Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 5, May 2013, pg.33 – 40

RESEARCH ARTICLE

MULTIPLE FAULT DIAGNOSIS FOR HIGH SPEED HYBRID MEMORY ARCHITECTURE

Ms. B. Kamala Soundari[1], Ms. M. Praveena[2]

[1]Assistant Professor, Department of Electronics and Communication,
PSNA College of Engineering &Technology, India

[2]Assistant Professor, Department of Electronics and Communication,
PSNA College of Engineering &Technology, India

[1] kamalasoundari@gmail.com; [2] praveenabe88@gmail.com

Abstract— This paper presents a built-in self-test (BIST)-based scheme for fault diagnosis that can be used to identify permanent failures and automatic correction in all memories & circuits. The proposed approach offers a simple test flow and does not require intensive interactions between a BIST controller and a tester. The scheme rests on partitioning of rows and columns of the memory array by employing low cost test logic. It is designed to meet requirements of at-speed test thus enabling detection of timing defects.

Key Terms: - Built-in self-test (BIST); deterministic partitioning; discrete logarithms; embedded read-only memory; fault diagnosis

I. INTRODUCTION

The International Technology Roadmap for Semiconductors predicts memories to occupy more than 90% of the chip silicon area in the foreseeable future. Due to their ultra large scale of integration and vastly complex structures, memory arrays are far more vulnerable to defects than the remaining parts of integrated circuits. Embedded memories have already started introducing new yield loss mechanisms at a rate, magnitude, and complexity large enough to demand major changes in test procedures. Many types of failures, often not seen earlier, originate in the highest density areas of semiconductor devices where diffusions, polysilicon, metallization, and fabricated structures are in extremely tight proximity to each other. Failing to properly test all architectural features of the embedded memories can eventually deteriorate the quality of test, and ultimately hinder yield.

Embedded memories are more challenging to test and diagnose than their stand-alone counterparts. This is because their complex structures are paired with a reduced bandwidth of test channels resulting in limited controllability.

Consequently, the memory built-in self-test (MBIST) has established itself as one of the mainstream design for test (DFT) methodologies as it allows one to generate, compress, and store on chip very regular test patterns and expected responses by using a relatively simple test logic. The available input/output channels, moreover, suffice to control built-in self-test (BIST) operations, including at-speed testing and detection of timing defects. Non-volatile memories are among the oldest programmable devices, but continue to have many critical uses. ROM, PROM, EPROM, EEPROM, and flash memories have proved to be very useful in a variety of applications. Traditionally, they were primarily used for long-term data storage, such as look-up tables in multimedia processors or permanent code storage in microprocessors. Due to the high area density and new sub micrometer technologies involving multiple metal layers, ROMs have also gained popularity as a storage solution for low voltage/low-power designs. Moreover, different methods such as selective pre-charging, minimization of non-zero items, row(s) inversion, sign magnitude encoding, and difference encoding are being

employed to reduce the capacitance and/or the switching activity of bit and word lines. Such design, technology, and process changes have resulted in an increase in the number of ROM instances usually seen in design. New non-volatile memories such as ferroelectric, magnetoresistive, and phase changed RAMs retain data when powered off but are not restricted in the number of operation cycles. They may soon replace other forms of non-volatile memory as their advantages, e.g., reduced standby power and improved density, and are tremendous.

It has become imperative to deploy effective means for testing and diagnosing non-volatile memory failures. A functional model employed for these memories remains similar to that of RAMs with relevant fault types such as stuck-ats and bridges being tackled through functional test algorithms. Also, all addressing malfunctions are covered by memory cell stuck-at fault tests as there are no writes in the mission mode. Typically, the basic test reads successive memory cells and processes output responses by performing a polynomial division to compute a cyclic redundancy code (signature). The same procedure can be used to detect certain classes of dynamic faults provided memory cells are designed with additional DFT features [14]. No longer, however, is it sufficient to determine whether a memory failed or not [3].

II. TEST LOGIC ARCHITECTURE

2.1. Memory Array Organization

Every row consists of *M* words, each *B*-bit long. Bits belonging to one word can be either placed one after another or interleaved forming segments, as illustrated in the figure. Decoders guarantee the proper access to memory cells in either a fast row or a fast column addressing mode, i.e., with row numbers changing faster than word numbers or vice versa. It is worth noting that algorithms proposed in this paper do not impose any constraints on the addressing scheme so that the memory array can be read using either increasing or decreasing address order. Fig. 1 shows the salient architectural features of a ROM.

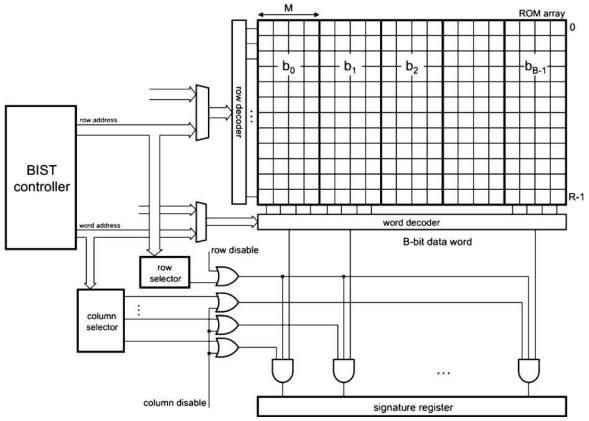

Fig. 1 Memory array architecture and diagnostic environment

2.2. Collection of Diagnostic Data

The same Fig. 1 summarizes the architecture of a test environment used to collect diagnostic data from the ROM arrays. In addition to a BIST controller, it consists of two modules and gating logic that allow selective observation of rows and Fig. 1. Memory array architecture and diagnostic environment. Columns, respectively. Assuming permanent failures, the BIST controller sweeps through all ROM addresses repeatedly while the row and column selectors decide which data arriving from the memory rows and/or columns is actually observed by the signature register. Depending on a test scenario, test responses are collected in one of the following test modes.

1) Row disable = 0 and column disable = 1; the row selector may enable all bits of the currently received word, thereby selecting a given row; this mode is used to diagnose row failures and, in some cases, single cell faults.

2) Row disable = 1 and column disable = 0; assertion of the row disable signal effectively gates the row selector off; the column selector takes over as it picks a subset of bit lines to be observed (this corresponds to selecting desired columns and is recommended to diagnose column and single cell failures).

3) Row disable = 0 and column disable = 0; de-asserting both control lines allows observation of memory cells located where selected rows and columns intersect; Fault diagnosis has a simple flow. It proceeds iteratively by determining a signature, which corresponds to the selected rows or columns, followed by a transfer of such a test response to the ATE through an optional shadow register. If the obtained signature matches the reference (golden) signature, we declare the selected rows and/or columns fault-free. Time required to filter out failing sites accurately depends on how selection of observable rows and columns is carried out. Our scheme employs an enhanced version of deterministic partitioning originally proposed for scan-based diagnosis

[4]. It assures the fastest possible identification of fault sources down to the array nodes that cannot be recognized as fault-free ones.

2.3. Signature Register

A signature register is used to collect all test responses arriving from selected memory cells. The register is reset at the beginning of every run (test step) over the address space.

Similarly, the content of the register is downloaded once per run. A multiple input ring generator (MIRG) driven by the outputs of gating logic is used to implement the signature register. It is worth noting that connecting each input to uniquely selected stages of the compactor makes it possible to recognize errors arriving from different input channels. This technique visibly improves diagnostic resolution, as is demonstrated in the following sections.

III. ROW AND COLUMN SELECTION

In this section, we introduce several hardware solutions for row and column selection. In particular, after presenting separate row and column selectors that implement a deterministic partitioning of a ROM array, we introduce a scheme that allows one to partition rows and columns simultaneously.

3.1. Row Selection

We start by introducing the general structure of the row selector shown in Fig. 2. Essentially, it is comprised of four registers. The up counters *partition* and *group*, each of size $n = \_0.5 \log 2\ R\_$, keep indexes of the current partition and the current group, respectively.

They act as an extension of the row address register that belongs to A linear feedback shift register (LFSR) with a primitive characteristic polynomial implements a *diffractor* providing successive powers of a generating element of GF(2*n*), which are subsequently used to selectively invert data arriving from the partition register. The same register can be initialized when its input load is activated

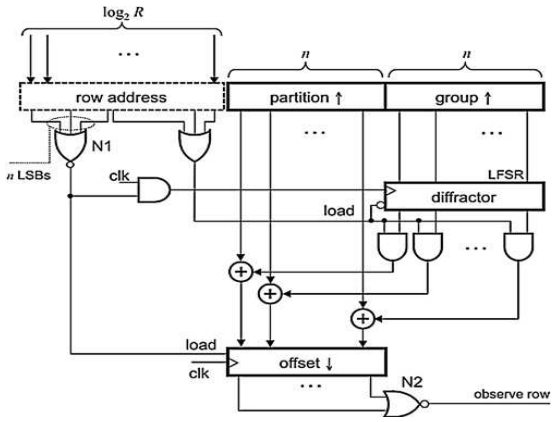

Fig. 2. Row selector

Similarly, one can initialize a down counter called *offset* by asserting its input load.

3.2. Column Selection

Fig. 3 shows the column selector used to decide, in a deterministic fashion, which columns should be observed.

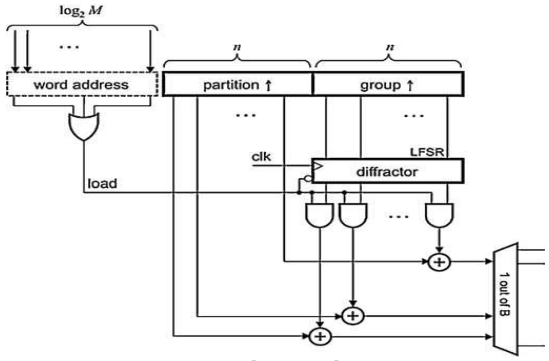

Fig. 3 Column Selector

Its architecture resembles the structure of the row selector as both circuits adopt the same selection principles. The main differences include the use of a BIST column address register and a diffractor clocking scheme. Moreover, the offset counter is now replaced with a combinational column decoder, which allows selection of one out of *B* outputs of the word decoder (see Fig. 1). It is worth noting that the diffractor advances every time the column address increments. Its content added to the partition number yields a required column address in a manner similar to that of the row selection.

3.3 Combined Row and Column Selection

In order to reduce the area overhead, some components of the row selector and the column selector can be shared.

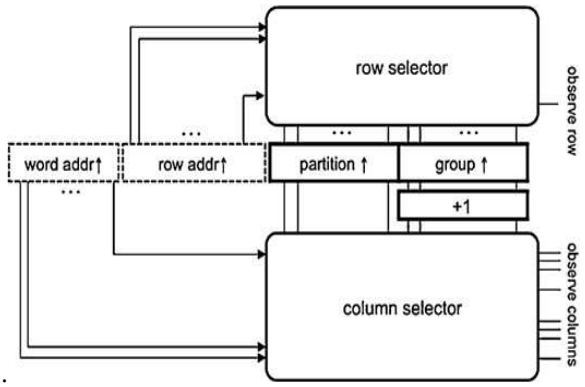

Fig. 4.Combined row and column selector

The circuit by which this concept is implemented is shown in Fig. 4where the partition and group registers feed both selectors. Since the word address increments prior to the row address, the memory array is read in the fast column addressing mode. As no interaction between control signals arriving from the word and row address registers is needed, the scheme enables reading the memory array in the fast row mode as well, after exchanging the row and column address registers. Furthermore, the combined row and column selector is designed in such a way that none of the components require clock faster than the one used to increment either the word or row address register. As a result, the proposed scheme allows reading memory at-speed, and thus detection of timing defects. Finally, as the combined selector makes it possible to collect the row and column signatures in parallel, such an approach allows one to reduce the diagnostic time by half. In this mode, however, two signature registers are required.

3.4. Trellis Selection

Given $x + 1$ groups of signatures, the selection schemes presented earlier allow one to identify correctly up to either The actual failure may comprise, however, faults occurring in rows and columns at once. illustrates a failure that consists of a single stuck-at column and a single stuck-at row. The black dots indicate failing cells

assuming a random fill—note that some cells of the faulty row and column store the same logic values as those forced by the fault. If diagnosed by using separate selection of rows and columns, such a fault would affect most of signatures as cells belonging to the failing column make almost all row signatures erroneous, and cells of the failing row would render almost all column signatures erroneous, as well.

IV. THE BIST ARCHITECTURE

.

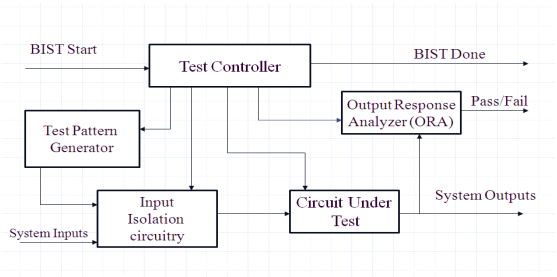

Figure 5 The BIST Architecture

The BIST architecture can be simple or complicated based on the purpose of the test being performed on the circuit. A basic BIST architecture for testing an FPGA includes a controller, pattern generator, the circuit under test, and a response analyzer.

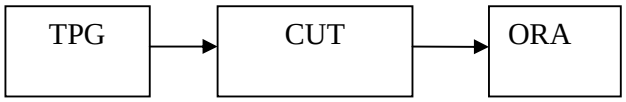

Figure 6 Basic Blocks of BIST

4.1. LINEAR FEEDBACK SHIFT REGISTER

A BIST have a pseudo-random binary sequence (PRBS) generator and a signature register. The PRBS generator is most easily implemented using a linear feedback shift register (LFSR). A PRBS generator allows us to generate all of the required binary patterns for the circuit under test. The LFSR can be used to both generate the test sequence for the design that is to incorporate BIST and with slight modification can be used to capture the response of the design and generate a signature.

An ML-LFSR enables us to create a PRBS generator. This ML-LFSR is going to be a 10-bit design. The principles outlined for this 10-bit design apply to any N-bit ML-LFSR. In order for the 10-bit LFSR to be maximal length, the choice of the inputs to the lfsr_tap. The maximal length (ML) LFSR generates data that is almost random. The output of the LFSR can be taken in parallel-out form or as a serial bit stream. The serial bit stream is usually taken from the MSB of the Test Pattern Generator.

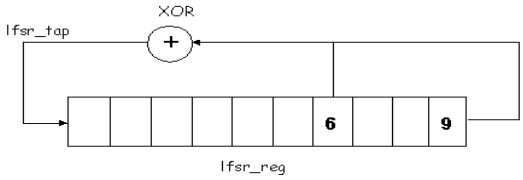

Figure 7 Linear Feedback Shift Register

It is a unit which is used for generating the test signals; this is done by methods called Exhaustive / Pseudo Exhaustive Testing.

In exhaustive testing all possible input patterns are applied to the circuit under test. Thus to an n input combinational circuit all possible 2^n patterns are applied to the circuit. The advantage of this approach is that all redundant faults can be detected. A modified form of exhaustive testing is Pseudo Exhaustive Testing.

It retains the advantages of exhaustive testing while significantly reducing the number of test patters to be applied. The basic idea is to partition the circuit under test into several sub circuits such that each sub circuit has a few enough inputs for exhaustive testing to be feasible for it. If the output response of the circuit under test, do not match the expected response when the stored test patterns are applied in the presence of a fault.

4.2. CIRCUIT UNDER TEST:

CUT is simulated with a random pattern sequence of a random length. The pattern is then generated by an algorithm and implemented in the hardware. If the response is correct, the circuit contains no faults. The problem with pseudo-random testing is that is has a low fault coverage unlike the exhaustive pattern generation method. It also takes a longer time to test.

4.3. TEST RESPONSE ANALYZER

The most important part of the BIST architecture is the test response analyzer (TRA). Like the pattern generator, its uses one output generator. It is designed based on the diagnostic requirements. The response analyzer usually contains comparator logic. Two comparators are used to compare the output of two CUTs. The two CUTs must be exact. The registered and unregistered outputs are then put together in the form of a shift register. The function generator within the response analyzer compares the outputs. The outputs are then OR ed together .Once compared, the function generator gives a response back of a high or low depending on if faults are found or not.

4.4. THE BIST PROCESS

The BIST process diagram was shown in the Form the figure 3.4. The test controller is used to start the test process. The pattern generator produces the test patterns that are inputted into the circuit under test. The CUT is only a piece of the whole FPGA chip that is being tested on and found within a configurable logic block or CLB The FPGA is not tested all at once but in small sections or logic blocks. After a test vector scans the CUT the output of the test is analyzed in the response analyzer. It is compared against the expected output.

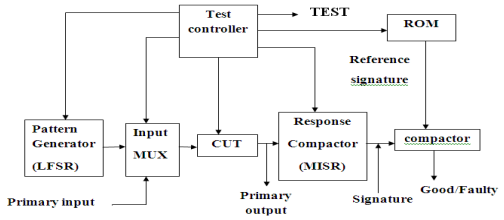

Figure 8 BIST Process

V. SINGLE CELL FAILURES

The methods presented in the previous section allow identification of failing sites with single-row and/or single-column accuracy.

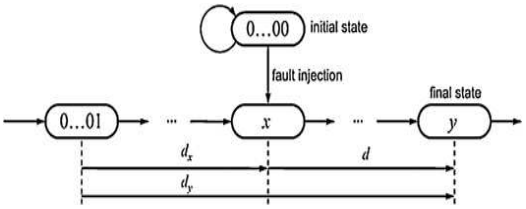

Fig 9 Single cell failure diagnosis

It is also possible to take diagnosis a step further and determine location of a single faulty cell within a row or a column. This section summarizes this approach.

VI. EXPERIMENTAL RESULTS

This section reports results of experiments carried out to characterize performance of the proposed diagnostic scheme. In particular, a diagnostic coverage is used as a primary figure of merit. Assuming that we target up to x failing rows or columns, all numbers presented in this section have been obtained by adopting the following procedure.

1) Run tests for $x + 1$ column partition groups. Let xc be the resultant number of failing columns.

2) Repeat the same tests for $x+1$ row partition groups. Let xr be the resultant number of failing rows.

3) If neither xc nor xr is less than or equal to x, then carry out the trellis selection and stop. Otherwise:

4) If $xc \leq x$, then: Examine signatures (one per a failing column) collected in step (1) against single cell faults (by using the discrete logarithms-based counting).

5) If $xr \leq x$, then: Examine signatures (one per a failing row) collected in step (2) against single cell faults.

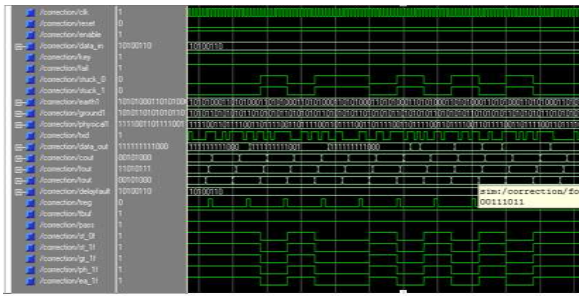

Fig 10.Simulation Output

VII. HARDWARE OVERHEAD

The silicon area of test logic amounts to a certain number of gates and flip-flops. The number of gates depends on the memory word size, which in turn affects the number of inputs of the signature register and the size of its XOR injection network. Furthermore, the number of gates is implied by the mux factor *M*, as the ratio of *M* and *B* determines the number *c* of columns to be observed at a time, and thereby gives the number of phase shifters, and thus XOR gates. Table VI provides the actual area costs computed with a commercial synthesis tool for four memory arrays of different capacity and architecture.

All components of our test logic were synthesized using a 90 nm CMOS standard cell library under 3.5 ns timing constraint. For indicated memory sizes and the relevant parameters *n* and *τ*, the table reports the resultant silicon area with respect to combinational and non-combinational devices, as well as their interconnection network. The total area taken by the proposed test logic is subsequently compared with the corresponding ROM array area (determined based on independent data provided by two silicon manufacturers).

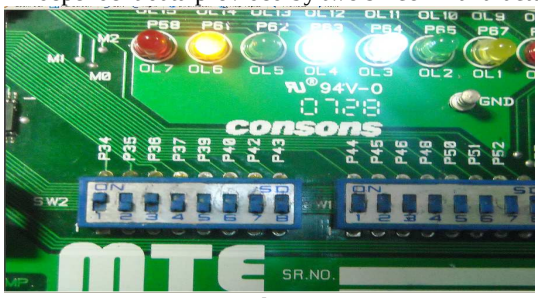

Fig 11.Hardware Output

VIII. CONCLUSION

In this paper, we proposed a new fault diagnosis scheme for embedded read-only memories. It reduces the diagnostic data that needs to be scanned out during ROM test such that the minimum information to recover the failure data is preserved, and the time to unload the data is minimized. The presented approach allows an uninterrupted collection and processing of test responses at the system speed. This has been achieved by using low-cost on-chip selection mechanisms, which are instrumental in very accurate and time efficient identification of failing rows, columns, and single memory cells. In particular, the scheme employs the original designs of row and column selectors with phase shifters controlling the way the address space is traversed. Furthermore, the new combined selection logic allows the scheme to collect test results in parallel (leading to shorter test time) without compromising quality of diagnosis. Results of experiments performed on several memory arrays for randomly generated failures clearly confirm high accuracy of diagnosis of the scheme provided the signature registers and the proposed selection logic are properly tuned to guarantee a desired diagnostic resolution.

ACKNOWLEDGEMENT

The authors would like to acknowledge a private communication with F.Poehl of Infineon Technologies AG, Munich, Germany, concerning logic synthesis of semiconductor memories.

REFERENCES

[1] B. Kamalasoundari and M. Praveena, “Multiple Fault Diagnosis For High Speed Hybrid Memory Architecture”. ICACC’12,jan 2012.

[2] D. Appello, V. Tancorre, P. Bernardi, M. Grosso, M. Rebaudengo,and M. Sonza Reorda, “Embedded memory diagnosis: An industrial workflow,” in Proc. ITC, 2006, paper 26.2.

[3] S. Barbagallo, A. Burri, D. Medina, P. Camurati, P. Prinetto, and M.Sonza Reorda, “An experimental comparison of different approaches to ROM BIST,” in Proc. Eur. Comput. Conf., 1991, pp. 567–571.

[4] I. Bayraktaroglu and A. Orailoglu, "The construction of optimal deterministic partitioning in scan-based BIST fault diagnosis: Mathematical foundations and cost-effective implementations," IEEE Trans. Comput.,vol. 54, no. 1, pp. 61–75, Jan. 2005.
[5] T. J. Bergfeld, D. Niggemeyer, and E. M. Rudnick, "Diagnostic testing of embedded memories using BIST," in Proc. DATE, 2000, pp. 305–309.
[6] T. Boehler and G. Lehmann, "Using data compression for faster testing of embedded memory," U.S. Patent 6 950 971, Sep. 27, 2005.
[7] J. T. Chen and J. Rajski, "Method and apparatus for diagnosing memory using self-testing circuits," U.S. Patent 6 421 794, Jul. 16, 2002.
[8] J. T. Chen, J. Rajski, J. Khare, O. Kebichi, and W. Maly, "Enabling embedded memory diagnosis via test response compression," in Proc.VTS, 2001, pp. 292–298.
[9] J. T. Chen, J. Khare, K. Walker, S. Shaikh, J. Rajski, and W. Maly, "Test response compression and bitmap encoding for embedded memories in manufacturing process monitoring," in Proc. ITC, 2001, pp. 258–267.
[10] D. W. Clark and L.-J. Weng, "Maximal and near-maximal shift register sequences: Efficient event counters and easy discrete logarithms," IEEE Trans. Comput., vol. 43, no. 5, pp. 560–568, May 1994.

