An Integrated ECC and Redundancy Repair Scheme for Memory Reliability Enhancement

Chin-Lung Su, Yi-Ting Yeh, and Cheng-Wen Wu
Laboratory for Reliable Computing (LaRC)
Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 30013

Abstract—*With the fast development pace of deep submicron technology, the size and density of semiconductor memory grows rapidly. However, keeping a high level of yield and reliability for memory products is more and more difficult. Both the redundancy repair and ECC techniques have been widely used for enhancing the yield and reliability of memory chips. Specifically, the redundancy repair and ECC techniques are conventionally used to repair or correct the hard faults and soft errors, respectively. In this paper, we propose an integrated ECC and redundancy repair scheme for memory reliability enhancement. Our approach can identify the hard faults and soft errors during the memory normal operation mode, and repair the hard faults during the memory idle time as long as there are unused redundant elements. We also develop a method for evaluating the memory reliability. Experimental results show that the proposed approach is effective, e.g., the MTTF of an* $32K \times 64$ *memory is improved by 1,412 hours (7.1%) with our integrated ECC and repair scheme.*

1 Introduction

The use of memory cores in system-on-chip (SOC) designs is growing rapidly. According to the recent ITRS report [1], memory cores represent a significant portion of a typical SOC, and dominate the chip yield. Built-in self-test (BIST) [2] has been widely used for reducing embedded memory testing cost. It is widely accepted by memory designers to implement redundancy repair schemes to improve the yield of memory products [3], i.e., memories with redundancy is commonly seen today, where redundant elements are used to replace faulty elements. Redundancy provides fault tolerance in chips and hence improves the product yield. Row and/or column redundancy are common types of redundant elements in use. For embedded memories, built-in self-repair (BISR) is considered a feasible solution for yield enhancement [4]. In [5], the author defines some infrastructure IPs (IIPs) for improving yield, and memory BISR circuits are considered as a kind of IIP. Conventionally, Error Correction Codes (ECC) are the most popular technique for memories to recover from soft error attacks. Parity bits are stored along with information bits, and certain number of errors in a codeword can be detected and even corrected depending on the employed ECC scheme. ECC has also been used at the board level for more than 35 years in memory systems [6]. However, with the advent of deep submicron technology, memory chips are becoming larger and denser, so soft errors are playing a more critical role in quality and reliability of memory products [7]. CMOS based semiconductor memories are especially prone to radiation induced soft errors [8]. The chip reliability decreases as the soft error rate increases, requiring on-chip ECC schemes to be implemented in more and more memory devices to enhance the memory reliability [9].

Although ECC and redundancy repair are both widely used fault tolerance techniques today, they have different objectives as they are used, i.e., the redundancy repair and ECC techniques are conventionally used to repair or correct the hard faults and soft errors, respectively. It, however, is interesting to note that both of them may be used to enhance the memory reliability and yield. Schemes which combine ECC and redundancy have been proposed to enhance the yield of memories recently [10, 11]. They use redundancy to repair clusters of faulty bits, and use ECC to detect and correct other isolated faulty bits. Nevertheless, while ECC is used for repairing fabrication faults, the chip becomes

Proceedings of the 2005 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)
0-7695-2464-8/05 $20.00 © 2005 IEEE

less immune to soft errors such that the reliability is harmed. In order to avoid the reliability loss, hard faults detected during production test should be repaired by redundancy elements if possible, instead of relying on ECC, especially when the soft error rate is high. After the production test, there may be some redundant elements left unused, which become our surplus. As the parametric defects are more and more difficult to fully detect by using the traditional burn-in method within an acceptable cost range, a more cost-effective repair scheme to detect and repair the faults caused by increasing parametric defects under normal operation is important.

In this paper, we proposed a new scheme to identify hard faults and soft errors, and progressively repair the hard faults by available redundancy in normal operation mode. This new scheme integrates the on-line ECC method and an off-line BISR scheme. Based on our previous work [12, 13], we also present a reliability evaluation procedure and tool that helps evaluate the reliability improvement of the integrated scheme.

The rest of the paper is organized as follows. Section 2 gives the preliminaries of the ECC and redundancy repair schemes. Section 3 introduces the proposed integrated ECC and redundancy repair scheme. In Sec. 4 we present the details of the memory test wrapper design. Evaluation of reliability enhancement is described in Sec. 5, and in Sec. 6 the experimental results are shown. Finally, Sec. 7 concludes the paper.

2 Preliminaries

We propose an integrated ECC and redundancy repair scheme which combines the existing off-line built-in self-repair (BISR) and on-line error correction code (ECC). This new scheme takes the advantages of unused redundant elements and ECC, operating both techniques on-line. During normal operation of the system, we start the repair mode using the memory idle cycles after an error is identified as a hard fault.

The failure modes of memory chips are generally classified into two main classes, i.e., those lead to hard faults and those to soft errors. Both are assumed to cause the memory chip to fail, so they affect the memory chip's reliability. There are three common sources of soft errors: 1) alpha particles, 2) high-energy cosmic ray induced neutrons, and 3) neutron induced B fission [14]. Soft errors can be eliminated after reading and re-writing corrected data back, but hard faults, though can be corrected as well by ECC, cannot be eliminated as the defects do not disappear over time. The sources of hard faults are also classified into three categories: 1) random defects, 2) systematic defects, and 3) parametric defects. Random defects have been thoroughly studied in the past [15]. The principal sources of the random defects are particle caused hard bridge (short) or open defects. Systematic defects are from anything with spatial or time-based issues during design, fabrication, packaging, etc. Parametric defects affect the device electrical parameters, mainly caused by process variations and uncertainties. ECC is known to be capable of detecting and correcting soft errors and enhancing the memory reliability [8]. For example, with a single-error correctable ECC, the memory will remain functionally correct if no more than one error occurs in each codeword. Hamming codes, which have low-complexity encoding and decoding schemes, are an efficient class of ECC. In [10], redundancy is used to repair clusters of faulty bits and ECC is used to cover single-cell faults (hard errors) during the production test. The scheme results in higher yield and greater degree of fault tolerance because it uses ECC in addition to redundant sections to repair hard faults. However, the ability of the ECC to tolerate soft errors thus reduces.

In the proposed error identification method for both hard faults and soft errors, we combine the two different fault tolerance techniques, i.e., redundancy repair and ECC, to increase the memory yield and avoid the loss of reliability. During normal operation, if an error is identified as a soft error, we correct it with ECC and no other actions will be made. If it is identified as a hard fault, we repair it by redundancy during the memory idle cycles (or off-line) if there are available redundant elements. With this method, we can increase the effectiveness of the redundancy and ECC schemes so far as the system reliability is concerned. Details will be given next.

3 Integrated ECC and Redundancy Repair Scheme

The proposed scheme is divided into two phases, i.e., the Error Identification phase and the Hard Repair phase. The phase (state) transitions are shown in Fig. 1. During fault-free normal operation, the fault tolerance mechanism is not

Proceedings of the 2005 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)
0-7695-2464-8/05 $20.00 © 2005 IEEE

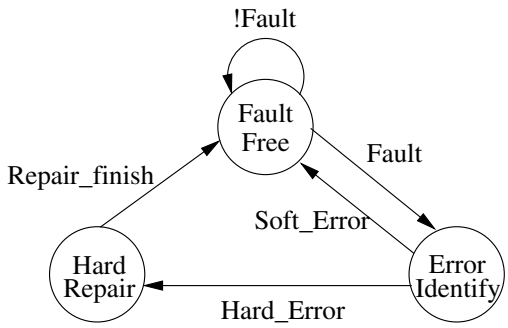

Figure 1: Phase transition of the proposed scheme.

activated, and the system stays in the Fault Free state. When a fault is detected by ECC, the system will immediately executes an Error Identification procedure to determine whether the fault is a hard fault or a soft error. After error identification the system may transit either to the Fault Free state or the Hard Repair state, depending on the error type identified. Assume that during the Error Identification phase, no other errors may occur. This assumption is reasonable because the error rate is normally very low compared with the system clock speed. We need to develop a method to distinguish between hard faults and soft errors. Hard faults are permanent, while soft errors only remain within a short interval called the scrubbing interval. According to its property, when an error is detected by the ECC circuit, the newly corrected data are written back and then the data are decoded again to see if they are still incorrect. If the data are still incorrect, the write-back and decoding steps are repeated. These repeated steps continue for a time interval which is slightly larger than the scrubbing interval. If the decoded data are eventually correct within the interval, then this detected error is considered a soft error, otherwise, we identify it as a hard fault. When a hard fault is identified, the system enters the Hard Repair state, and the ECC circuit sends a signal to hold the system's normal operation until the repair process is done correctly. Alternatively, the repair process can be postponed until the memory enters idle cycles. This can be controlled by the system software.

In the Hard Repair phase, there are two cases in the location of the hard fault. In the first case, the hard fault occurs in the main memory. If there are available redundant elements, the hard fault is repaired by spare allocation through a redundancy analysis algorithm. The faulty-word address is now remapped to a redundant-word address by a reconfiguration circuit (RC). The original data of the faulty cells are written into their corresponding redundant memory cells. In the second case, the hard fault occurs in the redundant memory. In addition to the operations as done in the first case, we mark the faulty redundant elements. This is to avoid memory failure due to the conflict in the RC circuit. During the repair process, the memory cannot be accessed. This can also be postponed till the memory is in the idle mode. As long as hard faults are removed in the Hard Repair phase, they will not degrade the reliability of the memory. In other words, the reliability is increased with respect to the original system, hence the mean time to failure (MTTF) is also increased.

4 Memory Test Wrapper

BIST is a popular scheme for reducing the test cost of embedded memories. An effective BIST design that we adopt is given in [16]. We now describe the implementation of the integrated ECC and redundancy repair scheme. Each memory in the proposed integrated scheme should be encapsulated by the memory test wrapper, which is described in what follows.

4.1 Hardware Architecture

The block diagram of the memory test wrapper is shown in Fig. 2. The operation mode of the memory is selected by the Memory BISR Select (MBS) signal. The BIST circuit receives the test commands from the tester and generates test signals for the memory core (containing information and parity bits) during the test mode. When a fault is detected, it

Proceedings of the 2005 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)
0-7695-2464-8/05 $20.00 © 2005 IEEE

sends the fault information, i.e., faulty word address and fault syndrome of the corresponding word, to the tester. With this fault information, the redundancy analysis algorithm decides how to repair these faults more efficiently. Then, the repair (redundancy) word address is stored in the RC circuit to replace the faulty word address. When the memory core receives a normal operation request, the user address is compared with the faulty-cell addresses recorded in RC. If there is a hit, the user address should be replaced with a redundant address on-line with the redundant match (RM) signal, i.e., the redundant memory (RMEM) is activated to replace the faulty parts of the memory (MEM). The user address is said to be remapped to the redundant address, and the memory is said to be reconfigured. With ECC circuits implemented, each word in RMEM and MEM contains the data and parity bits. The ECC encoder (ECCE) is used to generate the parity bits for data stored into the memory array and the ECC decoder (ECCD) is used to detect and correct the errors of data stored in the memory.

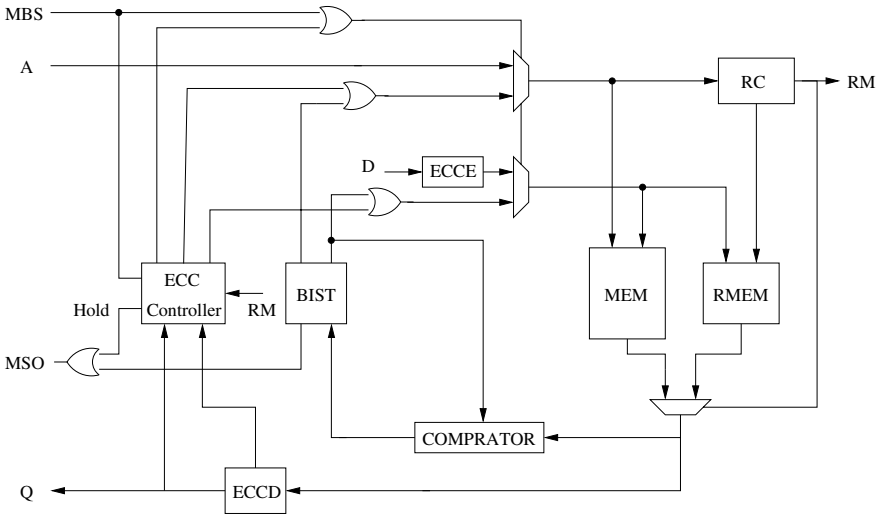

Figure 2: The memory test wrapper block diagram.

When a faulty word is detected by ECCD during normal operation mode, the ECC controller generates a Hold signal to break the normal operation and switch to error identification mode. In this mode, the ECC controller may do some read and write operations to/from the faulty word and repair this fault if necessary. The RM signal also connects to the ECC Controller to indicate whether the faulty word located in the MEM or RMEM. When the faulty word is located in RMEM, the repair scheme is more complex, and is described in Sec. 4.2. There are several OR gates in the memory test wrapper. In error identification mode, the multiplexers are switched to the memory BIST path. Corrected data, address and other operation signals generated by the ECC Controller may be applied to the memory array through the OR gates. Instead of using additional multiplexers, we use OR gates to avoid timing penalty on memory normal operation path. Also, the Hold signal is connected to an OR gate with the BIST data output port such that the pin count does not increase. The reconfiguration circuit (RC) is slightly modified based on our previous work, whose details can be found in [4].

4.2 ECC Controller

When an error is detected by ECC under normal operation, the error is either caused by a hard fault or is a soft error. We need to identify the error type first. The hard faults can be repaired by the unused redundancy. On the other hand, the soft errors can be eliminated after some correction-and-write-back iterations and the circuit is fault free again. In this section, the details of the ECC Controller are described.

The ECC control flow contains two phases—the Error Identification phase and Hard Repair phase. The ECC Controller itself has a Control Module, whose function can be represented by the state diagram as shown in Fig. 3. At the beginning, the reset signal, Reset, forces the ECC Controller to the initial state, i.e., the FFR state. When the ECCD detects a fault under normal operation and there are unused redundant elements, i.e., the FR signal is set, the

Proceedings of the 2005 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)
0-7695-2464-8/05 $20.00 © 2005 IEEE

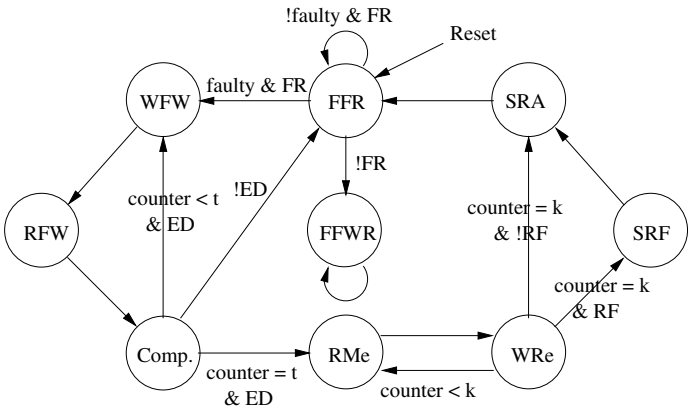

Figure 3: State diagram for the ECC Controller.

ECC Controller enters the Error Identification phase and sets the hold signal, Hold, to hold the normal operation. In this phase, the Controller first writes the corrected data back to the faulty word in the memory in the write-back faulty word state (WFW). Then, the data stored in the word is read and decoded in the read faulty word state (RFW). Finally, we check whether the data is faulty or not in the compare state (Comp.). The three operations, write, read and check, are done several times to ensure that the faults caused by soft errors are eliminated. If the fault is eliminated during the iterations, the Controller transits to the fault-free state and resets the hold signal to resume the normal operation. After t iterations of the three operations, we identify this fault as a hard fault and repair it by one of the unused redundant elements.

Depending on the redundancy architecture, e.g., redundant row, column, or block, the data stored in the same element (row, column, or block) should be copied to their corresponding redundant element. To copy the data, the Controller transits to the read memory (RMe) state to read the data from memory. Then the corrected data are written to unused redundancy in the write redundancy (WRe) state. Until all data stored in the same element are copied, the repaired element address needs to be recorded in the RC circuit. As described in Sec. 3, when the hard fault is located in the redundant memory with redundant faulty signal (RF), the fault flag, FF, of the corresponding redundancy needs to be set in order to mark the faulty redundancy in the set redundancy faulty (SRF) state. Then the repair address is the new redundancy used to repair this fault in the set redundancy address state (SRA). If the fault is located in the main memory instead of the redundant memory, the Controller directly transits to the SRA state without going through the SRF state. Finally, the integrated ECC and redundancy scheme finishes and the Controller transits to the fault-free state, and resets the hold signal to continue the normal operation. If there is no unused redundant element, the proposed scheme is interrupted and the Controller transits to the fault-free without redundancy (FFWR) state.

5 Reliability Evaluation

We now present some analysis in support of our integrated scheme. We propose a reliability evaluation flow as shown in Fig. 4. The main components are as follows: memory cell analysis, failure and fault bitmap generator, redundancy analysis/allocation (RA) algorithm and reliability evaluation. The memory cell analyzer analyzes the layout of memory cell based on our previous work [12]. Considering the particle defects or other defects during manufacturing, the analyzer supports different percentages for different failure types and fault types. For example, the percentages of single-cell fault, two-cell fault, faulty row and faulty column can be 50%, 40%, 5% and 5%, respectively. The analyzer may also generate the percentages of different fault types, such as the stuck-at-0 fault (SAF0), stuck-at-1 fault (SAF1), transition fault (TF), etc. With a given percentage allocation, the failure and fault bitmap generator generates the location sequence of the detected faults. Using this fault detection sequence, selected RA algorithm, and the redundancy organization, we can evaluate the distribution of the number of unused redundancy in the memory chips after all the detected faults are repaired in the production after the test and repair process. This evaluation can be achieved by improving our previous repair rate simulator [13]. Finally, with the number of unused redundancy, the reliability of the memory with redundancy can be calculated.

Proceedings of the 2005 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)
0-7695-2464-8/05 $20.00 © 2005 IEEE

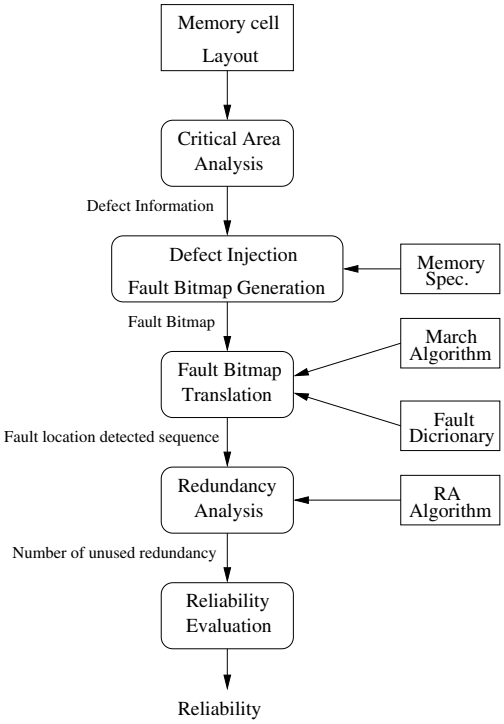

Figure 4: Evaluation flow.

Hard errors can not be eliminated, only corrected, by ECC. Since hard faults accumulate along with time, the reliability, $R(t)$, of a memory chip is also influenced by the hard fault rate. The reliability function $R(t)$ indicates the probability that a chip in the targeted set of chips stays alive up to time t. In order to examine the improvement of the reliability with our proposed methodology, we model the reliability of the memory with the integrated ECC and redundancy repair scheme. We use the Poisson distribution to determine the probability of error in a codeword. Of course other type of distribution can be adopted where appropriate.

We discuss here only the influence of hard errors—errors caused by hard faults. By assuming that errors occur independently in each codeword in a memory chip, the reliability function of a chip can be expressed as

$$R(t) = [(e^{-\lambda_b t})^{CW} + C_1^{CW}(1 - e^{-\lambda_b t})(e^{-\lambda_b t})^{CW-1}]^N, \tag{1}$$

where λ_b, CW and N denote the hard error rate, the codeword length and the number of words in the memory, respectively. The first term in the right-hand side indicates the probability that there is no error, while the second term indicates that there is one error occurring in the codeword.

With the scheme proposed in Sec. 3, we can improve the reliability by combining hard repair and ECC. Whenever an error is identified as a hard error, the hard repair mechanism is invoked. The faulty element is replaced by an unused redundant element, and thus the hard fault is removed. Therefore, the reliability of the memory remains to be one (1) until no spare elements are available for hard repair anymore. Assume we have redundant rows and columns. Let there be x unused redundant rows and y unused redundant columns on the chip after the production test, and the products are shipped and in field use. Let t' be the time when the expected number of faults equals the number of unused redundancy, i.e., $x + y$. The new reliability function is shown below in (2), where B is the number of bits in the memory.

$$R_{x,y}(t) = \begin{cases} 1, & (1 - e^{-\lambda_b t}) \cdot B \leqslant (x + y) \\ R(t - t'), & \text{otherwise} \end{cases} \tag{2}$$

In our previous work, we developed a simulator for simulating the memory repair rate, given the redundancy and fault information [13]. We modify the simulator to simulate the probability of different combinations of unused

Proceedings of the 2005 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)
0-7695-2464-8/05 $20.00 © 2005 IEEE

redundant elements after the memory goes through the production test and repair procedure. With the probabilities (weights) of all the possible combinations, we can compute and draw the average reliability curve for a given error rate. The new reliability function can be expressed as the sum of products of each reliability function and its corresponding probability:

$$R'(t) = \sum_{\substack{0 \leqslant x \leqslant rr \\ 0 \leqslant y \leqslant rc}} P_{x,y} \cdot R_{x,y}(t) \tag{3}$$

Since Hamming code have the advantage of smaller area overhead and less time-consuming for encoding and decoding when implemented in hardware, it is the most commonly used error correction code in practice. Therefore, we use memories with Hamming codes as examples in this paper.

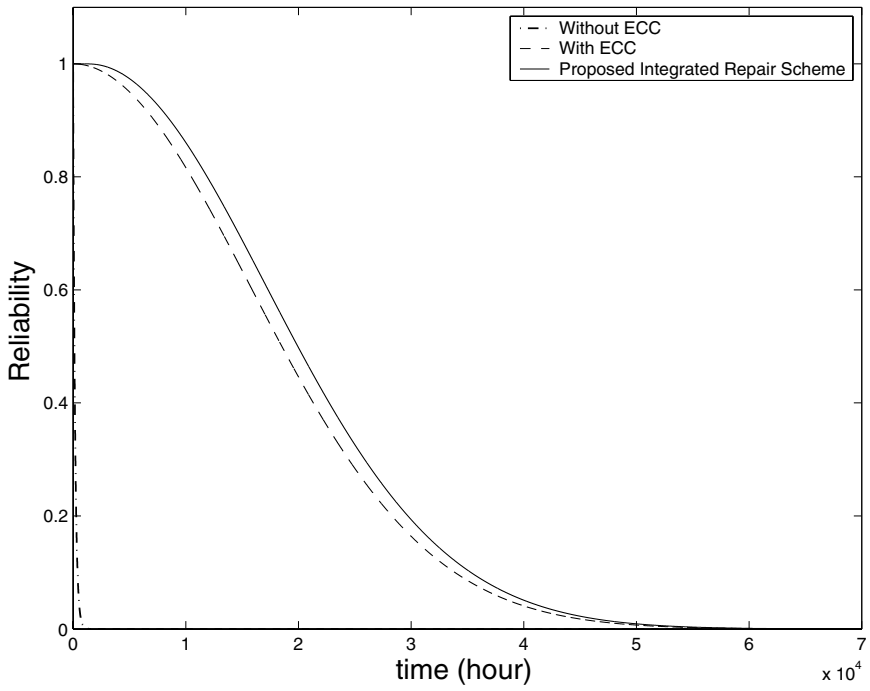

Figure 5: Reliability of the $32K \times 64$ memory with ECC (dashed line), without ECC (dot-dashed line), and that using the proposed method (solid line).

We use an $32K \times 64$ memory with Hamming code to describe the reliability enhancement of our proposed scheme. The memory contains 32K words and 64-bit I/O. We assume the memory module has a constant hard error rate, $\lambda_b = 10^{-8}$ per hour [11], and it contains 8 redundant rows and 4 redundant columns. With the constant error rate, we assume that all hard errors and soft errors are single bit solid errors during on-line operation. Also, during the Error Identification phase, no other errors may occur. From the redundancy simulation result, the probabilities that the unused redundancy $r + c$ = 0 to 12 are 0.01%, 0.22%, 1.1%, 1.55%, 2.65%, 7.3%, 8.63%, 12.83%, 14.6%, 17.9%, 17.9%, 15.27% and 0.04%, respectively, after the production test and repair. The results of the reliability figures are plotted in Fig. 5. From the figure, we can see that the reliability is improved with the integrated ECC and redundancy repair scheme. The mean time to failure (MTTF) is a measure for estimating the average lifetime of a chip. The MTTF is obtained by integrating the reliability function with respect to t:

$$MTTF = \int_0^\infty R'(t)dt = \int_0^\infty \sum_{\substack{0 \leqslant x \leqslant rr \\ 0 \leqslant y \leqslant rc}} P_{x,y} \cdot R_{x,y}(t)dt = \sum_{\substack{0 \leqslant x \leqslant rr \\ 0 \leqslant y \leqslant rc}} \int_0^\infty P_{x,y} \cdot R_{x,y}(t)dt \tag{4}$$

From the example discussed above, the MTTF of the $32K \times 64$ memory with ECC is 19,758 hours, and the MTTF

Proceedings of the 2005 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)
0-7695-2464-8/05 $20.00 © 2005 IEEE

of the same memory with our integrated ECC and repair scheme is 21,170 hours. Therefore, the improvement of MTTF is 1,412 hours (7.1%).

6 Experimental Results

We have implemented the integrated ECC and redundancy repair scheme with a 0.25 μm CMOS technology. We have designed the ECC Controller for a series of ECC SRAMs, which contain single error correction code (Hamming code) and redundancy memory elements, with different memory sizes and word lengths, as listed in Table 1. For example, a memory of size $32K \times 32$ has 32K words and 32-bit I/O. The number of parity bits is $log_2 32 + 1 = 6$. In the table, the ECC column shows the total gate count of the ECC Controller and OR gates. The BIST, and RC columns represent the gate counts of the BIST and RC circuits, respectively. The Total column shows the total area overhead of the memory test wrapper for the corresponding memory core, derived from (5).

$$\text{Total (\%)} = \frac{\text{ECC}}{\text{Memory Area} + \text{BIST} + \text{RC}} \tag{5}$$

Table 1: Hardware overhead comparison.

Memory size	ECC	BIST	RC	Total
$32K \times 32$	1,084	2,572	824	0.52%
$32K \times 64$	1,502	3,895	824	0.48%
$32K \times 128$	2,033	6,434	824	0.41%
$16K \times 64$	1,391	3,855	783	0.77%
$8K \times 128$	1,955	6,396	728	1.16%

In this experiment, the redundancy consists of eight spare rows and four spare columns in all cases. From Table 1, we can see that the area overhead of the memory test wrapper is low even for a small SRAM core. For a medium-sized memory, such as the 32K$\times$64 memory core, the overhead is below 0.5%. In general, the area of the test wrapper circuit increases at a log-scale rate with respect to the memory size. When we add parity bits to increase the capability of correcting errors, the area overhead will grow much faster than that using the proposed method. The overhead of even a single parity bit for the 32K$\times$64 memory is $1/(64+7) = 1.4\%$, not counting the encoder and decoder overhead. The proposed method thus is cost-effective. To reduce area overhead, the SEC scheme is more widely used. In table 2, we show the gate count, MTTF, and Cost comparison among different memories with and without SEC ECC. Again, the area overhead of our proposed method is low—only slightly higher than the original one. The MTTF (and reliability) of the memory is increased by a greater degree, so the Cost is reduced.

Table 2: Comparison among different memories with and without SEC ECC.

Memory size		Area	MTTF	Cost
$32K \times 32$	WECC	195,413	321.74	613.6
	SEC	220,862	37,147	5.96
	SECP	222,030	39,786	5.58
$16K \times 64$	WECC	171,522	344.36	498.1
	SEC	192,610	28,010	6.88
	SECP	194,153	30,834	6.30
$8K \times 128$	WECC	157,622	359.53	438.4
	SEC	176,159	20,681	8.52
	SECP	178,308	23,629	7.54

Proceedings of the 2005 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)
0-7695-2464-8/05 $20.00 © 2005 IEEE

7 Conclusions

We have presented an integrated ECC and redundancy repair scheme. During normal operation, if there are available unused redundant elements and a fault is identified as a hard fault in the Error Identification phase, we repair the faulty word by redundancy in the Hard Repair phase, effectively increasing the reliability of the memory chip. To reduce the area overhead, we can combine the ECC Controller and BIST circuit to reuse the address counter. By doing so, the additional OR gates are also excluded. We have also presented a reliability evaluation procedure. Experimental results show that the integrated scheme indeed improves the reliability of a memory chip, as it combines the benefits of redundancy repair and ECC. The approach also provides a cost-effective way to solve the problem of growing parametric defects in deep-submicron SOC products, and is good for other products demanding high reliability.

References

[1] Semiconductor Industry Association, "International technology roadmap for semiconductors (ITRS), 2003 edition", Dec. 2003.

[2] C.-T. Huang, J.-R. Huang, C.-F. Wu, C.-W. Wu, and T.-Y. Chang, "A programmable BIST core for embedded DRAM", *IEEE Design & Test of Computers*, vol. 16, no. 1, pp. 59–70, Jan.-Mar. 1999.

[3] I. Kim, Y. Zorian, G. Komoriya, H. Pham, F. P. Higgins, and J. L. Lweandowski, "Built in self repair for embedded high density SRAM", in *Proc. Int. Test Conf. (ITC)*, Oct. 1998, pp. 1112–1119.

[4] J.-F. Li, J.-C. Yeh, R.-F. Huang, C.-W. Wu, P.-Y. Tsai, A. Hsu, and E. Chow, "A built-in self-repair scheme for semiconductor memories with 2-D redundancy", in *Proc. Int. Test Conf. (ITC)*, Charlotte, Sept. 2003, pp. 393–402.

[5] Y. Zorian, "Embedded memory test & repair: infrastructure IP for SOC yield", in *Proc. Int. Test Conf. (ITC)*, Baltmore, Oct. 2002, pp. 340–349.

[6] C. V. Srinivasan, "Codes for error correction in high-speed memory systems—Part I: Correction of cell defects in integrated memories", *IEEE Trans. Computers*, vol. 20, pp. 882–888, 1971.

[7] T. Granlund, B. Granbom, and N. Olsson, "Soft error rate increase for new generations of SRAMs", *IEEE Trans. Nuclear Science*, vol. 50, no. 6, pp. 2065–2068, Dec. 2003.

[8] N. Derhacobian, V. A. Vardanian, and Y. Zorian, "Embedded memory reliability: The SER challenge", in *Proc. IEEE Int. Workshop on Memory Technology, Design and Testing (MTDT)*, San Jose, Aug. 2004, pp. 104–110.

[9] S. Gregori, A. Cabrini, O. Khouri, and G. Torelli, "On-chip error correcting techniques for new-generation flash memories", *Proc. of the IEEE*, vol. 91, no. 4, pp. 602–616, Apr. 2003.

[10] C. Wickman, D. G. Elliott, and B. F. Cockburn, "Cost model for large file memory DRAMs with ECC and bad block marking", in *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems (DFT)*, Albuquerque, Nov. 1999, pp. 319–327.

[11] M. Nicolaidis, N. Achouri, and L. Anghel, "A diversified memory built-in self-repair approach for nanotechnologies", in *Proc. IEEE VLSI Test Symp. (VTS)*, Napa Valley, Apr. 2004, pp. 313–318.

[12] Y.-T. Hsing, C.-W. Wang, C.-W. Wu, C.-T. Huang, and C.-W. Wu, "Failure factor based yield enhancement for SRAM designs", in *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems (DFT)*, Cannes, Oct. 2004, pp. 20–28.

[13] R.-F. Huang, J.-F. Li, J.-C. Yeh, and C.-W. Wu, "A simulator for evaluating redundancy analysis algorithms of repairable embedded memories", in *Proc. IEEE Int. Workshop on Memory Technology, Design and Testing (MTDT)*, Isle of Bendor, France, July 2002, pp. 68–73.

[14] R. C. Baumann, "Soft errors in advanced semiconductor devices—Part I: The three radiation sources", *IEEE Trans. Device and Materials Reliability*, vol. 1, no. 1, pp. 17–22, Mar. 2001.

[15] W. Kuo, W.-T. K. Chien, and T. Kim, *Reliability, Yield, and Stress Burn-in*, Kluwer Academic Publishers, Boston, 1998.

[16] C. Cheng, C.-T. Huang, J.-R. Huang, C.-W. Wu, C.-J. Wey, and M.-C. Tsai, "BRAINS: A BIST complier for embedded memories", in *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems (DFT)*, Yamanashi, Oct. 2000, pp. 299–307.