ISSN: 2278-0181
Vol. 2 Issue 3, March - 2013

Optimization Of Memory Built In Self Test And Repairability By Using March-SS Algorithm For SRAMS

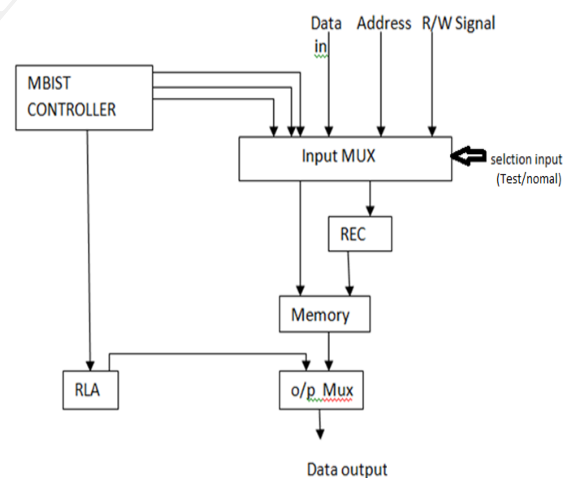
N.B.Jilani[1] K.Murali Krishna[2] and Dr.M.Sailaja[3]
[1]Student,ECE Dept,Aditya Engineering College,surampalem, E.G.Dt, AndhraPradesh
[2]Assistant professor, ECE Dept, Aditya engineering college, Surampalem , E.G.Dt, Andhrapradesh
[3]professor of ECE Dept, JNTU college of engineering, JNTUK, KAKINADA, E.G.dt, Andhrapradesh

***Abstract*-As embedded memory on-chip is increasing and memory density is growing, problems of faults is growing exponentially. For detecting and repairing certain new faults March-SS algorithm was introduced. For implementing this March-SS algorithm, a word oriented Memory Built-in self Repair Methodology is employed to repair the faulty locations indicated by the MBIST controller. This paper also presents to prevent a SRAM from executing successive multiple read operations on the same position, such that hard to detect defects can't manifest as functional faults. This can prolong the life time of the SRAM with latent hard-to detect defects. Experimental results show that the proposed reliability enhancement circuit (REC) can effectively improve the reliability of SRAMs.**

Keywords- Built-in self Test(BIST); Built-in self Repair(BISR); Memory Built in self Test(MBIST);Memory Built in self Repair(MBISR),March Element, Reliability Enhancement circuit(REC)

I.INTRODUCTION

Many SOCs consists of many diversified memory cores due to today's memory greedy applications. Therefore efficient reliability – enhancement techniques are necessary for modern random access memories (RAMs). Built-in self Repair (BISR) techniques have been shown to be a good approach for repairing embedded memories since. With the reduction transistor size and forceful design rules, memory cores are easily prone to manufacturing defects and reliability problems. Various BISR approaches for memories have been reported in [1]-[6]. A BISR circuit usually consists of a Built-in self-Test (BIST) component and Redundancy Logic array(RLA). The BIST is used to detect the targeted functional faults. The RLA allocates the redundancy (spare element) according to the detected fault patterns. To detect and repair certain new faults a March-SS algorithm was used. A new Microcoded BIST architecture here which is capable of employing this March-SS algorithm. The interface of BISR with REC is shown in fig.1



II.DETAILED ARCHITECTURE OF BISR

The DPM screening of a larger number of tests to a large number of memory chips, showed that many well known fault models developed before 1990's failed to explain the occurrence of major faults. This implied that new memory technologies involving high densities of dwindling devices lead to newer faults

IJERT

These new faults cannot be easily detected by established tests like March-C, rendering it insufficient/inadequate for today's and the future high speed memories. This stimulated the introduction of new fault models, based on defect injection and SPICE simulation. Some such newely defined fault models[2] are write disturb fault(WDF), Deceptive Read Disturb coupling fault (cfdrd). Another class of faults called Dynamic faults which require more than one operation to be performed sequentially in time in order to be sensitized have also been defined

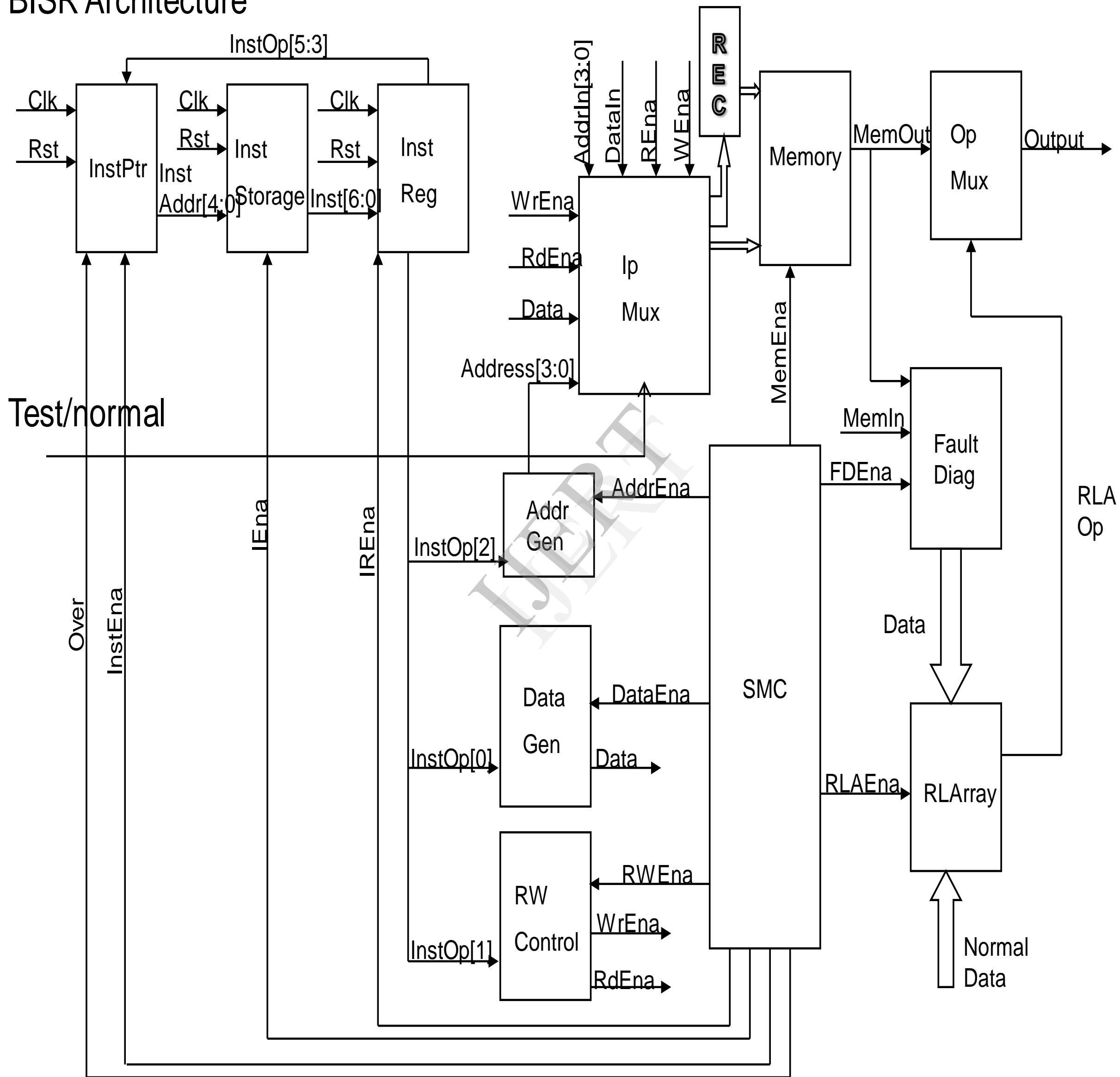


Fig2. Microcode MBIST controller Architecture and its interface with Reliability Enhancement circuit and fault diagnosis module through input MUX

More appropriate test algorithms like March-SS and March-RAW have been developed to deal with the new fault models like Deceptive Read destructive fault (DRDF) and Write disturb Fault (WDF) etc; March RAW covers some of the dynamic faults. Thus architecture which have been developed to put into practice earlier tests like March-C may not be able to easily implement these newer tests algorithms. The reason is that most of the newly developed algorithms have up to six or seven (or even more) number of test operations per test element. For example test elements M1 through M4 of March-SS algorithm have five test operations per element. This is in distinction with some of the algorithms developed earlier like March B, MATS+, March C which only had up to two operations per march element. Thus some of the recently developed architectures that had been specifically designed to implement these older algorithms can only implement up to two March operations per march element, exposé them incapable of easily implementing the new test algorithms[1][3]. The Generator, R/W control and data control. ***Pulse Generator-*** generates a 'start pulse' at positive edge of the 'start' signal marking the start of test cycle. ***Instruction pointer*** points to the next microword, that is the next operation to be applied to the memory under test (MUT). Depending on the test algorithm, it is able to i) point at the same address ,ii)point to the next address, or iii) jump back to a previous address. ***Instruction Register*** holds the microword (containing the test operation to be applied) pointed at by the instruction pointer. The various relevant bits of microword are sent to other blocks from IR. ***Address Generator*** points to the next memory address in MUT according to the test pattern sequence. It can address the memory in forwards as well as backward direction. ***R/W control*** generates read or write control signal for MUT, depending on the significant microword bits. ***Data control*** generates data to be written to or expected to be read out from the memory location being read out from the memory location being pointed at by the address generator. The Address proposed architecture has the ability to execute algorithms with unlimited number of operations per march element. Thus almost all of the recently developed March algorithms can be successfully implemented and applied using this architecture. This has been illustrated in the present work by implementing March SS algorithm. The same hardware has also been used to impliment other new march algorithms. This requires just changing the instruction unit, or the instruction codes and sequence inside the instruction storage unit. The instruction storage unit is used to store prearranged test pattern.

A)Methodology

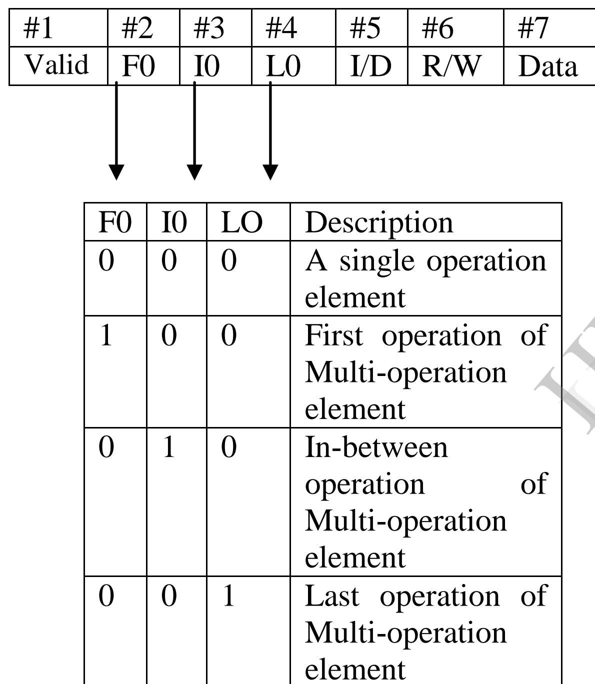
The block diagram of the BIST controller architecture together with fault diagnosis interface and Reliability Enhancement circuit(REC) through input MUX shown in Fig.2 . The BIST control circuitry consists of clock generator, pulse generator, instruction pointer, Microcode instruction storage unit, Instruction Register. The Test collar circuitry consists of Address Generator, R/W control and Data control together consist of the Memory *Test collar*.

Input Multiplexer directs the input to memory by switching between test algorithm input and input given externally during normal mode. The control signal for this multiplexer is also given externally by the user. If it indicates test mode then internally generated test data by BIST controller is given to the memory as input from the test collar [1]. In case of normal mode the memory responds to the external address, data and read/write signals. ***Fault Diagnosis*** module works during the test mode to give the fault waveform which consists of positive pulses whenever the value the value being read out of the memory does not match the expected value as given by the test collar. In addition, it also gives the diagnostic information like the faulty memory location address and the expected/correct data value. This diagnostic information is used for programming the repair redundancy array as explained in the following section.

IJERT

B) Microcode instruction specification

The microcode is a binary code that consists of a fixed number of bits, each bit specifying a particular data or operation value. As there is no standard in developing a microcode MBIST instruction [7], the microcode instruction fields can be structured by the designer depending on the test pattern algorithm to be used. The microcode instruction developed in this work is coded to denote one operation in a single microword. Thus a five operation March element is made up by the five micro code words. The format of 7-bit microcode MBIST instruction word is as shown in fig.3



#1	#2	#3	#4	#5	#6	#7
Valid	F0	I0	L0	I/D	R/W	Data

F0	I0	LO	Description
0	0	0	A single operation element
1	0	0	First operation of Multi-operation element
0	1	0	In-between operation of Multi-operation element
0	0	1	Last operation of Multi-operation element

Fig.3Format of microcode Instruction word

It's various fields are explained as follows: Bit#1(=1) indicates a valid microcode instruction, otherwise , it indicates the end of test for BIST controller. Bits #2,#3 and #4 are used to specify first operation , in-between operation and last operation of a multi-operation March element, interpreted as shown in fig.3. Bit#5(=1) notifies that the memory under test(MUT) is to be addressed in decreasing order; else it is accessed in increasing order. Bit#6(=1) indicates that the test pattern data is to be written into the MUT; else, it is retrieved from the memory under test. Bit#7(=1) signifies that a byte of 1s is to be generated (written to MUT or expected to be read out from the MUT); else byte containing all zeroes are generated. The instruction word is so designed so that it can accommodate any existing or future March algorithm. The contents of Instruction storage unit for March-SS algorithm are shown in table.1.

IJERT

	#1 Valid	#2 F0	#3 I0	#4 LO	#5 I/D (0/1)	#6 R/W (0/1)	#7 Data (0/1)
M0:χ W0	1	0	0	0	0	1	0
M1:↑{R0	1	1	0	0	0	0	0
R0	1	0	1	0	0	0	0
W0	1	0	1	0	0	1	0
R0	1	0	1	0	0	0	0
W1}	1	0	0	1	0	1	1
M2: ↑{R1	1	1	0	0	0	0	1
R1	1	0	1	0	0	0	1
W1	1	0	1	0	0	1	1
R1	1	0	1	0	0	0	1
W0}	1	0	0	1	0	1	0
M3:↓{R0	1	1	0	0	1	0	0
R0	1	0	1	0	1	0	0
W0	1	0	1	0	1	1	0
R0	1	0	1	0	1	0	0
W1}	1	0	0	1	1	1	1
M4:↓{R1	1	1	0	0	1	0	1
R1	1	0	1	0	1	0	1
W1	1	0	1	0	1	1	1
R1	1	0	1	0	1	0	1
W0}	1	0	0	1	1	1	0
M5:χ{R0	1	0	0	0	1	0	0
	0	χ	χ	χ	χ	χ	χ

Tab.1 Contents of instruction storage unit for March SS Algorithm

The first March element M0 is a single operation element, which writes zeros to all memory cells in

any order, whereas, the second March element M1 is a multi-operation element, which consists of five operations: i) R0, ii)R0, iii)W0, iv)R0 and v)W1. MUT is addressed in increasing order as each of these five operations is performed on each memory location before moving on to the next location.

III)WORD REDUNDANCY MBISR

The MBISR logic used here can function in two modes.

A)Mode1: Test & Repair Mode

In this mode the input multiplexer connects test collar input for memory under test as generated by the BIST controller circuitry. As faulty memory locations are detected by the fault diagnosis module of BIST controller, the redundancy array is programmed. A redundancy word is shown in fig.4.

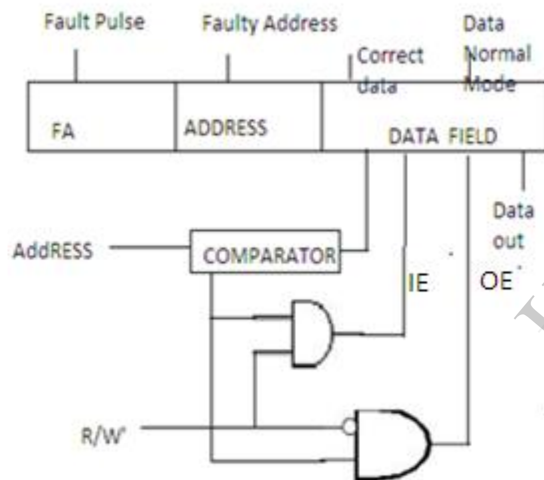


Fig.4 Format of Redundancy word Line

The fault pulse acts as an start signal for programming the array. The redundancy word is divided into three fields. The FA(Fault asserted) indicates that a fault has been detected, The address field of a word contains the faulty address, where as the data field is programmed to contain the correct data which is compared with the memory output. The IE and OE signals respectively acts as control signals for writing into and reading from the data field of the redundant word. An overflow signal indicates that memory can no longer be repaired if all the redundancy words have been programmed.

B) Mode 2: Normal Mode

During the normal mode each incoming address is compared with the address field of programmed redundant words. If there is a match, the data field of redundant word is used along with the faulty memory location for reading and writing data. The output multiplexer of Redundant Array Logic then ensures that in case of a match, the redundant word data field is selected over the data read out(R'/W=0) of the faulty location in case of read signal. This can be easily understood by the redundancy word detail shown in fig.4.

IV. RELIABILITY ENHANCEMENT:

Hard-to-detect resistive open defects

The fault behavior of a dynamic fault caused by a resistive open defect may vary with the defect size(the resistance value of the defect). For example, if a resistive open defect existing between the Vdd and the pull-up transistor of an 6T SRAM cell, then the defect may cause a data retention fault(DRF)[7] or a DRDF(also called Dynamic destructive read fault)[6]. The testing of DRF in SRAM is difficult. To cope with this problem, some efficient design-for-Testability techniques were proposed. If a resistive open defect in the pull-up transistor of a SRAM cell causes a DRDF, then it is detectable by a read after write operation. But if the size of the Resistive open defect is small then multiple read operation after one write operation are needed to detect it [2],[5]. This fault is regarded as a deceptive multiple read destructive fault (DMRDF)[2]. The sensitizing sequence for DMRDF is $(1\ w0\ r0)^k$ or $(0w1r1)^k$ which means that k read operations after a one write operation are executed. Note that if a cell with DMRDF, then the cell content is correct when the (k-1)th read operations are executed, and the cell content is changed when kth read operation is executed but the read data of the kth read operation is still correct. Therefore, an additional read operation is still correct. Therefore an additional read operation is necessary to observe the fault.

To further investigate the characteristics of the resistive –open defect, we will use a circuit level

IJERT

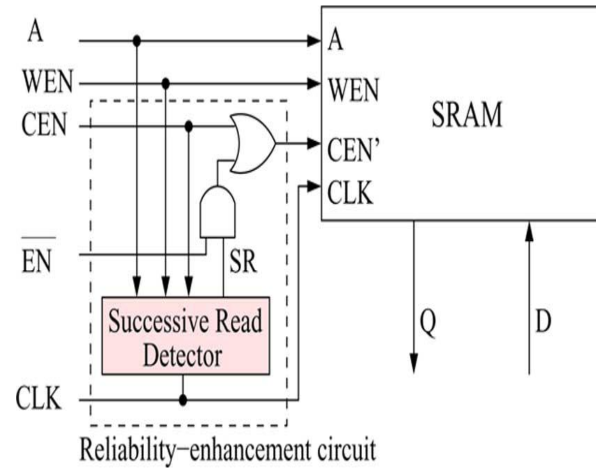
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 3, March - 2013

simulator, Hspice, to analyze the relation between the number of successive read operation and the size of the resistive open defect using 0.18m TSMC CMOS Technology.

Consider an SRAM cell is operated at 1.4ns. If the bit-line capacitance (cb) is 25fF, then one successive read operation required from 0.99 to 1.13M; three successive read operations are required for detecting the defect which size is ranged from 0.979M to 0.99Ω. If the defect size is smaller than 0.979M, then the SRAM cell can be operated correctly.

B. Design for Reliability scheme

If the resistance of a resistive open defect in an SRAM is small, then the SRAM cell may be operated correctly. However, the resistance of the resistive open defect may become large during the lifetime of RAMs. Once the defect size is larger than the threshold value, it causes a dynamic fault in the RAM. Therefore the lifetime of RAMs is reduced. i.e; the reliability of RAMs is reduced. Also, the number of successive read operations required to sensitize the defect is decreasing with the increase of defect size. One simple approach to prolong the lifetime of a RAM cell with this type of latent defects to prevent the RAM cell from successive multiple read operations. Thus, if no successive multiple read operations on a RAM cell is allowed, then the larger defect size can be tolerated for the RAM cell. Fig.5 shows the proposed reliability Enhancement circuit(REC) for RAMs. The REC major consists of successive read detector (SRD). The SRD checks whether successive multiple read operations on the same position are executed. The enable signal (EN) determines if the SRAM is protected by the REC. If EN is logic 1 then successive read (SR) output is blocked. So, CEN'=CEN, if EN is logic 0 then CEN'=CEN│SR, where '│' denotes bit-wise OR operation.



Therefore, if SR=1(i.e; successive read operation on the same address is detected), then CEN' is logic 1 regardless of the logic value of CEN and the SRAM is in idle state.

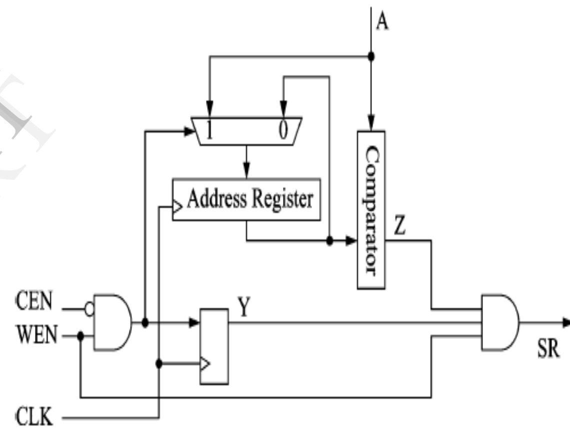


Fig. 2. Detail implementation of the Successive Read Detector.

Fig 6.depicts the detailed implementation of the SRD.

Assume that a SRAM executes three successive read operations on the addresses A1,A1 and A2. The address A1 is stored in the address register when the SRAM executes the first Read operation, since the control signal of the multiplexer is 1(CEN=0 and WEN=1). Also, the Y will be logic 1 after the first Read operation is done. Subsequently, if the SRAM executes the second read operation, the input

IJERT

address A1 is compared with the address A1 stored in the address register. Since the address of the output z of the comparator is logic 1. Therefore, SR=Y&WEN&Z=1 when the second read operation is executed , where & denotes a bit-wise AND operation. So the CEN of the SRAM is forced to logic 1 and the SRAM is in idle state. The data at the output of the SRAM is the same as that of first read operation. Thus we still can get the right data from the SRAM even the SRAM is in idle state. Finally, the SRAM executes the third read operation and the output of comparator is logic 0, since the address stored in the address register is different from the input address. Therefore, the SR becomes 0 and the SRD becomes transparent to the SRAM.

Simulation Results:

Mentor Graphic's modelsim has been used to verify the functionality and timing constraints of Verilog coded BIST module, Repair redundancy array and reliability Enhancement circuit. The full architecture containing all these modules has been successfully synthesized using Xilinx ISE 9.1i. Design synthesis reports shows that total of 136 slices, 107 slice fliflops, 203 4 input LUTs and 13 bonded IOBs have been used in the synthesis. The simulation waveform Reliability Enhancement And Fault Free is shown in fig.7. As the start signal goes high, indicating the start of test, The first March element M0 of March SS algorithm is executed. This being a write signal, no values are read out from the memory to be compared with the expected or correct values and hence output FAULT waveform second read operation is same as the first one, the

of comparator is showing high impedance for some initial clock cycles. As read operation starts at the beginning of execution of M1 element, the values from MUT are read out and compared with the expected values.

Conclusion: As fig.7 shows that, Reliability enhancement circuit works in Normal mode, to improve the life time of SRAM. When the clock is either pos edge os neg edge and when Rst is 1, ENB is 0 then CEN is equal to CENB. So, when CEN=0 then CENB is also 0, hence SRAM will be in idle condition. As fig.8 shows that, it works in two modes. Those are normal mode and Test mode. PS and NS signals represent these two modes. If pS is 1, it works in Test Mode and NS is 1, it works in Normal mode. In Test mode it detects the faults and stores faulty address and data in redundant logic memory (RLmem). In normal mode if WrEna is high, it compares the AddrIn with stored redundant logic address, if match found, it will write the data into RLmem otherwise it will write into normal memory(Mem). In normal Mode if RdEna is high, it compares AddrIn with stored redundant logic address, if match found, it will read the data from the RLmem otherwise it will read from Normal memory(Mem). Data out comprises of fault pulse, fault address and Data. Fault Address is 8 and Data is 0. In Memout it is showing 1. But in Muxout we are getting the correct data.

IJERT

Output Waveforms:

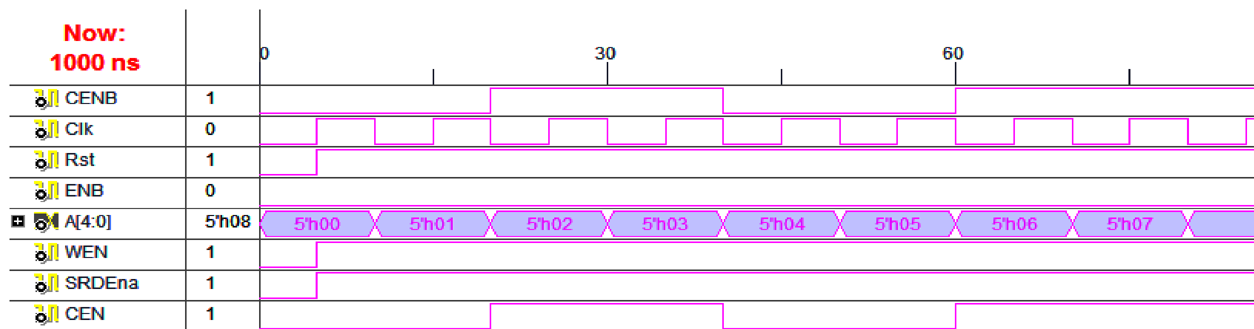


Fig .7 output waveform representing Reliability Enhancement of SRAM

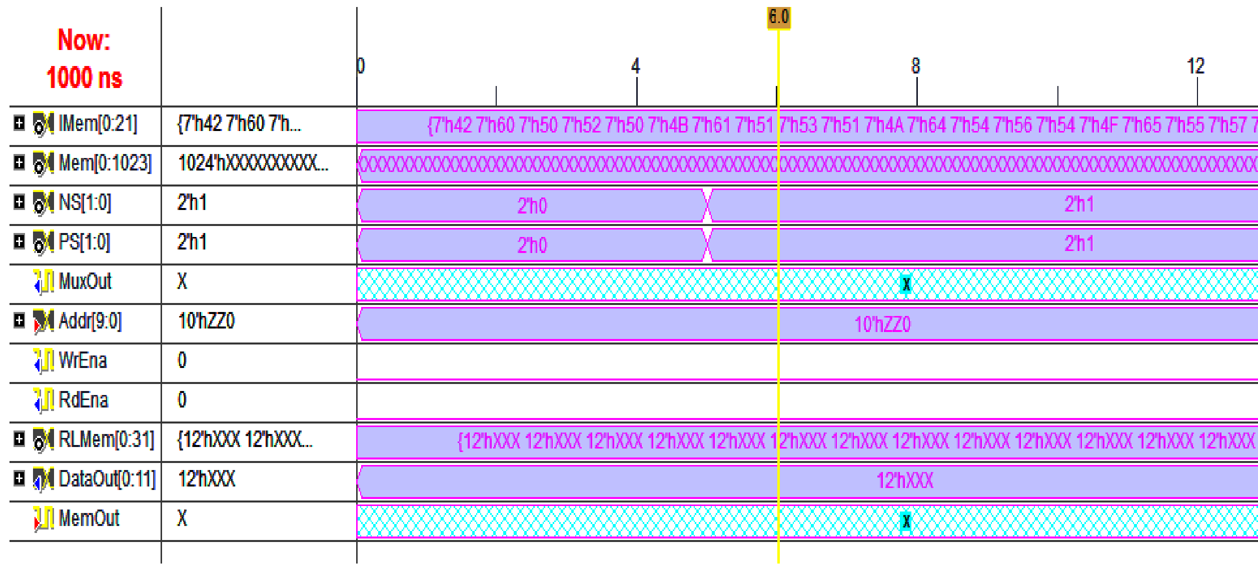


Fig.8 Output waveform representing output data after fault Repairing

REFERENCES

[1] Dr.R.K.Sharma, Aditi Sood,"Modeling and simulation of Multi-operation Microcode-based Built-In self Test For Memory Fault Detection And Reapair"vol 18 ,No.9,Pg no.381-386

[2] jin-Fu Li , Tsu-Wei Tseng and chih-sheng Hou," Reliability Ennhancement and self Repair Schemes for SRAMs with static and Dynamic Faults" vol 18 ,No.9, pg.no.1361-1366

[3] International SCHEMATIC ,"International technology Roadmap for semiconductors(itrs);edition 2001"

[4] s.hamdioui, G.N.Gaydadjiev, A.J.Van de Goor , "state – of –art and Future trends in testing embedded memories", international workshop on memory technology, design and testing(MTDT.04)

[5] s.hamdioui, Z.AL-Ars,A..J.van de Goor, " Testing static and dynamic faults in random accessmemories"

[10] N.B.Jilani ,k.Murali Krishna ," Reliability Enhancement of SRAM and it's Fault Repairing by using March-SS Algorithm" in International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181Vol. 1 I

in proc . of IEEE VLSI test symposium PP-395-400,2002

[6] s.hamdioui , et.al" importance of dynamic faults for new SRAM technologies", in IEEE Proc. Of European Test workshop, pp-29-34, 2003

[7] s.Himadioui ,A.J. Vande Goor and M..Rodgers," March-ss: A test for all statics simple RAM faults ", In proc .of IEEE, Internnational workshop on memory technology, design and testing, pp.95-100, bendor,france,2002

[8] N.Z. Haron , S.A.M Junos , A.S.A. Aziz,"Modeling and simulation of microcode built-in self test architecture for embedded memories", in proc.Of IEEE international symposium

[9]W. needham ,c.prunty,and E.H.yeoh," High volume microprocessor test escape, an analysis of defects our tests are missing" pp.25-34 faults for new SRAMTechnologies"pp.29-34

IJERT