Low Power SoC Memory BIST

Yuejian Wu[1] and Andre Ivanov[2]
[1]Nortel, 3500 Carling Ave, Ottawa, ON., Canada, K2H 8E9
[2]Dept. of ECE, Univ. of British Columbia, Vancouver, BC., Canada, V6T 1Z4,
{yuejian@nortel.com, ivanov@ece.ubc.ca}

Abstract

With the ever increasing number of memories embedded in a System-on-Chip (SoC), power dissipation due to test has become a serious concern. This paper studies power dissipation in SRAMs and proposes a novel low power memory BIST. Its effectiveness is evaluated on memories in 130 and 90 nm technologies. As demonstrated, up to 30% power reduction can be achieved with virtually zero hardware overhead.

1. Introduction

With the ever increasing memory requirements in modern systems, a large number of memories are being embedded in System on Chip (SoC). These memories occupy a significant percentage of silicon real-estate in a SoC. In 2001, memories in SoCs have already accounted for over 60% of the silicon area [1]. The percentage is fast growing and is expected to rise to 94% by 2016 [9, 11]. Memory test in such an environment faces many challenges. One such challenge is long test time. An effective test time reduction technique is parallel test whereby multiple memories are tested simultaneously with dedicated or shared memory Built-In Self-Test (BIST) controllers.

Like in logic test, memory power dissipation during test is often much higher than that in mission mode operations due to higher signal activities [2, 4, 5]. When a large number of memories are tested simultaneously, the total test power dissipation can exceed power constraints, thus generating excessive heat and potentially causing device damage, yield loss and reliability issues. A straightforward solution to the power problem is to reduce the number of memories tested in parallel so that the power constraint be respected. However, such solutions impose more test time. Much research has been conducted on power constrained test scheduling [3, 6, 14]. The goal of such scheduling is to maximize the parallelism of memory and other IP core testing without exceeding power limits. The effectiveness of these solutions is limited by the power dissipation of each and every memory and other IP cores during test.

In order to reduce memory power dissipation during test, proposals for low power memory BIST have been reported [4, 5]. These proposals all use a gray code address generator instead of binary code in order to minimize signal activities in memory address decoders. Except for very small memories, however, power dissipation in decoders is often a negligibly small portion of total memory power [7-9]. In comparison, power dissipation on the memory data path constitutes significant percentage of total memory power [7-9]. This paper studies the power dissipation on memory data paths and proposes a novel solution that effectively reduces the total test power dissipation when multiple memories are tested in parallel.

2. SRAM Power Dissipation

Memory power dissipation consists of three major components: power dissipations in the memory array, in the decoders (row and column) and in its peripheral circuits (sense amplifiers and write drivers) [8]. Among the three, power dissipation in the memory

0-7695-2706-X/06 $20.00 © 2006 IEEE

array is the most significant due to its large capacitance and high signal activities. Figure 1 shows a single column of a 6T SRAM along with its bit lines (BL and BLb) and word lines (WL).

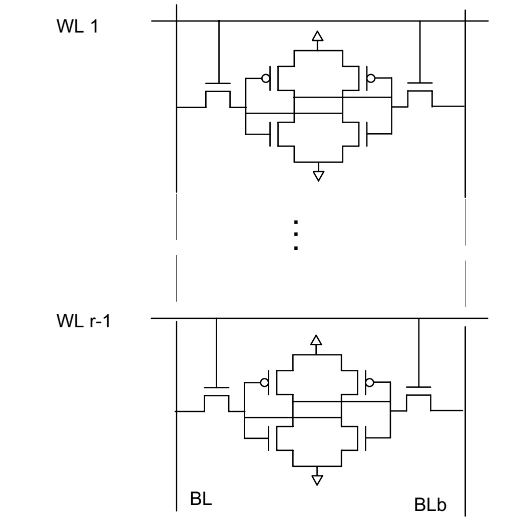


Figure 1. a single column of memory cells.

As shown in Figure 1, a large number of memory cells share a pair of differential bit lines (BL and BLb). Depending on the specific memory configuration, the number of memory cells on the same bit lines can be as high as 512 or even 1024 [7, 8, 10, 12], thus resulting in significant total capacitance given the load of the cells and the wires.

Memory read and write operations both begin with pre-charged high bit lines. During a read cycle, row decoder asserts the word line (WL) of a selected row, i.e., asserting WL = 1. Once selected, all memory cells on the same WL try to discharge their corresponding bit lines, BL or BLb. In practice, the number of memory cells on a single WL can also be fairly large, e.g., 512 [7, 10, 12]. Even though all the cells on the same WL are activated, only those bit lines that are selected by the column decoder are used to generate read data. In order to speed up read operations and to reduce power on the bit lines, the duration of WL = 1 is often fairly small, during which the bit line discharge due to read is often <10% of Vdd. After a read operation is completed, all the bit lines, BL and BLb, must be re-charged back to Vdd to be ready for the next memory operation. The current drawn during a read cycle is [8]:

$$Idda(r) = [m * Idc(r) * Dt + m * Cd * DV(r)]*f, \quad (1)$$

where m represents the number of cells on the same WL; Idc(r) is the DC current on the bit lines during read; Dt is the duration of WL=1; Cd the capacitance of a bit line, BL or Blb; DV(r) denotes the bit line voltage swing during a read; and f is an operating frequency or 1/Tc with Tc being system clock cycle time. The first term in Equation 1 represents DC current draw when WL = 1 and the second term is the current draw from Vdd for pre-charging the bit lines after a read is performed.

During a write cycle, the row decoder asserts WL=1 to select the row to write. Similarly to a read operation, once a row is selected, all the memory cells on the row are activated [7, 8, 12]. However, except for the memory cells to write, all the other cells selected are essentially performing pseudo read operations, whereby data read from these cells are ignored [8, 12]. For the cells to write, write drivers pull down their bit lines BL or BLb. For example, to write a 0 into a cell, a write driver pulls down its BL while leaving its BLb floating high. On the other hand, to write a 1, the write driver drives its BLb low while leaving its BL floating high.

In order to reliably write memory cells at high speed, the voltage swing of the bit lines during a write is usually Vdd. As a result, a write operation dissipates significantly more power as compared to a read operation. The current draw on the bit lines during a write is:

$$Idda(w) = \{p*[Idc(w) * Dt + Cd * DV(w)] + (m-p)*[Idc(r) * Dt + Cd * DV(r)]\}*f, \quad (2)$$

In Equation (2), p is the number of memory cells to write (otherwise known as memory word width); Idc(w) is the DC current during write; Dt is the duration of WL = 1; DV(w) the voltage swing on the bit line in a write cycle. The term (m-p)*[Idc(r)*Dt + Cd*DV(r)]*f is due to the fact that all the memory cells on the same row but not participating in the write are essentially performing read operations.

From Equation (2), the two parameters that have most impact on write power dissipation are p and DV(w). Assuming bit line voltage swing during write to be Vdd and bit line voltage swing during read to be 10% of Vdd, a write operation draws about 10 times more current than a read operation, for the same word width *p*.

For a given DV(w), the larger the *p* the higher the power dissipation. As shown in [7], when p=8, 28% of the power on the data path is due to bit line charge after each write; when p=256, this percentage rises to 90%. As SoCs become more complex towards higher performance, embedded SRAMs tend to have wider word width, *p*. It is common to see *p* = 32 ~ 512 [7, 12] and in some cases, p = 640 [10] or much higher.

This paper explores the power dissipation differences between read and write operations and proposes a novel low power memory BIST solution for SoCs.

3. Traditional BIST for Multiple Memories

Memory test algorithms are all made of a sequence of read and write operations. For example, Figure 2 shows the March C- algorithm [13].

$$\begin{array}{cccccc} \{ \updownarrow(w0) & \uparrow(r0, w1) & \uparrow(r1, w0) & \downarrow(r0, w1) & \downarrow(r1, w0) & \updownarrow(r0) \} \\ M0 & M1 & M2 & M3 & M4 & M5 \end{array}$$

Figure 2. March C- structure

In a SoC, where there are many memories to test, multiple memories are often tested in parallel in order to minimize test time. In many cases, the test algorithms used for these memories are the same or very similar, especially when the memories being tested are of the same type and configurations. As a result, when a memory is performing a read operation, all other memories are performing read operations at the same time. Similarly, when a memory is conducting a write operation, all other memories are conducting write operations as well.

Figure 3 illustrates the application of March element M1 on two memories being tested in parallel, where $(r0, w1)_i$ represents the test (r0, w1) is applied to address *i*. As shown, the read operations applied to both RAM 1 and RAM 2 are aligned in time and so are the writes.

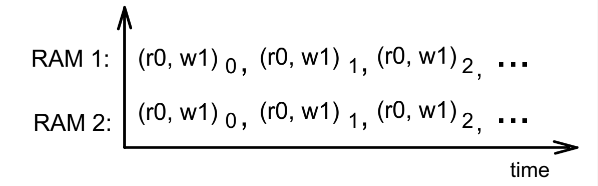

RAM 1:
(r0, w1) 0, (r0, w1) 1, (r0, w1) 2, ...
RAM 2:
(r0, w1) 0, (r0, w1) 1, (r0, w1) 2, ...
time

Figure 3. March element M1 on two memories

As shown Figure 3, all the memories tested in parallel always perform writes simultaneously. Consequently, when one memory is in its peak power dissipation state, all other memories being tested are also in their peak power dissipation states. Under a given power constraints, the limiting factor is the sum of peak power dissipation from all the memories. This is illustrated in the following example.

Example 1: Two identical memories are to be tested, each has a write power dissipation P(w) = 100 mW and a read power dissipation P(r) = 40 mW. If the two memories are tested in parallel in the traditional way, the peak power dissipation during the test would be P = 2*P(w) = 200mW. If a power budget of only 150 mW is allowed, the two memories cannot be tested at the same time as P exceeds the power constraint. They must be tested one after another, thus increasing the total test time by a factor of 2.

4. Low Power Memory BIST

4.1 Basic Idea

In Example 1, the two memories cannot be tested simultaneously because their peak power dissipation states coincide in time causing the total peak power to exceed the power budget. If we "skew" read and write operations being applied to the two memories, i.e., if we ensure that one memory is performing a less power demanding read operation when the other memory is conducting a write operation or vice versa, the total peak power dissipated at any given time will be no more than P(w)+P(r), which is significantly less than 2P(w). Consequently, the total peak power dissipated during parallel test of the two memories may be kept under the power constraint and thus resulting in test time reduction. This is illustrated in Example 2.

Example 2: Assume the same two memories and power budget as in Example 1. If a test is designed such that the two memories never conduct write operations simultaneously, the peak power when testing the two memories in parallel becomes P(w) + P(r) = 100 mW + 40 mW = 140 mW, well under the power constraint of 150 mW. As a result, the two memories can now be tested in parallel, thus saving total test time.

The basic idea of the proposed low power memory BIST is to properly arrange read and write operations amongst multiple memories tested in parallel such that the total peak power is less than the sum of peak powers.

4.2 Low Power BIST Realization

For simplification, let us begin with testing two identical memories in parallel using the March C- test algorithm outlined in Figure 2. The results of the discussion, however, can be extended to more general cases.

The March C- algorithm shown in Figure 2 consists of three types of March elements. The type 1 March element is M0, which conducts consecutive write operations for memory initialization; the type 2 March elements consist of M1~M4, which can be described as (r*d*, w*db*) with *d* being any binary number and *db* being its complementary; the type 3 March element is M5, which performs consecutive read operations over the entire memory space.

Apparently, for the type 3 March element M5, no modification is needed when it is simultaneously applied to both memories. In this case, the total peak power = 2*P(r) < 2*P(w) because both memories conduct only read operations.

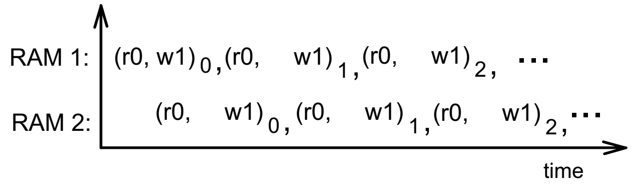

RAM 1:
$(r0, w1)_0, (r0, w1)_1, (r0, w1)_2, \cdots$
RAM 2:
$(r0, w1)_0, (r0, w1)_1, (r0, w1)_2, \cdots$
time

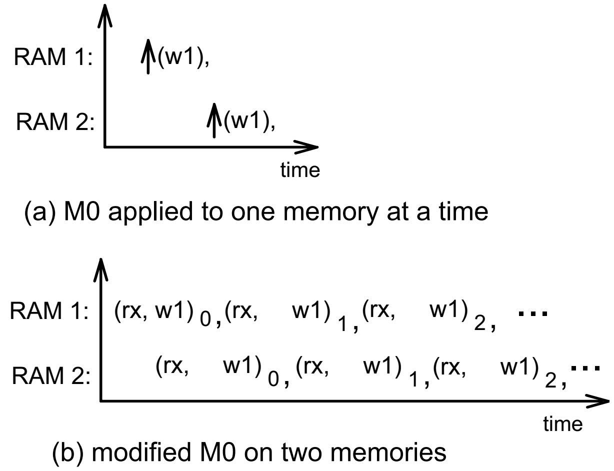
Figure 4. low power March element M1 on two memories

For the type 2 March element M2 -- M4, no modification to the algorithm itself is needed. However, we must skew the application of the read and write operations to both memories so that no simultaneous write occurs. Figure 4 illustrates the skew of the read and write operations when applying M1 to the two memories.

As shown in Figure 4, when one memory undergoes a write operation, the other memory is always performing a read operation, thus resulting in the total peak power dissipation P(w) + P(r) < 2*P(w). As shown in Figure 4, the test application time for M1 is 2*n*+1, where *n* is the number of addresses in each memory. Compared to the conventional test application, where the test application time for M1 is 2*n*, an overhead of 1 additional cycle is required.

Similarly, the same scheme as shown in Figure 4 can be used for March elements M2 -- M4. When applying M1 -- M4 sequentially, the additional cycle required at the end of M1 due to the skewed read and write can be overlapped by the first cycle of M2. As a result, the time for applying M1 -- M4 is 2*n* * 4 + 1, where 2*n* is the complexity of each March element M1~M4; the factor of 4 is due to the application of 4 March elements M1 -- M4; and 1 is the additional clock cycle required due to skewed read and write operations. Therefore, the time overhead of the proposed scheme for applying M1 -- M4 is 1 additional clock cycle in total.

As for the type 1 March element M0, there exist at least two ways to apply it to both memories. The first is to apply M0 to one memory at a time. In other words, apply M0 to memory 1 while placing memory 2 in idle. Once the first memory is initialized, M0 is then applied to memory 2 while placing memory 1 in idle. After both memories are initialized with M0, the application of M1 -- M5 is conducted as described Figure 4. The test time overhead in this case is *n* cycles. This is illustrated in Figure 5 (a).


RAM 1:
⇑(w1),
RAM 2:
⇑(w1),
time
RAM 1:
(rx, w1) 0 ,(rx, w1) 1 ,(rx, w1) 2 , ...
RAM 2:
(rx, w1) 0 ,(rx, w1) 1 ,(rx, w1) 2 ,...
time

(a) M0 applied to one memory at a time

(b) modified M0 on two memories

Figure 5. low power March element M0

Alternatively, as shown in Figure 5(b), M0 can be modified to (rx, w0), where rx is a read operation with read data ignored or so-called read don't care. With such a modification, M0 can be applied using the same scheme as that shown Figure 4. The test time overhead in this case is also *n* clock cycles since the number of memory operations in the modified M0 is doubled even though it can be applied to both memories in parallel using the scheme shown in Figure 4 or Figure 5(b). The advantage of the modified M0 is its consistency to the other March elements M1~M4, which leads to simplified BIST controller implementation.

To summarize the two memory scenarios, we are able to reduce test peak power from 2*P(w) down to P(w) + P(r) at the cost of additional *n*+1 clock cycles. Considering the test time for the original March C- to be 10*n*, the cost of the additional test time to power saving is ~10%.

The contribution of the peak power reduction to test time minimization depends on the ratio P(r) / P(w) as well as power constraints.

Example 3: Assume the same two memories and power budget as in Example 1. With the traditional BIST approach, the two memories must be tested sequentially, thus requiring 2*10*n* = 20*n* clock cycles. With the proposed low power BIST, the two memories can be tested in parallel, requiring only 10*n* + (*n*+1) = 11*n* + 1 clock cycles. The test time reduction due the reduced peak power is about (20*n* - 11*n*) / 20*n* = 45%.

However, for some other power budgets or P(r) / P(w) ratios, the benefits resulting from the low power BIST may not be obvious. For example, if the power budget used in Example 3 is reduced from 150 mW to 120 mW, peak power reduction to P(w) + P(r) = 140 mW is not good enough to test both memories in parallel. In this case, no direct time saving is possible if only two memories exist in a SoC.

Furthermore, if the P(r) / P(w) is greater than 40 / 100 as used in Example 3, the proposed scheme does not result in direct time saving if only two memories exist. For example, if P(w) = 100 mW but P(r) = 60 mW, P(w) + P(r) = 160 mW, which is still greater than the power budget of 150 mW. Therefore, the two memories cannot be tested in parallel even with the low power BIST.

In a SoC with hundreds or even thousands of embedded memories and other IP cores, any peak power reduction for the memories provides new opportunities for test time optimization using power-constrained test scheduling algorithms such as those described in [3, 6, 14]. Consequently, a shorter overall test time for the entire SoC can be achieved. For example, if there are 800 memories embedded in a SoC and 200 of them can be tested in parallel due to power constraint. If each memory has P(w) = 100 mW and P(r) = 60 mW, [P(w)+P(r)] / [2*P(w)] = 20% peak power reduction can be achieved for each memory, which allows another 20% or 40 more memories to be tested in parallel under the same power constraint. In this case, shorter test time for the entire SoC become possible with the low power BIST.

4.3 Cost Analysis

There are two types of costs in implementing the proposed low power BIST. One is the additional test time required to avoid simultaneous write operations during memory initialization stage and the other is hardware costs.

4.3.1 Test Time Costs: The test time cost of the proposed scheme depends heavily on the test algorithms used. For example, when applying the low power BIST to the March C-, as described in the previous section, it requires about 10% additional test time to avoid simultaneous write during March element M0. If a more complex test algorithm for more realistic faults is used here, the test time cost in percentage would decrease. For example, when applying the low power BIST idea to the March RAW test algorithm [15], whose complexity is 26*n*, the test time cost of the proposed scheme is only about 1*n* / 26*n* = 3.8%.

Furthermore, most memory tests in practice often include a couple of ~100ms delay or wait periods for the detection of data retention faults. In this case, the test time cost of the proposed scheme becomes negligible.

4.3.2 Hardware Costs: The hardware costs of the low power BIST vary depending on the particular implementations. Here, we present a scenario in which the proposed scheme can be implemented with zero hardware cost.

Assume *N* identical memories to be tested using two BIST controllers, each controlling the test of *N/2* memories in parallel. We also assume that ↕ (w0) in the test algorithm has been replaced with ↕ (rx, w0) as described in Figure 5(b). To implement the low power BIST in this case, we only need to enable one BIST controller one clock cycle before enabling the other BIST controller so that the memory write operations between the two BIST controllers are always skewed by one clock cycle. In this case, the *N/2* memories that share a BIST controller can have simultaneous write, incurring a peak power of *(N/2)**P(w). However, since the operations of the two BIST controllers are always off or skewed by one clock cycle, when one BIST controller performs write, the other BIST controller always performs a read or vice versa. Therefore, the peak power at any given time becomes *(N/2)**[P(w) + P(r)] as opposed to *N**P(w).

As far as hardware costs are concerned in this case, it is essentially zero since the proposed scheme is implemented simply by skewing the starting time of the two BIST controllers.

5. Evaluations

To evaluate the effectiveness of the low power BIST, read and write power numbers from a few embedded SRAMs in 130nm and 90nm are collected in Tables 1 and 2 respectively.

The first columns in Table 1 and Table 2 report the memory configurations; the second column in the tables show the ratio of P(r) / P(w); and the peak power reduction shown in the third columns of the two tables report a figure of merit defined as {2P(w) – [P(w) + P(r)]} / 2P(w) = ½ [1 + P(r) / P(w)].

Table 1. Peak power reduction (130nm).

Memory configuration	P(r) / P(w)	Peak Power Reduction
388 x 76	69.23%	15.39%
436 x 76	68.86%	15.57%
432 x 8	71.67%	14.17%
48 x 52	83.63%	8.18%
512 x 20	78.95%	10.52%
1K x 20	80.94%	9.53%
16K x 8	77.07%	11.47%
16K x 8	92.46%	3.77%

Table 2. Peak power reduction (90nm).

Memory configuration	P(r) / P(w)	Peak Power Reduction
16K x 8	57.41%	21.30%
1K x 128	38.16%	30.92%
2K x 16	71.78%	14.11%
4K x 64	41.33%	29.34%
16K x 64	35.78%	32.11%
8K x 16	45.43%	27.29%
128 x 64	70.00%	15.00%
128 x 4	80.63%	9.69%

As shown in the tables, the effectiveness of the proposed solution ranges from 3.77% to 32.11%, with an average of 16.77%. Due to the lack of information on memory internal architecture, the exact reason for the wide range in peak power reduction is unknown. We suspect that memories with less power reduction be optimized for high speed applications, in which long bit lines and word lines are segmented as described in [8]. As a result, fewer cells share the same bit lines. On the other hand, memories with higher power reduction must have been optimized for density in which more memory cells share the same bit lines.

6. Discussions

This section discusses some limitations of the proposed low power BIST scheme and its application to testing multiple port memories and memories of different sizes.

6.1 Testing Memories of Different Sizes

For easy presentation, previous sections assume the test of multiple identical memories. In fact, the proposed solution applies equally well to memories of different sizes.

Let us consider an example where the March element M1 is applied to two memories of different sizes. In the traditional way, the application of (r0,w1) to both memories aligns in time, i.e., both memories execute the r0 simultaneously and both execute the w1 simultaneously. Apparently, the smaller one of the two memories will finish M1 first and move on to the execution of M2 before the larger one finishes M1. In this case, the smaller memory executes M2 with (r1,w0) while the larger one continues to execute M1 with (r0,w1). Although the read and write data to both memories are different in this case, their read and write operations still align with each other. In other words, when the smaller memory performs a r1 operation the larger one conducts a r0 operation. Similarly, when the larger one executes a w1 operation the smaller one executes a w0 operation. Their read operations coincide in time and so do their write operations.

With the proposed low power solution, if we delay the test start time of the smaller memory by a clock cycle as compared to that of the larger one, both memories will never conduct simultaneous write operations, thus resulting in reduced peak test power dissipation.

6.2 Testing Multi-Port Memories

The low power BIST scheme is applicable only if there exist multiple memories in a SoC, which is often true in practice. Earlier discussions in this paper assume multiple single port memories. In fact, the same scheme can also be applied to multi-port memories.

In the case of dual-port memories, for example, each memory can be considered as a pair of single-port memories of equal address spaces [16]. Identical test can be simultaneously applied to both ports [17] in order to reduce test time. This is very similar to testing two identical single port memories as discussed earlier in this paper. In the case of a dual-port memory, the proposed low power BIST idea can be applied by skewing the read and write operations between the two ports of the same memory. In other words, when one port is performing a write operation, the other port conducts a read or vice versa. Consequently, the peak power dissipation during memory test is reduced from 2P(w) to P(w) + P(r).

6.3 Limitations

Some test algorithms contain (..., w*d*, w*db*, …). In this case, skewing the test to two memories by a single clock cycle as suggested in the previous sections does not directly apply here. In order to make it work with such algorithms, the March element with (…, w*d*, w*db*, …) can be modified to (…, w*d*, r*d*, w*db*, …). The cost of such modification is increased time overhead.

Recently, in order to reduce memory power consumption, a group of low power memory design techniques have been proposed [7]. These techniques aim at the reduction of bit line voltage swing during write operations, thus minimizing the voltage swing differences between read and write operations. As a result, the proposed scheme will become ineffective for such memories. Although these techniques are very effective in reducing memory power, they have several issues such as increased silicon overhead, memory speed degradation, read/write stability and reduced signal/noise ratio. These issues need be resolved before they become practical.

In summary, the proposed low power BIST takes advantage of the power dissipation differences between read and write operations. The greater the differences, the more effective the proposed scheme. In other words, the effectiveness depends heavily on the architecture of the memories themselves. When a memory is optimized for speed, it is likely to have limited number of cells sharing the same bit lines. Consequently, the difference between read and write power dissipation become small and the proposed scheme become less effective. On the other hand, when a memory is optimized for density, more memory cells are likely to share the same bit lines. In this case, the difference between read and write power dissipation tends to be large and makes the proposed scheme more effective. In addition, when a large number of memories are embedded in a SoC, most of these memories tend to have large word width, p. As shown in Equations 1 and 2, the larger the p, the larger the difference between read and write power dissipations, thus making the proposed solution more effective.

7. Conclusions

Due to the large voltage swing on bit lines during memory write operations, a memory write often consumes much higher power than a memory read. This is especially true if the bit lines are shared by many memory cells and each memory word consists of a large number of bits. Taking advantage of the power dissipation differences between memory read and write operations, this paper proposed a novel low power memory BIST technique. The implementation of the proposed low power BIST was also discussed. As shown, the proposed scheme can be implemented with virtually zero hardware overhead. However, it does require a slight test time overhead. The percentage of such overhead varies dependently on the adopted memory test algorithms. Two examples were discussed. If the March C- test is used, 10% test time overhead is required. However, only 3.8% test time overhead is required if the comprehensive March RAW test is adopted. The effectiveness of the proposed scheme was evaluated using a few exemplary industrial memories. Although the evaluation is limited by the lack of information on memory internal architecture, it did demonstrate the effectiveness of the proposed scheme on some memories. Further study on the correlation between the effectiveness of the proposed scheme and the memory architecture is needed. The proposed low power BIST aims at the power reduction on memory data path. It can be combined with other low power techniques aiming for decoder power reduction such as those in [4, 5] for further test power reduction.

Acknowledgement: The authors are grateful to Mr. Robert Gibbins for helpful discussions.

8. References

[1] R. Rajsuman, "Design and Test of Large Embedded Memories: An Overview", IEEE Design and Test of Computers, May 2001, pp. 16-27.

[2] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices", Proc. IEEE VLSI Test Symp. pp.4-9, 1993.

[3] B. Wang et al. "A Retention-Aware Test Power Model for Embedded SRAMs", ASP-DAC, pp. 1180-1183, 2005.

[4] H. Cheung and S.K. Gupta, "A BIST Methodology for Comprehensive Testing of RAM with Reduced Heat Dissipation", Int. Test Conf., pp.386-395, 1996.

[5] B.H. Fang and N. Nicolici, "Power-Constrained Embedded Memory BIST Architecture", IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT'03), 2003.

[6] R.M. Chou, K. Saluja, and V.D. Agrawal, "Scheduling Test for VLSI Systems under Power Constraints", IEEE Trans. on VLSI, pp. 175-184, 1997.

[7] K. Kanda, H. Sadaaki and T. Sakurai, "90% Write Power-Saving SRAM Using Sense-Amplifying Memory Cell", IEEE J. Solid-State Circuits, vol. 39, No. 6, June 2004.

[8] K. Itoh, K. Sasaki and Y. Nakagome, "Trends in Low-Power RAM Circuit Technologies", Proc. of IEEE, Vol. 83, No. 4, 524-543, April 1995.

[9] R. Saleh, et al., "Trends in Low Power Digital System-on-Chip Designs", Proc. Int. symp. on Quality Electronic Design (ISQED'02), pp. 373-378, 2002.

[10] R. Gibbins, et al. "Design and Test of a 9-Port SRAM for a 10Gb/s STS1 Swtich", IEEE Int. Workshop on Memory Tech., Design and Testing, pp. 83-87, 2002.

[11] Semiconductor Industrial Association (SIA), "International Techonology Roadmap for Semiconductors (ITRS)", 2003

[12] L. Dilillo, et al., "Data Retention Fault in SRAM Memories: Analysis and Detection Procedures", Proc. IEEE VLST Test Symp., 2005.

[13] A.J. van de Goor, Testing Semiconductor Memories, Theory and Practice, COMTEX Publishing, Gouda, Netherlands, 1998.

[14] V. Iyengar and K. Charkrabarty, "System-on-a-Chip Test Scheduling with Precedence, Relationships, Preemption, and Power Constraints", IEEE Trans. CAD., vol. 21, No. 9, 2002, pp. 1088-1094.

[15] S. Hamdioui, et al., "Testing Static and Dynamic Faults in Random Access Memories", Proc. VLSI Test Symp., 2002, pp. 395-400.

[16] Y. Wu and S. Gupta, "Built-In Self-Test for Multi-Port RAMs", Proc. Asian Test Symp., 1997.

[17] F. Karimi and F. Lombardi, "Parallel Testing of Multi-Port Static Random Access Memories for BIST", Proc. IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT01), 2001.