy to implement using on chip resources.

Keywords*-component: Built in self test , design for testability, march –c algorithm , Ram Testing.*

1.INTRODUCTION

Random Access Memories (RAMs) tend to increase in size and to occupy much more of the chip area. The 2006 version of the International Technology Roadmap for semiconductors (ITRS) indicates that the area occupied by memory increases from 71 percent to 82 percent as we move from the 130 nm to the 45 nm technology Typically, chips contain a large number of embedded small-to- medium sized memories and very few large blocks. This series of remarks calls for an expansion of the arsenal of schemes for built-in testing of such modules.

March algorithms have been extensively used for testing RAMs since they are simple, time efficient, “built-in self-test (BIST) friendly” (that is, it can be easily implemented using on-chip resources compared to other testing schemes) and have been proven to provide high fault coverage in the case of bit-organized RAMs [1],[a].

The solution of considering the RAM as a 2D array and applying the march sequence to all of the cells of the 2D array in a cell-per-cell fashion [2], [3] has some interesting advantages, first it makes the BIST scheme independence of the RAM word size. And second it makes the BIST scheme independent of the considered coupling faults. Such a solution is therefore a good candidate scheme for small-sized to medium-sized RAMs.

Various BIST schemes for word-organized RAMs have since evolved . For example, Savir [5] proposed the utilization of two LFSRs and a comparator. The two LFSRs operate in parallel: One feeds the inputs of the RAM while the other is compared to the RAM outputs. The scheme is based on pseudorandom testing and the fault coverage depends on the number of times that the test is allowed to repeat; for example, it is claimed in [5] that, in order to cover the cell coupling faults with a 99 percent certainty, a 60 * n *M test is required.

Huang and Jone extended the work of Nadeau-Dostie et al. [3]. They proposed a bidirectional serial interface to test for fault masking during the serial scan of the test. The hardware implementation of the scheme is n 3-to-1 multiplexers and n latches.

This EBBSR based BIST scheme paper proposes that using the functionally optimized barrel shifter enhanced with all the

necessary functions a simple BIST scheme can be developed without any extra hardware overhead , Barrel shifter which would be available in almost all processor, barrel rotation Operation with carry will be a added facility for the processor which in turn will be useful in creating a BIST scheme with reduced algorithmic complexity and with least area overhead .

2 .March algorithm Overview

A march algorithm for bit-organized memories consists of L march elements, denoted by Mi, with 0 <= i < L. Each march element is comprised of march steps, one for each examined memory cell. Each march step is comprised of zero (or one) read operations denoted by r0 (r1), meaning that 0 (1) is expected to be read from the memory cell and zero (or one) write operations denoted by w0 (w1), meaning that 0 (1) is written to the RAM cell.

2.1 Steps of March-c algorithm:[six march elements]

1. Initialize RAM to zero – write zero.
2. Write 1's in ascending order – read zero and write one repeat this in each word in ascending order bit by bit
3. Write 0's in ascending order – read one and write zero repeat this in each word in ascending order bit by bit
4. Write 1's in descending order –read zero and write one repeat this in each word in descending order bit by bit
5. Write 0's in descending order –read one and write zero repeat this in each word in descending order bit by bit
6. Read zero in ascending order.

Pictorial representation of C-march algorithm :

TABLE - 1

M0	←/→	[W 0]
M1	→	[R0W1]
M2	→	[R1W0]
M3	←	[R0W1]
M4	←	[R1W0]
M5	→	[R0]

The C- algorithm (table.1) consists of six march elements, denoted by M0-M5, and has been proven to detect all cell stuck-at, address decoder stuck-at, cell-inverting, and idempotent coupling faults [1],[a] In the case of word-organized RAMs ,several RAM cells are simultaneously being written . Thus intra word coupling faults may appear. In order to cope with these faults, one solution that can be used is to treat the memories as 2Darrays and to try applying the march elements in a bit-per-bit fashion [2].This way, addressing the sequence between different addresses of the RAM is the same as in the case of bit-organized memories. Ascending order sequence is transformed into a shift-right pattern while the descending order sequence is transformed into a shift-left pattern, With in the bits of the same word. This way, all intra word faults are detected as long as the march algorithm detects the corresponding faults in the bit organized memories (inter word faults). Each march element is substituted by n operations for each bit in the word.

3. Algorithm to produce march C pattern:

March C element 1: Initialize 0 - write 0 to all location

March C element 2: Right shift 1
EX : 100--- 110 ---111

This operation can be produced by arithmetic shift right except at the start of 1st bit of a word ,During the first bit write RRC with carry equal to 1 can be done or load 100 can be adopted, When writing to the1$^{st\ bit\ of}$ next word Acc should be negated so that it will be all zero and the same process as before is followed for all word .

C-ALGORITHM FOR N-WORD M BIT RAM: TABLE - 2

March element	For a M-word * N bit RAM
[W 0] ← 0R →	For (P= M-1 ; p>=0; p--) W(0* N)
[R0W1] →	for (P= M-1 ; p>=0; p--) { for (K=0;K< =N-1 ; K++) R(1*k 0*(n-k)) W(1*(k+1) 0*(n-k-1)) }
[R1W0] →	for (P= M-1 ; p>=0; p--) { for(K=0;K< =N-1 ; K++) R(0*k 1*(n-k)) W(0*(k+1) 1*(n-k-1)) }
[R0W1] ←	for (P= M-1 ; p>=0; p--) { for (K=0;K< =N-1 ; K++) R(0*k 1*(n-k)) W(0*(k-1) 1*(n-k+1)) }
[R1W0] ←	for (P= M-1 ; p>=0; p--) { for(K=0;K< =N-1 ; K++) R(0*k 1*(n-k)) W(0*(K-1) 1 *(n-k+1)) }
[R0] ←	for (P= M-1 ; p>=0; p--) R(O*n)

March C element 3: Right shift 0

EX: 011---001---000

This operation can be done by arithmetic right shift except at the start of 1st bit of a word ,During the first bit write RRC with carry equal to 0 can be done or load 011 can be adopted, When writing to the1st bit of next word Acc should be negated so that it will be all one and the same process as before is followed for all word.

March C element 4: Left shift 1

EX: 001---011---111

This operation can be done by left shift carry (lsc)or rotate left thru carry with carry equal to1 ,When writing to the 1st bit of next word Acc should be negated so that it will become all zero and the same process as before is followed for all word. In talbe.4 *indicates that if lsc is used at first bit then for all bit lsc can be used else we cant. refer the function lsc in table.3

March C element 5: Left shift 0

EX : 110 ---- 100 ---- 000

This operation can be done by logical left shift with continues from previous When writing to the 1st bit of next word negate the Acc so thet it will be all ones and the same process as before is followed for all word.

March C element 6: Read all 0 .

3.1 Enhanced barrel shifter to generate march –c pattern: TABLE - 3

s3	**S2**	**S1**	**S0**	**D3**	**D2**	**D1**	**D0**	**function**
Cntrol signals				Data bits				
0	0	0	0	C	A3	A2	A1	rsc
0	0	0	1	A3	A3	A2	A1	Ars1
0	0	1	0	A3	A3	A3	A2	Ars2
0	0	1	1	A3	A3	A3	A3	Ars3
0	1	0	0	A2	A1	A0	C	lsc
0	1	0	1	A2	A1	A0	0	Lls1
0	1	1	0	A1	A0	0	0	Lls2
0	1	1	1	A0	0	0	0	Lls3
1	0	0	0	C	A3	A2	A1	rrc
1	0	0	1	A0	A3	A2	A1	rr1
1	0	1	0	A1	A0	A3	A2	rr2
1	0	1	1	A2	A1	A0	A3	rr3
1	1	0	0	A2	A1	A0	C	rlc
1	1	0	1	A2	A1	A0	A3	rl1
1	1	1	0	A1	A0	A3	A2	rl2
1	1	1	1	A0	A3	A2	A1	rl3

rs -right shift , ls-left shift , rr- rotate right , rl-rotate left , rr- rotate right, c- carry, A- arithmetic , L – Logic .

For a N shift barrel shifter logbase2 N number of control signal is needed with one control signal for direction(right/left) and one control signal to specify shift or rotate and of course we have to include with carry or without carry operations also. Generally most of the processor include barrel shifter and supports most of function , here functionally optimized Barrel shifter enhanced with all the necessary functions were used , the barrel shifter function state table table.3 shows the controls signals, data bits , function for a 4 bit data. In this table.3 logical right shift lrs N (N specifies number of shift)is not included since we can get lrs from Ars Arithmetic right shift Ars N easily by neglecting /making zero operation will always be zero , if accumulator is not zero it indicates a error at that word of Ram currently read and the location of 1 in the Accumulator value indicates the error bit of the word. During Accumulator initialize process, it is loaded with value of last written Word of Ram , in this scheme no extra circuit is used for BIST purpose , where as in Ionnis[4] 3 universal gates and 1 N bit OR gate is used.

4.1 ALU operations to generate march elements using EBBSR : TABLE – 4

1st bit	2nd bit	… …	nth bit	location	Function: March C element
C=1 ;RRC/load 100	ARS	… …	ARS	1	Right shift 1 (Write 1 →)
~Acc & ”	”	”	”	2	
~Acc & ”	”	”	”	m	
C=0 ; RRC/load 011	ARS	… …	ARS	1	Right shift 0 (Write 0 →)
~Acc & ”	”	”	”	2	
” ”	”	”	”	m	
C=1; *LSC/RLC/load 001	*LSC/ C=1; RLC	… …	*LSC/ C=1;RLC	1	Left Shift 1 (← Write 1)
~Acc &C=1;*LSC/ RLC	”	”	”	2	
” ”	”	”	”	m	
LLS	LLS	… …	LLS	1	Left shift 0 (← Write 0)
~Acc & LLS	”	”	”	2	
”	”	”	”	m	

5. Comparison

The march C which is an test generation algorithm for RAM has high fault coverage than other test generation algorithm for RAM - Van de goor [1] , hence we deal here the effectiveness of the proposed scheme with respect to readcheck process the present written value is read from N bits from msb after Ars N operation, but it is not as straight forward to get Ars from Lrs as we did Lrs from Ars . hence this table provides a functionally optimized barrel shifter enhanced with all the possible operation that a barrel shifter can support.

4. The core logic of applying March C pattern and fault identification:

Using the above defined ALU operations in table.4 march C pattern is generated and applied (written) to the locations of RAM in march C order pattern, after every write of a pattern, readcheck and accumulator initialize process is done , during and its subtracted with the ALU op (Acc) which is nothing area overhead , algorithmic time for testing ,quality of test, the possibility of error masking and the mean time to detect a faulty cell value. Based on the above criteria the proposed scheme is compared with the Nadeau-dostie [2] serial interfacing for embedded ram testing , and with D-Ch,Haung [3] a parallel built in self diagnostic method for embedded memory arrays and with Ionnis voyiatzis [4]. Consider a two input mux have a hardware overhead of three gate equivalents ,From the table we see the proposed scheme as the lowest /Zero hardware overhead and testing time ,furthermore the mean time to detect an error is n/2 for the competitive schemes where n is the RAM word width (since the contents of ram need to be shifted in order for a fault in a cell to be detected and this done with n shifts in the first two scheme) but 1 in Ionnis and proposed scheme because during every readcheck process a fault in a cell is immediately detected after it is written , since in this BIST method test pattern is written in a parallel manner as different word pattern which in turn write the pattern Bit by Bit , hence there is no serial fault masking. Algorithmic time is a factor indicates number of

steps to be worked out. For March C as in table.1 contains 4 steps each containing two operations read and write , M0 only has write operations , M5 only has read operations hence totally 10 operations for the algorithm to be worked out for n*M Ram. where n is the number of bits of a word and M is the number of words. Hence algorithmic time comes to 10*n*M for the Ionnis[4] and this scheme. Which is less compared to other schemes as shown in table.5.

TABLE - 5

Test scheme	Detected intraword coupling faults	Algorithmic Time	h/w gates	Serial fault masking	Mean time to detect
Nadeau – dostie[2]	Not all (half)	20*n*M	8*n	yes	n/2
D.Ch.Haung[3]	Yes	20*n*M	11*n	no	n/2
Ionnis [4]	Yes	10*n*M	n+3	no	1
proposed	yes	10*n*M	0	no	1

6.Conclusion

We have presented an enhanced barrel shifter based BIST scheme for word organized RAMs (EBBSR). With the proposed scheme, the March algorithm is applied utilizing an enhanced barrel shifter and ALU, Compared with other schemes that have been proposed in the open literature [2],[3],[4] the proposed scheme achieves equal fault coverage with Zero hardware overhead and least testing time, Any march algorithm can be implanted because of the versatility of the processor imparted due to the enhanced barrel shifter and also in this scheme location of error at bit level can be identified.

7.References :

[1]] A.J. van de Goor, “Using March Tests to Test SRAMs,” IEEE Design and Test of computers, vol. 10, 1993.

[2] B. Nadeau-Dostie, A. Silburt, and V. Agarwal, “Serial Interfacing for Embedded-Memory Testing,” IEEE Design and Test of Computers,vol.7,no. 2, pp. 52-63, Mar.-Apr. 1990.

[3]D.-C. Huang and W.-B. Jone, “A Parallel Built-In Self-Diagnostic Method for Embedded Memory Arrays,” IEEE Trans. Computer- Aided Design of Integrated Circuits and Systems, vol. 21, no. 4, pp. 449-465, Apr. 2002.

[4]Ionnis Voyiatzis “An ALU based BIST scheme for word organized RAMs” IEEE Transactions on Computers, vol. 57, no. 5, pp. 577-590, May, 2008

[5]] J. Savir, “RAM BIST,” Proc. 17th IEEE Instrumentation and Measurement Technology Conf., vol. 1, pp. 204-211, Jan.-Mar. 2000.

Books referred:

[a]M.Abramovici, M.A.Breuer and Friedman”digital systems and testable design”,jaico publishing house 2002.

[b]M.L.Bushnell and V.D.Agarwal, “Essentials of electronic testing for digital memory and mixed signal VLSi circuits”,Kluwar academic publishers,2002.

Mr. M.L.V.Krishnan received M.E. degree in Applied electronics form CEG – Anna University, Chennai, India in 2009 and B.E Degree in Electronics and Communication Engineering from Bharath Institute of Science and technology- Anna University, Chennai India in 2006.

He is currently working as a scientist in VECC his research area includes , Analog , Digital and Mixed signal vlsi design, and application of Digtal Estimation techniques in power system measurement, regulation and control .