Extended Abstracts of the 2010 International Conference on Solid State Devices and Materials, Tokyo, 2010, pp1130-1131

High-speed MRAM Random Number Generator using Error-Correcting Code

Tetsufumi Tanamoto, Naoharu Shimomura, Sumio Ikegawa, Mari Matsumoto, Shinobu Fujita and Hiroaki Yoda

Advanced LSI Technology Laboratory, Corporate R&D Center, Toshiba Corporation,
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki 212-8582, Japan
Tel: +81-44-549-2192, FAX: +81-44-520-1257, E-mail: tetsufumi.tanamoto@toshiba.co.jp

Abstract

A high speed random number generator (RNG) circuits based on magnetoresistive-random-access-memory (MRAM) using error-correcting coding (ECC) post-processing circuits is presented. It is shown that this post-processing block can be shared with conventional memory ECC block and powerfully improves the quality of randomness with minimum overhead.

Introduction

On-chip RNG is one of key elements for secure system-on-chip. Various kinds of on-chip RNGs are proposed using physical noise signals such as transistor with random telegraph noise [1,2], diode junction noise or magnetic tunnel junction(MTJ) (Fig.1) for MRAM[3]. In particular, the RNG based on MRAM [3] is very promising for future secure chip by using MRAM as both RNG and nonvolatile reliable memory system. In any physical RNG system, after AD-converting physical noise signal, post-processing using digital circuits are needed to balance “0” and “1” and to eliminate periodicity and correlation of original signals. “Rejection-method” is most frequently used for the post-processing circuit (PPC), where two serial bit sequence “01” and “10” are replaced by “0” and “1”, whereas “00” and “11” are discarded. Bose-Chaudhuri-Hocquenghem (BCH) coding is also suggested for PPC [4]. However, such PPC is much larger and consumes more power than the noise sources, therefore, the PPC dominates area and power of the RNG.

In this paper, we present a novel MRAM-based RNG system, where ECC circuit which conventionally corrects memory errors is effectively *converted to the post-processing circuit* when RN is needed. MRAM produces RN by reducing the programming current to cause writing errors intentionally in RNG mode. This change of circuit between normal MRAM memory system and RNG is controlled from processors or other logic blocks.

Post-processing using ECC circuit

Fig.3 shows a proposed PPC shared with conventional error-correction block. The role of the circuit between error-correction of memorized bits and that of RNG post-processing is changed by switching a small number of transistors expressed by ECC-SW and RNG-SW. The data compression by this configuration of ECC plays a role of increasing entropy of RN. Block of n bits is transferred into k bits by ECC circuit. In Fig.3, ON/OFF of transistors g_0,.., g_{n-k} are determined by chosen BCH codes, which are given by a generator polynomial $G(x)=g0+g_1x+...+g_{n-k}x^{n-k}$. Detailed statistical tests mentioned below are carried out by connecting this circuit to external checking system. Fig.5 shows an implemented form of shared block of Fig.3 (Because of limited space, we show a case of (7,4) code). In a rejection method, the number of bits is reduced to about one-fourth of original data (compression rate is about 1/4). On the other hand, PCC by (n,k,t) BCH code transform n bits to k bits, thus the compression rate is k/n (t is an error correcting capability.

MRAM RNG

MRAM cell arrays are integrated by 130-nm front process and 240-nm back-end process [6-8] (Fig.6). MRAM RN is generated when a probability of change of magnetism of free layer is 50 % (Fig.1). Fig. 7 shows MRAM switching probability P_{write} as a function of switching current with various pulse widths. The slope of P_{write} slightly becomes smaller for shorter pulse width t_{write} (speed-up). This means shorter pulse is favorable against current fluctuation.

Results

Figs. 9-11 show results of statistical tests of randomness after ECC post-processing for n=31, 63, and 127 codes (Fig. 4) as a function of error-correcting capability t. We treat three typical types of data (Fig.9) which a rejection method compresses to 24.8% (data A), 20.0% (data B) and 23.1% (data C), respectively. In contrast, for example, for t=2 ECC, these rates are fixed to k/n and given as 90.0% (n=127), 80.0% (n=63) and 67.7% (n=31). Statistical test suite SP 800-22 [9] that contain 16 types of tests and judges 160 test results is used for million number of bits with setting P-value to 0.01. Although there are several kinds of Pass/Fail judgement to this test, here we simply count the number of test failure and judge Pass if it is 2 or less. Note that raw bit sequence just after MRAM unit fails to almost all these tests.

Fig. 9 shows the result for 0 and 1 balanced bits (data A) and proves the ability of improvement by the ECC post-processing. Here we could not see the correlation regarding t. This does not follow the prediction of [4] in which better RN is obtained as t (or d) increases (d and t has a relation $d>2t$). Because RNG speed downs at rate k/n, and $n-k$ extra redundant memory cells are required, this result shows that we can choose small t for balanced bits. Fig. 10 and 11 show results of ECC post-processing for unbalanced bits with t_{write} =30ns (data B) and with t_{write} =10ns (data C). We found that coupling with linear feed back resistor (LFSR) is effective (Fig.5). We examined several length of LFSR and found that length two or three LFSRs are sufficient. In addition, Fig. 11 show that speed up of current pulse from t_{write} =30ns to t_{write}=10ns increases the quality of randomness. When the read time is 10ns, one bit RN can be generated for just two cycles at 50Hz clock, resulting in 25MHz for RNG. Table 1 shows that generation speed of MRAM-based RNG is much faster than those of state of the arts reported in [1][2]. Thus, we have proven that the proposed ECC effectively increases both randomness quality and generation speed.

Table. 1. Comparison of RN generation speed.

RNG	speed
[1]	2MHz
[2]	0.2MHz
MRAM	25MHz

References

[1] Matsumoto et al. ISSCC 2008, p414. [2] R. Brederlow, R. Prakash,et al, ISSCC 2006, p536. [3] A. Fukushima et al. Japanese Patent P 2008-310403. [4] P. Lacharme, FSE2008 p 334. [5] S.B. Wicker, *Error Control Systems for Digital Communication and Storage*, 1995. [6] T. Kishi et al. IEDM 2008, p309. [7] H. Aikawa et al., Spintech IV, 2007, Poster 79. [8] S. Ikegawa et al., IEEE MML2007, TUE-16. [9] http://csrc.nist.gov/publications/

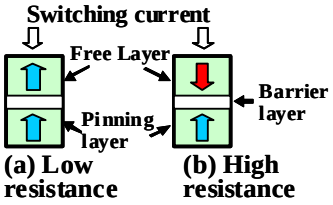

Fig. 1. MRAM device based on perpendicular magnetic tunnel junction (MTJ). Random signals are generated by reducing the programming current to cause writing errors intentionally[3]. To reset free layer, 800mV is applied to MTJ.

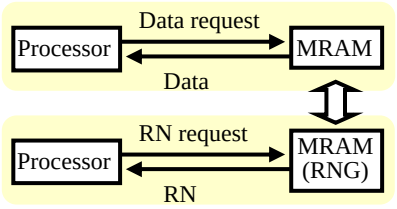

Fig.2. MRAM is used to generate RN when processor requests RNs.

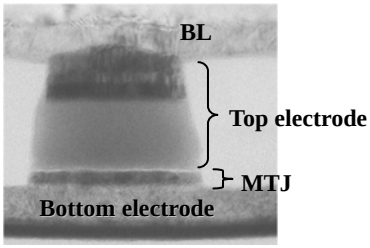

Fig.6. Cross-sectional TEM image of MTJ patterned by ion-milling method. The diameter of element is 55nm. MTJ contains capping layer /perpendicular reference layer /MgO / Fe-based L10-alloy /underlayer.

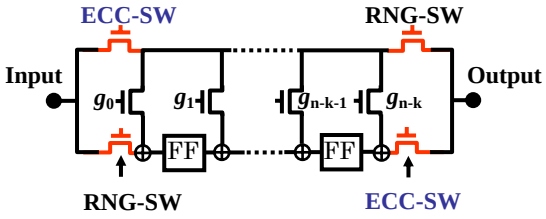

Fig. 3. Proposed RNG post-processing unit which is shared with memory bit error-correcting block. FF is a flip-flop element. In ECC mode, "ECC-SW" is ON and in RNG mode, "RNG-SW" is ON. $g_0,...,g_{n-k}$ is determined by G(x). For example, when $G(x)= x^5+x^2+1$, g_0, g_2, and g_5 are 1 and corresponding transistors are ON states.

n	*k*	*t*	G(x)
31	26	1	45
31	21	2	3551
31	16	3	107657
31	5	5	5423325
63	57	1	103
63	51	2	12471
63	45	3	1701317
63	39	4	166623567
127	120	1	211
127	113	2	41567
127	106	3	11554743
127	99	4	3447023271

Fig.4. BCH codes we used. Number of G(x) shows polynomials [5]. For example, 45 → 100,101 → $G(x)=x^5+x^2+1$.

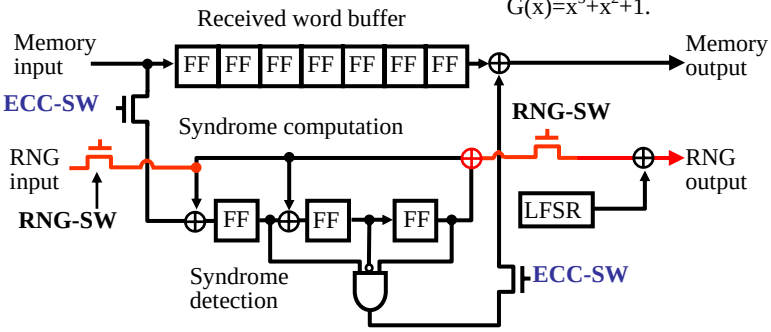

Fig.5.Proposed RNG post-processing unit shared with ECC decoder. ECC for RNG is easily changed by switches RNG-SW and ECC-SW. Example for (7,4) code.

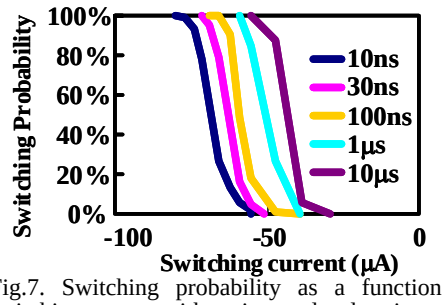

Fig.7. Switching probability as a function of switching current with various pulse duration t_{write}.

	balance	t_{write}
data A	51.1% "1"	30ns
data B	27.6% "1"	30ns
data C	36.3%"1"	10ns

Fig.8. Typical RN examined here. Data *A* is an average sample. Data *B* is a low-quality sample. Data *C* is a higher speed sample.

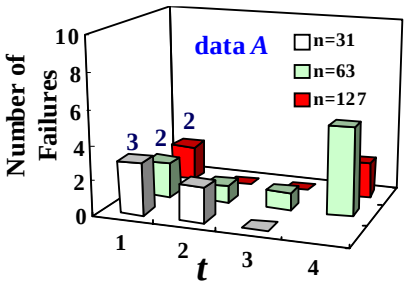

Fig.9. Result of statistical test (NIST800-22). Number of failures after ECC post-processing for good original RN bits (data *A* in Fig.9) as a function of error correcting capability *t* for n=31, 63 and 127 BCH codes.

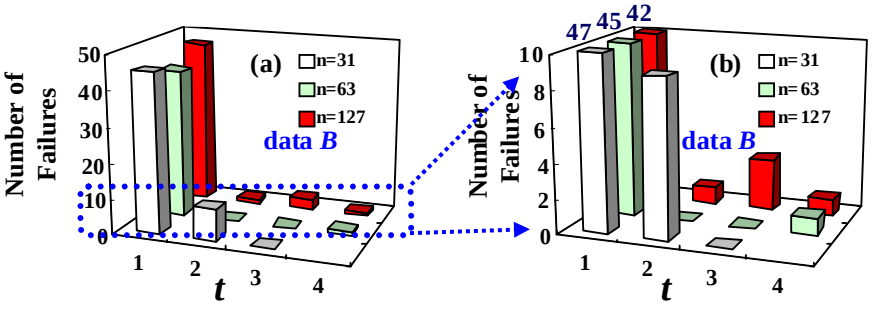

Fig.10. Result of statistical test (NIST 800-22) for data *B* (unbalanced bits). (a) ECC post-processing with 3bit LFSR (Fig.5). LFSR improves the quality of randomness. (b) Part of Fig. (a).

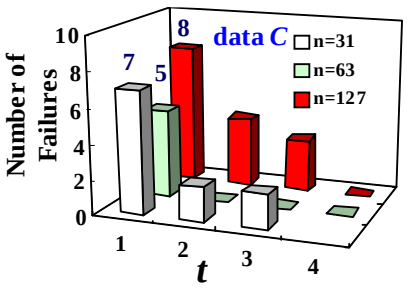

Fig.11. Result of statistical test (NIST 800-22) for data *C* (fast unbalanced bits). ECC post-processing with 2bit LFSR (Fig.5). Compared with Fig.10, speed up and coupling with LFSR improves the quality of randomness.