The article has been accepted for inclusion in a future issue of ETRI Journal, but has not been fully edited. Content may change prior to final publication.
http://dx.doi.org/10.4218/etrij.15.0115.0040

Fully Programmable Memory BIST for Commodity DRAMs

Ilwoong Kim, Woosik Jeong, Dongho Kang, and Sungho Kang

ABSTRACT— In order to accomplish a high speed test on low speed Automatic Test Equipment (ATE), a new instruction based, fully programmable memory, Built-in Self-Test (BIST) is proposed. The proposed memory BIST generates a high speed internal clock signal by multiplying the external low speed clock signal from the ATE. For maximum programmability and small area overhead, the proposed BIST receives test algorithms from the external ATE and stores them as a unique set of instructions, as well as test sequences of unique instructions. An external ATE is able to provide complicated and hard-to-implement functions, such as loop operations and refresh-interrupts, instead of implementing them into the memory BIST, thereby simplifying the BIST hardware structure. The proposed memory BIST is a practical test solution for reducing the overall test cost for the mass production of commodity DDRx SDRAMs.

Keywords— Built-in self-test, DRAM, low cost, at-speed test.

I. Introduction

As the capacity and density of semiconductor memories has rapidly increased, maintaining acceptable yields and quality have become the most critical challenges in semiconductor memory manufacturing. Most commodity Dynamic Random Access Memory (DRAM) is tested with a multitude of test algorithms in a mass production test, by using external automatic test equipment (ATE). Recently, in order to test high density and high speed commodity memories such as Giga-bit DDR3 SDRAM with at-speed, a high-end ATE platform is required but comes with an extremely high cost. This problem will get worse in the next-generation DDRx SDRAMs. There are some major requirements of a BIST for commodity DRAMs: high fault coverage, high test frequency, high diagnostic capability, and small area overhead. Most of the previous researches on BIST techniques have been developed for embedded memories on System on a Chip (SOC) [1]–[4], but the test algorithms for embedded memories are inappropriate for the mass production testing of commodity DRAMs. Specifically, programmability is mandatory, not optional; programming is used to build various types of test algorithms for the mass production test, as well as to determine the root cause(s) when critical faults occur, in order to screen them out. To satisfy these requirements, an instruction-based programmable memory BIST is the most suitable for commodity DRAMs. This paper proposes a new memory BIST, presents experimental data, and provides our conclusions on the proposed BIST.

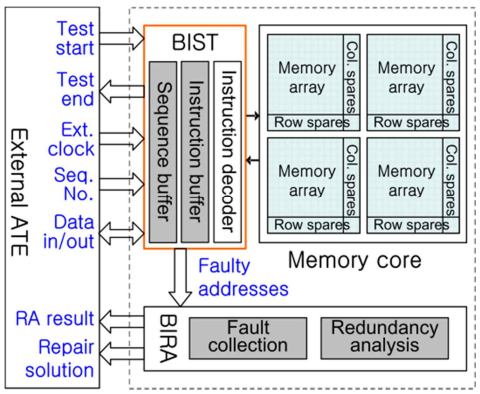


Fig. 1. Block diagram of the proposed memory BIST

Manuscript received Jan. 16, 2015; revised Apr. 4, 2015; accepted Apr. 21, 2015.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2012R1A2A1A03006255).

Ilwoong Kim (woong@soc.yonsei.ac.kr), Dongho Kang (fourier2@soc.yonsei.ac.kr), and Sungho Kang (corresponding author, shkang@yonsei.ac.kr) are with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Rep. of Korea.

Woosik Jeong (woosik.jeong@sk.com) is in the Product Development Group, Hynix Semiconductor Inc., Gyounggi-do, Rep. of Korea.

The article has been accepted for inclusion in a future issue of ETRI Journal, but has not been fully edited. Content may change prior to final publication.
http://dx.doi.org/10.4218/etrij.15.0115.0040

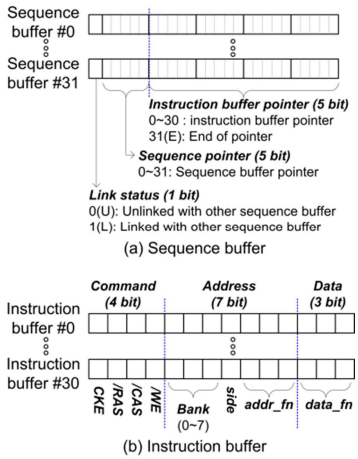
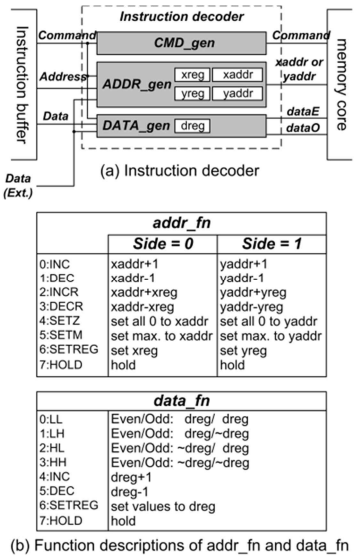


Fig. 2. Bit fields for a sequence buffer and an instruction buffer



addr_fn	Side = 0	Side = 1
0:INC	xaddr+1	yaddr+1
1:DEC	xaddr-1	yaddr-1
2:INCR	xaddr+xreg	yaddr+yreg
3:DECR	xaddr-xreg	yaddr-yreg
4:SETZ	set all 0 to xaddr	set all 0 to yaddr
5:SETM	set max. to xaddr	set max. to yaddr
6:SETREG	set xreg	set yreg
7:HOLD	hold	hold

data_fn	
0:LL	Even/Odd: dreg/ dreg
1:LH	Even/Odd: dreg/~dreg
2:HL	Even/Odd: ~dreg/ dreg
3:HH	Even/Odd: ~dreg/~dreg
4:INC	dreg+1
5:DEC	dreg-1
6:SETREG	set values to dreg
7:HOLD	hold

(b) Function descriptions of addr_fn and data_fn

Fig. 3. Instruction decoder and function descriptions

II. The Proposed Memory BIST

The main concept of the proposed memory BIST is to maximize the programmability and reusability of low-performance ATEs which are commonly used in the mass production testing of commodity DRAMs. Even if a commodity DRAM has a good BIST, an external ATE is required in order to provide input signals and automatically acquire test results.

Fig. 1 shows the block diagram of the proposed BIST which consists of a sequence buffer, instruction buffer, and instruction decoder. In the proposed memory BIST, a built-in redundancy analysis (BIRA), which consists of fault collection from the BIST and redundancy analysis to extract the repair solution, is used for wafer level test and repair [5].

Test algorithms for a commodity DRAM generated by a commercial ATE, generally comprise five unit operations as follows: command instructions, loop operations, refresh-interrupts for managing various types of retention tests, address operations, and data operations. In order to guarantee a high level of programmability for test algorithms as flexible as memory ATEs can provide, multiple levels of loop operations and various types of timers for refresh-interrupts for testing the retention time of DRAMs, must be supported. This is managed by an external ATE in the proposed BIST. Therefore, the proposed BIST handles only three unit operations: a command instruction, address operation, and data operation; there are also two types of buffers: an instruction buffer and a sequence buffer. An instruction buffer contains a command instruction, an address operation, and a data operation.

The proposed BIST is designed to perform at-speed test with low-frequency ATE. The external test clock frequency of the ATE can be multiplied by 4 or 8 to generate a faster internal clock frequency for the BIST, by using modified DLL logic. For this reason, several internal instructions must be delivered from the external ATE in an external clock cycle, so that a sequence buffer pointer, which contains pointers for instruction buffers to be executed sequentially, is transferred from the ATE to the BIST. Subsequently, a series of instructions are able to be executed with fast internal clocks.

Fig. 2 shows the bit fields for a sequence buffer and an instruction buffer. The proposed memory BIST has only 31 instruction buffers (from #0 to #30) and 32 sequential buffers. These are sufficient for storing complicated test algorithms for DRAM mass production.

Each sequence buffer of Fig. 2(a) consists of 26 bits. The first bit of a sequence buffer represents the Link status with the other sequence buffer. The next 5 bits represent the sequence buffer pointer. The final 20 bits represent instruction buffer pointers to be executed sequentially.

An instruction buffer pointer is comprised of 5 bits and represents a valid instruction buffer pointer when its value is from 0 to 30, while it represents the end of the instruction buffer when its value is 31.

If the first bit of sequence buffer A is set to 1 and the next 5 bits point to sequence buffer B, it means that all of the valid instructions of sequence buffer B will be executed after completing those of sequence buffer A.

Each instruction buffer consists of 14 bits for memory having 8 banks, such as a 1G-bit DDR3 SDRAM. In Fig. 2(b), the first 4 bits of the instruction buffer represent the command instruction for the DDR3 SDRAM, such as Active, Write, Read, and Precharge. The next 7 bits represent the address operations; the 5th to the 7th bits represent the bank address. The 8th bit selects

The article has been accepted for inclusion in a future issue of ETRI Journal, but has not been fully edited. Content may change prior to final publication.
http://dx.doi.org/10.4218/etrij.15.0115.0040

whether *x* addresses or *y* addresses are to be modified by the address function. The next three bits represent the address functions and the last three bits represent the data function.

Fig. 3 shows the structure of the instruction decoder of the proposed BIST and provides descriptions of the address function and data function for the instructions. The instruction decoder decodes each bit of the instruction buffer and generates a command, *x* address and *y* address, and *dataE/dataO* for the memory core.

The address generator of the proposed BIST (i.e., *addr_gen* of Fig. 3(a) has two address registers, *xaddr* and *yaddr*, to hold the current x and y address, respectively. It also has two registers, *xreg* and *yreg*, which hold address values to be added or subtracted into address registers by INCR or DECR functions of the *addr_fn* of Fig. 3(b).

The data generator of the BIST has two data registers (*dreg*) to hold the current data values, *dataE* for a clock rising edge and *dataO* for a clock falling edge.

III. Communication between BIST and ATE

Fig. 4(a) shows an example of a Scan algorithm with Checkerboard data background for a DDR3 SDRAM, which includes five loops (L1–L5) and they construct a 3-level loop (L1). Fig. 4(b) and Fig. 4(c) represent a set of unique instructions which are saved in the instruction buffers, and a set of sequence buffers to construct the same test algorithm described in Fig. 4 (a).

Fig. 5 shows an example of a timing diagram between an ATE to the proposed BIST; it is assumed that the internal clock frequency is eight times faster than the external clock frequency from the ATE. In order to construct the same test algorithm, the external ATE generates sequence pointers in every clock cycle, such as shown in Fig. 5. If a generated sequence number is synchronized with the external low frequency clock signal, the internal instructions for the sequence buffer are executed to the memory core. On every rising edge of the external clock, the sequence buffer pointer value is stored into a sequence register of the BIST. At the same time, external data is also stored into an external data register for setting *xreg*, *yreg*, or *dreg* of the BIST.

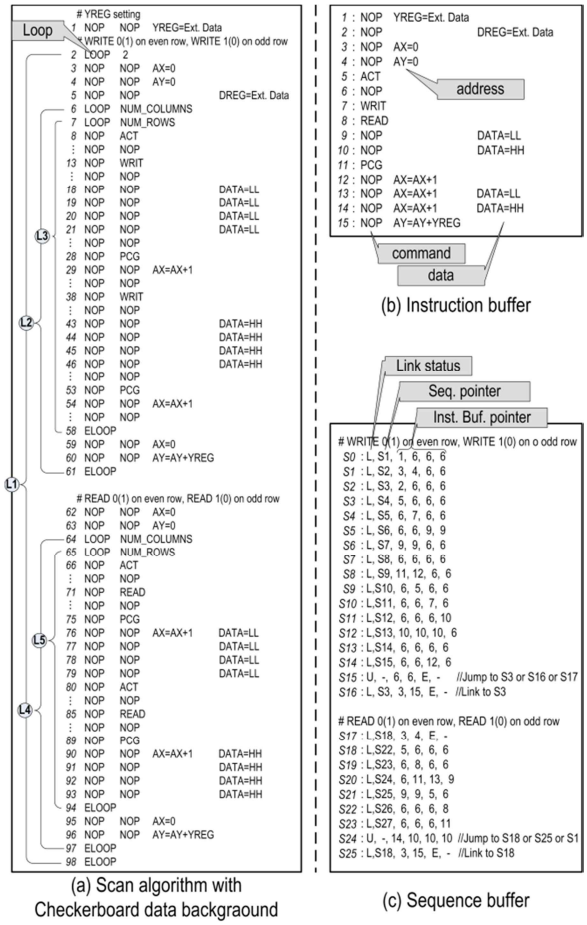


Fig. 4. Example of a test algorithm (Scan with Checkerboard data background) for a DDR3 SDRAM

When the Link status is set to 0 and an instruction pointer of a sequence buffer reaches the "End of pointer" or the fourth instruction pointer, the BIST retrieves the next sequence buffer

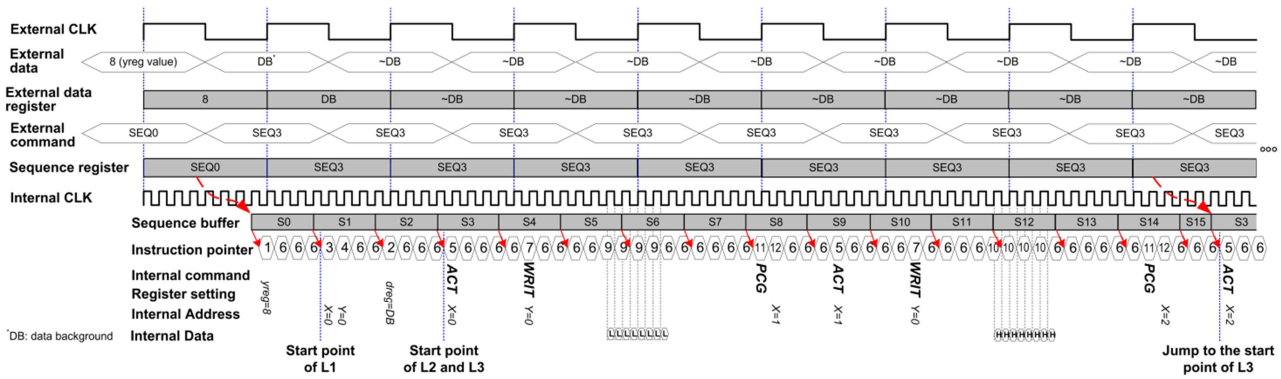


Fig. 5. Example of a timing diagram between ATE and BIST

The article has been accepted for inclusion in a future issue of ETRI Journal, but has not been fully edited. Content may change prior to final publication.
http://dx.doi.org/10.4218/etrij.15.0115.0040

pointer value from the sequence register, otherwise the sequence pointer value is ignored.

For this case, the external ATE generates a SEQ0 once and consecutive SEQ3s. SEQ0 indicates S0 in the first sequence buffer of Fig. 4(c). S0 is linked with S1 and S1 is the start point of the L1. S1–S14 are sequentially linked and S14 is linked with S15. However, S15 is not linked with any other sequence buffer. Therefore, the BIST retrieves SEQ3 from the sequence register and jumps to S3 which is the start point of the L2 and L3.

The external ATE needs to control the sequence buffer pointers and internal register setting; it is sufficiently knowledgeable with the *loop control* and *refresh-interrupt control* to build complicated test algorithms. Therefore, complicated control functions are not required in the proposed memory BIST which makes the structure of the proposed BIST very simple. In addition, the required area overhead for saving test algorithms into the memory BIST is very small, all the while guarantying full programmability and high speed testing with a low speed ATE.

IV. Experimental Results

The proposed BIST was designed with Verilog HDL and synthesized with 0.13-μm CMOS technology. Table 1 shows a performance comparison of the proposed BIST with previous researches [1]–[4]. In Table 1, the second row represents the area overhead of each BIST. There is a trade-off between the area overhead and the programmability. In a BIST for a commodity DRAM, a very high programmability has to be supported in order to cover the mass production test whose test conditions can be constantly changed. Although the area overhead of the proposed BIST is larger than those of [1]–[3], it shows an approximately three times smaller area overhead as compared with that of [4] and provides a very high programmability as shown in the third rows in Table 1. Unlike the previous researches [2]–[4], the timing control for retention test can be managed by ATE without extra hardware and additional test channels. This fully enables various kinds of retention testing algorithms. Therefore, typical mass production test algorithms (ex. Scan, MATS+, Moving Inversion, GalRow, GalCol, March LAd [6], etc.) for a commodity DRAM can be supported. Using dual instruction decoders, the clock speed of the proposed BIST is 1.25 ns and it can perform at-speed test for DDR3 SDRAM. Most of the current wafer level ATEs such as MT6060 (YOKOGAWA) and T5377 (Advantest) are able to generate clock frequencies above 100 MHz. If a DDR3 SDRAM adopts the proposed BIST and generates an eight times faster internal clock frequency than the external clock speed, the memory is able be tested at up to 800 MHz (1.6 G bps).Without high-speed ATEs, the proposed BIST can provide a practical solution for testing DRAM with at-speed.

Table 1. Performance Comparison

Feature	[1]	[2]	[3]	[4]	Proposed
Area (Gates)	~2.5K	> 300	7.9K	50K	15K
Programmability	Medium	Medium	High	High	Very High
Retention Test Support	N/A	Partial	Partial	Partial	Full
Testing Freq.	N/A	N/A	400MHz	972MHz	800MHz

V. Conclusions

In this paper, an instruction-based fully programmable high speed memory BIST solution for commodity DRAM mass production tests with low-speed ATEs is proposed. The proposed BIST provides very high programmability with reasonable area overhead, making it is possible to encompass typical mass production test algorithms for DRAMs. The proposed BIST can be a viable solution for maintaining yield and quality of commodity DDRx SDRAMs without using high-end ATEs.

VI. References

[1] B. Ismet, O. Caty, and Y. Wong, “Highly configurable programmable built-in self test architecture for high-speed memories”, *Proceedings of IEEE VLSI Test Symposium (VTS’05),* pp. 21-26, 2005.

[2] S. Boutobza, M. Nicolaidis, K.M. Lamara, and A. Costa, "Programmable memory BIST", *Proceedings of IEEE International Test Conference,* pp. 10, Nov. 2005.

[3] Y. Park, H. Kim, I. Choi, and S. Kang, “A Flexible Programming Memory BIST for Embedded Single-Port Memory and Dual-Port Memory,” *ETRI J*, vol. 35, No. 5, pp. 808-818, Aug. 2013.

[4] Kume, M., Uehara, K., Itakura, M., Sawamoto, H., Kobayashi, T., Hasegawa, M., Hayashi, H., "Programmable at-speed array and functional BIST for embedded DRAM LSI", *Proceedings of IEEE International Test Conference*, pp. 988 - 996, Oct. 2004.

[5] T. Han, W. Jeong, Y. Park, and S. Kang, “High Repair Efficiency BIRA Algorithm with a Line Fault Scheme,” *ETRI J*, vol. 32, No. 4, pp. 642-644, Aug. 2010.

[6] Al-Ars and A.J. van de Goor, “Static and Dynamic Behavior of Memory Cell Array Spot Defects in Embedded DRAMs,” *IEEE Trans. Computers*, vol. 52, no. 3, pp. 293-309, Mar. 2003.

