Hindawi Publishing Corporation
VLSI Design
Volume 2007, Article ID 67019, 6 pages
doi:10.1155/2007/67019

Research Article

Low-Power Built-In Self-Test Techniques for Embedded SRAMs

Shyue-Kung Lu, Yuang-Cheng Hsiao, Chia-Hsiu Liu, and Chun-Lin Yang

Department of Electronic Engineering, Fu Jen Catholic University, Taipei County 24205, Taiwan

Received 30 January 2007; Accepted 5 September 2007

Recommended by Bernard Courtois

The severity of power consumption during parallel BIST of embedded memory cores is growing significantly. In order to alleviate this problem, a *row bank-based* precharge technique based on the divided wordline (DWL) architecture is proposed for low-power testing of embedded SRAMs. The memory cell array is first divided into *row banks*. The effectiveness of the row bank-based precharge technique is due to the predictable address sequence during test. In low-power test mode, instead of precharging the entire memory array, only the current accessed row bank is precharged. This will result in significant power saving for the precharge circuitry. The precharge power can be reduced to $1/b$ of that of the traditional precharge techniques, where b denotes the number of row banks in the memory array. With simple transmission gates and inverters, the modified precharge control circuitry was also designed. The hardware overhead for implementing the low-power technique is almost negligible. Moreover, the corresponding BIST design to implement the low-power technique is almost the same as the conventional BIST designs. It is also notable that the inherent low-power characteristics of the DWL architecture can be preserved. According to experimental results, 48.9% power reduction can be achieved for a 256×256 bit-oriented SRAM. The memory is divided into 8 row banks. Moreover, if the number of row banks increases, the power saving will also increase.

1. INTRODUCTION

VLSI technology keeps greatly increasing the degree of circuit integration in recent years. With this trend, high-density and high-capacity embedded memories are often implemented in a system chip. According to the Semiconductor Industry Association (SIA, Calif, USA) and ITRS 2003, the relative silicon area occupied by embedded memories will approach 94% by 2014 [1]. For example, the Compaq Alpha EV7 chip employs 135 million transistors for RAM cores alone, while the entire chip has 152 million transistors. However, the negative by-product of this highly integration is the high power dissipation. Therefore, many low-power design techniques have been proposed for embedded memories [2–5]. Moreover, due to the large area occupied by embedded memories, they will dominate the yield of the system chips. There are also many techniques proposed to increase the reliability and yield of system chips.

In order to assure the quality of system chips, BIST techniques are usually used to test embedded memory cores [6–8]. To speed up the test process and reduce the BIST area overhead, BIST sharing techniques [9] are widely adopted. That is, many memory cores will share the same BIST controller. However, parallel testing of memory cores will cause power consumption that far exceeds that during normal operation mode. This excessive power consumption will in turn damage the circuit. One solution to cure this dilemma is to reduce the number of memory cores which can be concurrently tested (BIST scheduling under power constraints). However, this approach will increase the overall test time of the system chips. Another promising solution is the development of memory BIST techniques which will achieve high fault coverage and also minimize the test power.

There are numerous papers proposed [10, 11] on constraining BIST power consumption for logic cores. However, there are only few papers which address the low-power BIST issue for embedded memories. In [4], an architecture-level low-power solution for RAMs was proposed. This methodology provides a systematic tradeoff between power consumption and area. Moreover, it also allows tradeoff between test time and test power consumption. In [12], the precharge activity can be reduced through the predictable addressing sequence. A BIST methodology for comprehensive testing of RAM with reduced power consumption was proposed in [13] where the power models of the memory system were analyzed.

In this paper, a *row bank-based* precharge technique for low-power testing of embedded SRAMs is proposed. This technique is based on the fact that the address sequence during test is predictable. Therefore, it is not necessary to precharge the entire memory array. We first divide the memory array into *row banks* based on the divided word-line (DWL) technique [2]. In low-power test mode, instead of precharging the entire memory array, only the precharge circuits in the current accessed row bank is activated. This will result in significant precharge power saving. The precharge power can be reduced to $1/b$ of that of the traditional precharge approaches, where b denotes the number of row banks in the memory array. The modified precharge control circuits are also designed. The hardware overhead to implement the low-power technique is almost negligible. Moreover, the BIST design to implement the low-power technique is almost the same as the conventional BIST designs. It is also notable that the inherent low-power characteristics of the DWL architecture can be remained.

According to simulated results, if the memory is divided into 8 row banks, 48.9% power reduction can be achieved for a 256×256 bit-oriented SRAM. Moreover, if the number of row banks increases, the power saving will also increase.

2. REVIEW OF DWL ARCHITECTURE

The simplified organization of an $m \times n$ memory chip is shown in Figure 1. Activating one of the m word lines is performed by the row decoder. The column decoder selects the column whose $\log_2 n$-bit address is applied to the column decoder input. The intersection of a word line and a column denotes a memory cell. The memory array contains m word lines ($W_0 - W_{m-1}$), n bit lines, and bit line bar ($BL_0 - BL_{n-1}BLB_0 - BLB_{n-1}$) as shown in Figure 2. The precharge control signals are denotes as PR_j, $0 \leq j \leq n-1$. The precharge signals are used to control the precharge circuitry denoted as PRE in this figure. In general, the precharge control signals for the unselected columns will be activated for the entire clock cycle. Alternately, precharge signal for the selected column will only be activated for 1/2 clock cycle.

The basic concept of divided word-line (DWL) technique for RAMs was first proposed in 1983 [2]. The DWL technique eliminates the load of the word lines in unselected row blocks and reduces the selected word-line delay appropriately. Due to the power down technique for unused memory cells, its main advantages include lower power consumption and faster access time. The structure of DWL technique is shown in Figure 3. This approach has been widely applied in a variety of commercial memory chips.

From Figure 3, the main scenario of the DWL technique is that each row of the memory cell array is divided into b *row blocks* by the word-line segments. If the memory has m rows ($W_0 - W_{m-1}$) and n columns ($C_0 - C_{n-1}$), n/b columns are included in each *row bank* (RB_i, $0 \leq i \leq b-1$). The local word line in each row block is activated by *switching gates*, which have two inputs–the *row select line* and the *row bank select line* (RBS_i, $0 \leq i \leq b-1$). Although the DWL technique possesses several inherent advantages, the divided structure has not been used for low-power BIST applications.

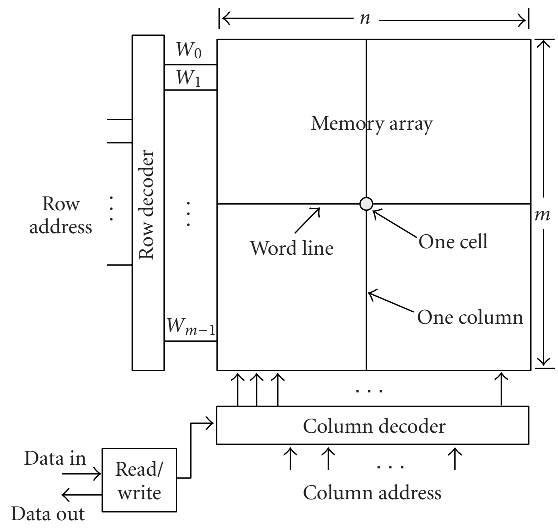

FIGURE 1: Simplified organization of an $m \times n$ memory.

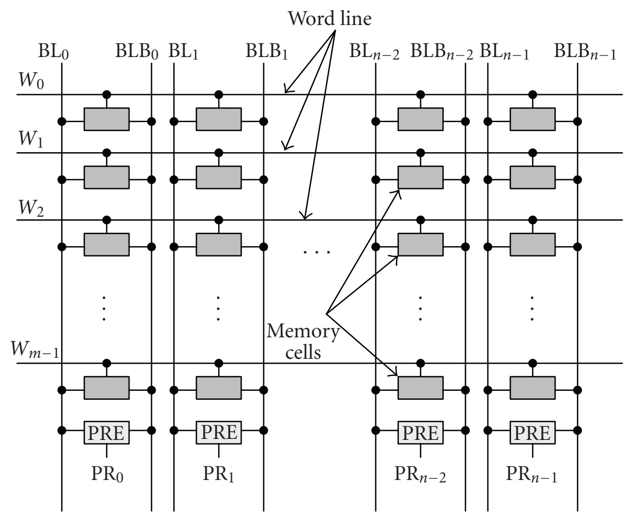

FIGURE 2: The $m \times n$ memory cell array with precharge circuitry.

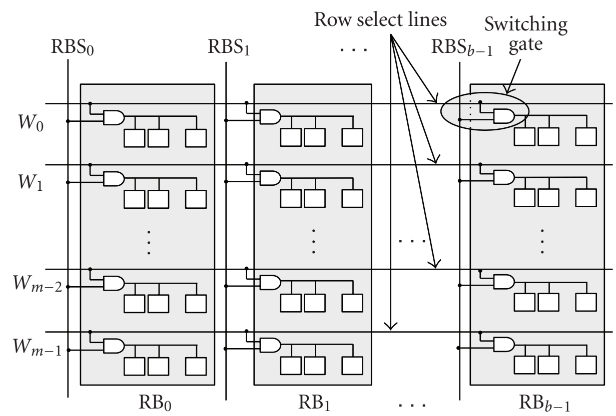

FIGURE 3: The divided word-line architecture.

This paper is the first attempt trying to use this divided architecture for low-power BIST designs. The row bank select signals are usually generated by a *row bank decoder* (to be described in Section 4). According to our technique, the row bank select signals can also be used to control the precharge activity during test mode. In other words, instead of the original precharge signals (PR_j), the RBS_j's can also be used to activate the precharge circuitry.

3. REDUCING PRECHARGE ACTIVITY BASED ON DWL ARCHITECTURE

There are many memory test algorithms, for example, March algorithms, which are widely used for memory testing [14]. Any address sequence can be identified as a ↑ sequence or a ↓ sequence as long as all addressed addresses occur exactly once. Therefore, the address sequence is *predictable*. There are few papers which use the address prediction feature for low-power BIST of embedded memories except [12]. However, the DWL technique has not been used in company with the address prediction feature. Therefore, the inherent low-power merits of DWL architecture cannot be preserved. Moreover, if the DWL technique is also adopted, it will result in simpler control circuitry.

In this paper, the DWL architecture and the address prediction characteristic are both adopted for the low-power BIST designs. For this purpose, the address sequence is selected as "word line after word line" [12] as shown in Figure 4. From this figure we can see that consecutive n/b memory cells within the same row bank are accessed and then the next row bank will be accessed. Therefore, for the current unselected row banks, we can deactivate their precharge signals. The modified memory array is shown in Figure 5. Instead of the original precharge signals (PR_0 − PR_{n-1}), the row bank precharge ($RBPR_j$) signals are added, $0 \leq j \leq b-1$. In normal operation mode, the original precharge signals control the precharge activity of all bit lines. In low-power test mode, the row bank precharge signals take over to control the precharge activity. It should be noted that only one of the row bank precharge signals will be activated. Therefore, instead of precharging the entire memory array, the bit lines within a single row bank will be precharged in test mode. Therefore, the precharge power can be saved significantly.

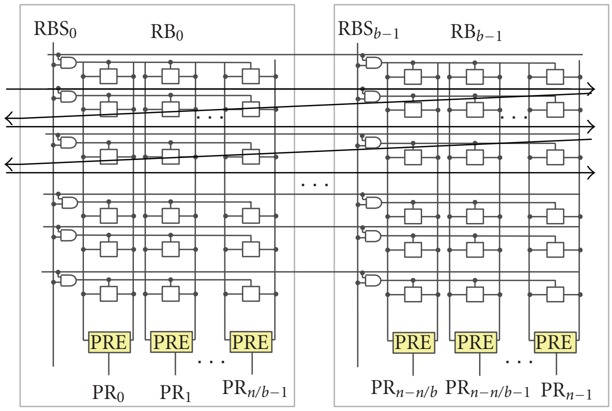

Figure 4: The "word line after word line" access sequence.

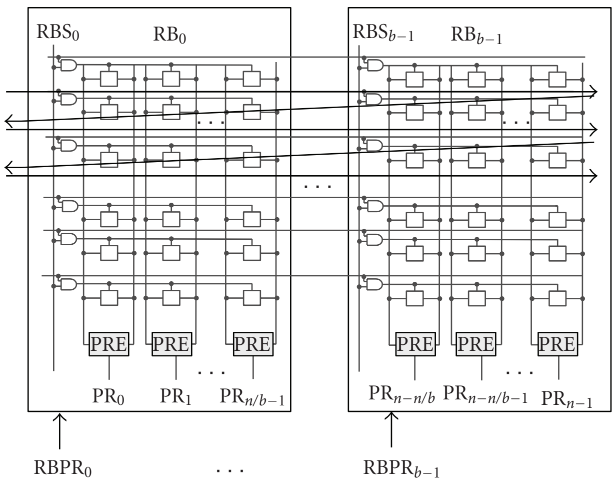

Figure 5: The modified memory array.

4. DESIGN OF THE CONTROL CIRCUITRY

Based on the proposed bank-based precharge methodology, the memory array contains two different operation modes: the *normal* mode and the low-power *test* mode. In the normal mode, the memory cells are accessed normally. In the low-power test mode, the precharge activity is confined to a single row bank for each clock cycle. However, when the last column of a row bank is accessed, we should also precharge the next row bank in order to restore the voltage level of all bit lines in the next row bank to V_{DD}. This operation is required to assure the correct access of the next row bank.

A practical control circuitry design for the bank-based precharge approach is shown in Figure 6. This control circuitry allows switching between the normal mode and the low-power test mode (controlled by the *mode* signal). In this figure, two transmission gates (acting together as a multiplexer) are added to the precharge circuitry of each column. When $mode = 0$ (normal mode), the original precharge signal PR_j is connected to the precharge circuitry (the block marked PRE).

When $mode = 1$ (low-power test mode), the left transmission gate is turned ON. Then, the output signal of the OR gate is used to control the precharge circuitry. The inputs of the OR gate contain two signals. One is the row bank select signal (RBS_j), $0 \leq j \leq b-1$, which is generated by the row bank decoder (to be discussed later). The other input signal is the column select signal CS_k, where k denotes the column index of the last column of the previous row bank. As described in Section 3, when the last column of the current accessed row bank is accessed, it should activate the precharge circuitry of the next row bank. Therefore, the column select signal CS_k can be used to perform this operation.

To generate the row bank select signals, a row bank decoder (RB) is required as shown in Figure 7. Since the memory array is divided into b row banks, $\log_2 b$ bits of the column

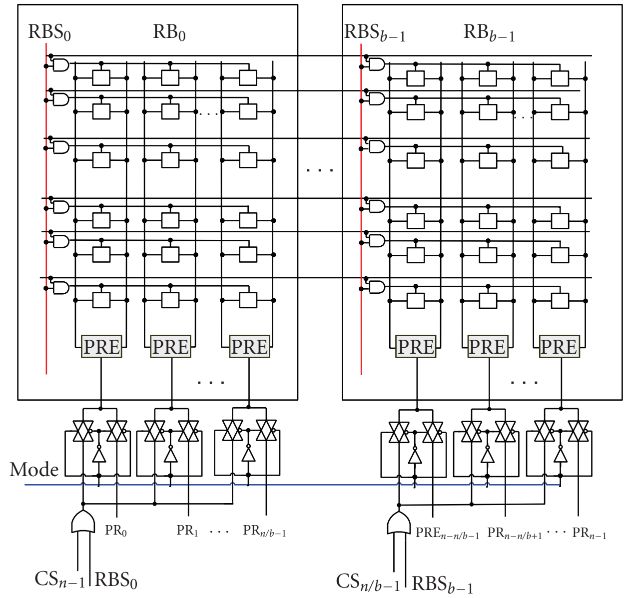

FIGURE 6: The control circuitry.

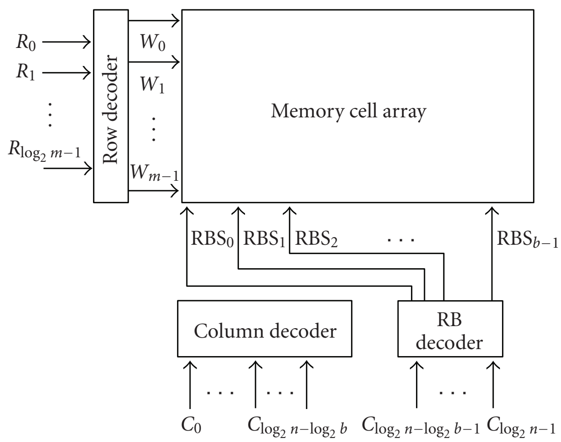

FIGURE 7: Architecture of the bank-based precharged memory array.

address are entered to the RB decoder. The generated row bank select signal RBS_j then can be used to activate the precharge circuitry as shown in Figure 6. The column selected signals (CS_j) are generated by the column decoder. The inputs of the column decoder contain $\log_2 n - \log_2 b$ bits. The outputs of the column decoder are used to identify a unique column within the specified row bank. Moreover, if the decoded column select signal denotes the last column of current accessed row bank, then this signal can also be used to activate the precharge circuitry of next row bank as shown in Figure 6. We can see from Figure 6 that only simple logic gates are sufficient to implement the bank-based precharge technique. Therefore, we can predict that the incurred hardware overhead will be very low. Moreover, the low-power technique is transparent to the BIST circuitry. Therefore, it is not necessary to modify the BIST circuitry.

The BIST circuitry used to implement the bank-based precharge technique is almost the same as the traditional BIST designs. The major modifications are within the memory cell array. That is, we should modify the precharge control circuitry as shown in Figure 6. Moreover, a row bank decoder should also be included.

5. SIMULATION RESULTS

The power dissipation reduction of the proposed row bank-based precharge technique depends on the memory organization. In other words, it depends on the parameters n and b. Since only one row bank is precharged in low-power test mode, the precharge power is approximately $1/b$ of that of the original precharge approach. However, in normal mode, since the accessed memory cell is not predictable, it is necessary to precharge all memory columns for correct operation. Therefore, it will consume more precharge power. It is evident that the low-power characteristics of the DWL architecture can be maintained. Moreover, the memory cells in the unselected row banks will not suffer from a stress called read equivalent stress (RES) [15]. We can see that the memory cells in the unselected banks will not consume power due to the RES.

The hardware overhead to implement this technique involves the modified control circuitry and the row bank decoder. The row bank decoder is basically parts of the original column decoder. Therefore, its hardware overhead can be neglected. From Figure 6 we can see that only two transmission gates and one inverter are added for each memory column. Moreover, one OR gate is added for each row banks. The number of transmission gates N_{TG} for the modified control circuitry is $2n$. Similarly, the number of inverters N_{INV} is n. The required number of 2-input OR gates N_{OR} is b. Therefore, the number of transistor counts (N_{TR}) for the modified control circuitry can be expressed as

$$\begin{aligned} N_{TR} &= 2 \times N_{TG} + 2 \times N_{INV} + 6 \times N_{OR} \\ &= 4n + 2n + 6b \\ &= 6(n + b). \end{aligned} \tag{1}$$

For an $m \times n$ memory array, we assume that 6-transistor SRAM cells are used. The number of transistors of the memory array (N_{TRMEM}) is $6mn$. The hardware overhead of the control circuitry (HO) can be expressed as

$$HO = \frac{N_{TR}}{N_{TRMEM}} = \frac{6(n+b)}{6mn} = \frac{n+b}{mn}. \tag{2}$$

In our analysis, the transistor count of the memory cell array is considered. The transistor counts of other circuitry (e.g., row/column decoders, precharge circuitry, timing generation circuitry, etc.) are neglected for simplicity. It is evident that the real values of hardware overhead will be less than the analyzed values.

Table 1 shows the hardware overhead of the modified control circuitry for different combinations of m, n, and b. From this table, we can see that the values of b and n have no effect upon the hardware overhead. For a fixed value of n, if

Table 1: Hardware overhead of the control circuitry.

	$n = 256$		$n = 512$		$n = 1024$	
	$b = 4$	$b = 8$	$b = 4$	$b = 8$	$b = 4$	$b = 8$
$m = 128$	0.8%	0.8%	0.8%	0.8%	0.8%	0.8%
$m = 256$	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%
$m = 512$	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%
$m = 1024$	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%

Table 2: Comparisons of power consumptions for RB = 4.

Row bank	Conventional		RB = 4			
Operation	read	write	read	write	read	write
Bank Pre.	∼BP	∼BP	∼BP	BP	∼BP	BP
256 × 128	18.7	19.2	15.1	12.5	15.5	12.8
Power saving (%)			19.3%	33.2%	19.3%	33.3%
256 × 256	32.1	32.5	22.7	16.3	23.0	16.6
Power saving (%)			23.4%	42.4%	24.6%	42.2%
512 × 256	39.0	39.6	33.1	29.6	33.9	30.0
Power saving (%)			15.1%	24.1%	14.3%	24.2%
512 × 512	63.8	64.0	51	42.9	52.5	43.2
Power saving (%)			20.1%	32.8%	18.0%	32.5%

we increase the value of m, then the hardware overhead will decrease. Since the values of hardware overhead are all less than 1%, we can conclude that the hardware overhead of the control circuitry is almost negligible.

Extensive Spice simulations are conducted to compare the power consumption between the traditional pre3charge methods and the proposed bank-based approach. Since the transistor count of the additional gates is greatly less than that of a memory column, therefore, the power consumption of the additional hardware is negligible. Simulation results for RB = 4 and RB = 8 are shown in Tables 2 and 3, respectively. Two memory sizes are used: 256 × 128 and 256 × 256. They are considered bit-oriented. In these tables, the "Conventional" column denotes that the conventional precharge technique is used. The columns with "∼BP" denote that the bank-based technique is not used. Similarly, the columns with "BP" denote that the bank-based technique is used. Both the power consumptions for "read" and "write" operations are shown. From these tables we can see that if more row banks are used, then we will achieve higher power saving. According to these tables, 48.9% power reduction can be achieved for a 256 × 256 bit-oriented SRAM when the memory is divided into 8 row banks.

The additional hardware is only an OR gate for each row bank and two transmission gates for each memory column. Since the transistor count for the additional gates is greatly less than that for a memory column, therefore, the power consumption of the additional hardware is negligible. Moreover, the power consumptions shown in Tables 2 and 3 are simulation results for the whole memory chip. They have already involved the power consumption of the additional hardware.

Since one extra transmission gate is added for the original precharge signal, one transmission gate delay will slightly

Table 3: Comparisons of power consumptions for RB = 8.

Row bank	Conventional		RB = 8			
Operation	read	write	read	write	read	write
Bank Pre.	∼BP	∼BP	∼BP	BP	∼BP	BP
256 × 128	18.7	19.2	14.1	11.0	14.5	11.3
Power saving (%)			24.6%	41.2%	24.5%	41.1%
256 × 256	32.1	32.5	22.7	16.3	23.0	16.6
Power saving (%)			29.3%	49.2%	29.2%	48.9%
512 × 256	39.0	39.6	32.6	27.5	33.0	28.8
Power saving (%)			16.4%	29.4%	16.7%	27.3%
512 × 512	63.8	64.0	49.2	39.8	49.7	40.9
Power saving (%)			22.9%	37.6%	22.3%	36.1%

affect the overall performance. However, we can cure this dilemma by the popularly used transistor-sizing technique. It will of course incur some hardware overhead. However, if power consumption is the main concern, the slightly increased hardware overhead will be tolerable.

As compared with previous works [12, 13], our results are still superior than theirs. For example, the results shown in [12] are based on a linear memory architecture. This architecture is not suitable for today's high-capacity memories. Moreover, over-simplified models for parasitic capacitance are used. Therefore, although about 40% power reduction (in average) can be achieved for small-size memories, the proposed models are not accurate enough. In [13], the authors still use a simplified model which counts the number of read and write operations to estimate the power consumption. Moreover, since more OR gates are required for the memory array, it will incur higher hardware overhead.

Instead of using simple power models to estimate the power consumption, we have conducted massive Spice simulation for different sizes of memory chips. It is evident that our results are more accurate. From Tables 2 and 3, we can see that significant power saving can be achieved as compared with the conventional memory arrays (without bank precharge technique). Since the address sequences are not modified during the BIST session, our approach will also not decrease the fault coverage. This conclusion can be found in [14]

6. CONCLUSIONS

In this paper, a row bank-based precharge technique based on DWL architecture is proposed for low-power testing of embedded memories. The characteristic of predictable address sequence during test is adopted to reduce the test power. Instead of precharging the entire memory array in the traditional approaches, we only precharge the current accessed row bank. The precharge power then can be reduced to $1/b$ of that of the traditional precharge approaches. The hardware overhead to implement the row bank-based techniques is analyzed. Simulated results show that the hardware overhead is almost negligible. Moreover, the BIST design to implement the low-power technique is almost the same as the conventional BIST designs. According to experimental results, 48.9% power reduction can be achieved for a 256×256 bit-oriented SRAM when the memory array is divided into 8 row banks. Moreover, if the number of row banks increases, the power saving will also increase.

REFERENCES

[1] Semiconductor Industry Association (SIA), "Int'l Technology Roadmap for Semiconductors," 2003.

[2] M. Yoshimoto, K. Anami, H. Shinohara, et al., "A divided word-line structure in the static RAM and its application to a 64K full CMOS RAM," *IEEE Journal of Solid-State Circuits*, vol. 18, no. 5, pp. 479–485, 1983.

[3] A. Karandikar and K. K. Parhi, "Low power SRAM design using hierarchical divided bit-line approach," in *Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD '98)*, pp. 82–88, Austin, Tex, USA, October 1998.

[4] S. Bhattacharjee and D. K. Pradhan, "LPRAM: a novel low-power high-performance RAM design with testability and scalability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 637–651, 2004.

[5] S. Bhattacharjee and D. K. Pradhan, "A low power RAM design," U.K. Patent filed, June 2003.

[6] C. Cheng, C.-T. Huang, J.-R. Huang, C.-W. Wu, C.-J. Wey, and M.-C. Tsai, "BRAINS: a BIST compiler for embedded memories," in *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT '00)*, pp. 299–307, Yamanashi, Japan, October 2000.

[7] A. Benso, S. Di Carlo, G. Di Natale, P. Prinetto, and M. Lobetti Bodoni, "A programmable BIST architecture for clusters of multiple-port SRAMs," in *Proceedings of IEEE International Test Conference (ITC '00)*, pp. 557–566, Atlantic City, NJ, USA, October 2000.

[8] A. Benso, S. Chiusano, G. Di Natale, and P. Prinetto, "An online BIST RAM architecture with self-repair capabilities," *IEEE Transactions on Reliability*, vol. 51, no. 1, pp. 123–128, 2002.

[9] M. Miyazaki, T. Yoneda, and H. Fujiwara, "A memory grouping method for sharing memory BIST logic," in *Proceedings of the Conference on Asia South Pacific Design Automation (ASP-DAC '06)*, pp. 671–676, Yokohama, Japan, January 2006.

[10] P. Girard, "Survey of low-power testing of VLSI circuits," *IEEE Design & Test of Computers*, vol. 19, no. 3, pp. 80–90, 2002.

[11] P. Rosinger, B. M. Al-Hashimi, and N. Nicolici, "Scan architecture with mutually exclusive scan segment activation for shift- and capture-power reduction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 7, pp. 1142–1153, 2004.

[12] L. Dilillo, P. Rosinger, B. M. Al-Hashimi, and P. Girard, "Minimizing test power in SRAM through reduction of pre-charge activity," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '06)*, vol. 1, pp. 1159–1164, Munich, Germany, March 2006.

[13] H. Cheung and S. K. Gupta, "A BIST methodology for comprehensive testing of RAM with reduced heat dissipation," in *Proceedings of IEEE International Test Conference on Test and Design Validity*, pp. 386–395, Washington, DC, USA, October 1996.

[14] A. J. van de Goor, *Testing Semiconductor Memories: Theory and Practice*, COMTEX, Gouda, The Netherlands, 1998.

[15] L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel, S. Borri, and M. Hage-Hassan, "Efficient March test procedure for dynamic read destructive fault detection in SRAM memories," *Journal of Electronic Testing: Theory and Applications*, vol. 21, no. 5, pp. 551–561, 2005.