2020 25th IEEE European Test Symposium (ETS)

MBIST Support for Reliable eMRAM Sensing

Jongsin Yun
Mentor, A Siemens Business
Wilsonville, USA
jongsin_yun@mentor.com

Benoit Nadeau-Dostie
Mentor, A Siemens Business
Ottawa, Canada
benoit_nadeau-dostie@mentor.com

Martin Keim
Mentor, A Siemens Business
Wilsonville, USA
martin_keim@mentor.com

Cyrille Dray
ARM
Sophia Antipolis, France
Cyrille.Dray@arm.com

Mehdi Boujamaa
ARM
Sophia Antipolis, France
Mehdi.Boujamaa@arm.com

***Abstract*—eMRAM (embedded Magnetoresistive Random Access Memory) is an attractive solution in many non-volatile memory applications because of its small size, fast operation speed, and good endurance. However, due to a relatively small on/off resistance separation, it is a challenge to set an optimal reference resistance to reliably differentiate between a read memory data "1" and "0". Several trimming circuits are described in the literature to finely adjust a reference resistance value. These circuits are controlled from chip inputs causing time-consuming tests and off-chip engineering analysis. This paper presents a fully automated on-chip trimming process leveraging existing memory BIST (Built-In Self-Test) resources. It analyzes a massive amount of array property data with a minimal number of tests and optimizes the reference trim settings on-chip without the need for any external intervention.**

Keywords—MRAM, yield, trim, reference, read operation.

I. Introduction

STT (Spin-Transfer Torque)-MRAM is a type of memory that encodes data as a spin polarity of magnets in its ferromagnetic metal layer. Based on the polarity of the layer, the resistivity of the MRAM cell can be switched to either a high (anti-parallel, R_{AP}) or a low (parallel, R_P) value as shown in Fig. 1. The switching of a memory datum is achieved by spin-polarized tunneling current flowing through MTJ (Magnetic Tunnel Junction) which exerts torque on the local magnetization. Slonczewski [1] and Brinkman [2] well explained the switching mechanisms and tunneling conductance behaviors. Several studies show fast switching and high endurance of MRAM to propose MRAM as a potential replacement solution for last level cache memory [3-6].

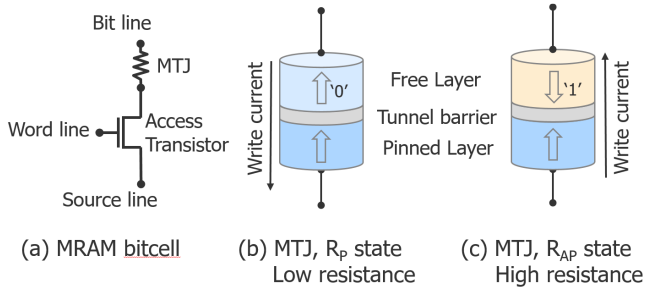

Fig. 1. The component of MRAM bitcell and its MTJ states

During Read operation, an MTJ resistance in a selected bitcell (R_{bit}) is compared with a reference resistance (R_{REF}). The read value is said to be "1" ("0") when its resistance is higher (lower) than that of R_{REF}. The resistance comparison is done by amplifying the differential of Bit Line (BL) discharging rates between the active BL and the reference BL.

An example of On and Off state resistances distribution of an MRAM is illustrated in Fig. 2 (a). To read all memory data correctly, the reference resistance value must be set lower than any R_{AP} state ("1") resistance and, at the same time, higher than any R_P state ("0") resistance. The reference resistance should be set between the distribution tails of R_P and R_{AP}. These tail bits are weakest bits in the array that is unstably changing by process update, array configuration, total array size, temperature use case, etc. Setting R_{REF} is especially challenging in MRAM because the On and Off state resistance separation is 1 or 2 orders of magnitude smaller in MRAM with respect to other types of resistive memories. Moreover, inherent variations from a large memory array and inhomogeneous temperature sensitivity between R_P and R_{AP} makes it even more challenging to set the correct reference resistance.

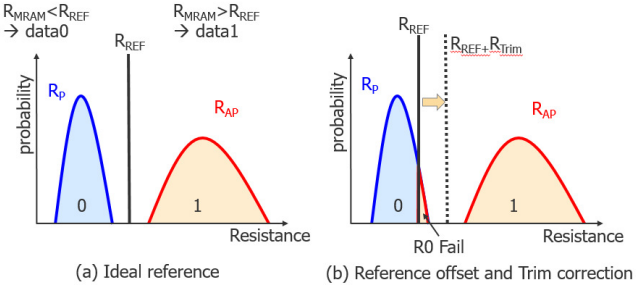

Fig. 2. distribution of MRAM Resistance for value "1" and "0" with sensing reference resistance setting.

A hardware preset R_{REF} may become inadequate for some of the bit-cells due to process update and other variances that happen after the design completion. This reference setting will incorrectly interpret some of the logical "0" state memory data as a logical "1" and hence cause a Read0 (R0) fail as we illustrate in Fig. 2 (b). Reference trimming circuits are suggested in recent MRAM application literature [3, 4] to accommodate this issue. The authors connect switchable series resistances to the reference BL to add additive resistance R_{Trim} into R_{REF}. This shifts the overall reference resistance value to a proper value. Such a trimming circuit allows post-manufacturing fine-tuning of the reference resistance from the device IOs. The reference trim adjustment process improves the read margin for both of the data types '1' and '0', therefore, increasing reliable sensing of the stored logical value and increase yield.

Collecting and analyzing the full set of memory properties is another challenge. It requires an impractically large amount of test resource to measure and analyze analog properties of a full array, such as resistance value of each cell at the state of "1" and "0". If only a small number of cells would be sampled for the test, it may be difficult to capture the tail behavior of

978-1-7281-4312-5/20/$31.00 ©2020 IEEE

the R_P and R_{AP} distributions. One way of efficiently collecting and analyzing the massive amount of memory properties is by using on-chip generated and evaluated test data. The test result abstracts the analog properties into binary pass/fail data.

In this paper, we will describe a method to determine the R_{Trim} value based on memory test data collected from an existing MBIST circuit. This method leverages existing MBIST resources to accumulate MRAM memory failures. The collected fail information is analyzed in a new hardware component, the Trim Feedback Circuit, to evaluate and determine the optimum trim value. This fully automated, on-chip solution determines the optimal trim value at a relatively low-cost point compared to other methods, which require external access and off-chip processing.

II. Reference Value Search

We call a memory bitcell a 'fail bit' when it's read data does not match its previously written data. The bit cell may fail because of a defective device but can also fail due to an inadequate reference setting as shown in Fig 2 (b). In matured technology, the probability of a defective memory cell is far lower than that of a good memory cell, thus the defective fail bit count (FBC) is very low.

The number of fails associated with incorrect reference settings modulates based on the $R_{REF}+R_{Trim}$ setting relative to the resistance distribution of the memory array. The FBC at each trimmed reference resistance setting is illustrated in Fig 3. The fail count will exponentially increase as $R_{REF}+R_{Trim}$ setting moves into the resistance distribution range of the tested memory array. The MBIST circuit will compare FBCs to compute the final $R_{REF}+R_{Trim}$ above the highest R_P tail and below the lowest R_{AP}, achieving the lowest FBC. During the process, the analog properties of an entire memory array were converted to few FBC numbers.

The trim procedure is applied prior to all other memory tests to avoid any incorrect repair decision or excessive ECC budget evaluation. Therefore, some of the arrays may contain a few faulty bits but are still considered to be working, assuming that spare rows/columns or even ECC can eliminate these failures. The presence of faulty bits complicates the trimming procedure significantly as there is no trim setting for which all read operations are successful as shown in Fig 3. The trim procedure runs analysis for both, the W0R0 (write "0" followed by read "0") fail and the W1R1 fail. The R_{Trim} value will be adjusted to the safe zone where both R0 fail count and R1 fail count are minimized.

Let's look at the W1R1 case first. When the W1R1 operation is executed by MBIST and then analyzed by the Trim Feedback Circuit, we can collect FBC for each trim setting-under-test. A good memory array without defective cells will show no fail reads when the combined reference resistance ($R_{REF}+R_{Trim}$) is lower than any R_{AP} value of the memory. The fail count increases when memory cells have a R_{AP} value smaller than the combined reference resistance value ($R_{REF}+R_{Trim} > R_{AP}$). Therefore the FBC will monotonically increase as R_{Trim} increases. Similarly, W0R0 fail numbers decrease by increasing the trimmed reference value. When the reference value becomes higher than the highest R_P resistance of any memory cell, FBC will reach zero and stay zero beyond the reference value. Since the resistance distribution of R_P and R_{AP} are not expected to overlap for a good die, the trim value with the first W1R1 fail is supposed to be higher than the trim resistance value of the last W0R0 fail. The trim range between these two transients is a safe zone where we can settle the trim setting without read operation failure. To give equal read margin for the data "1" and the data "0", the final trim is set in the middle of the safe zone.

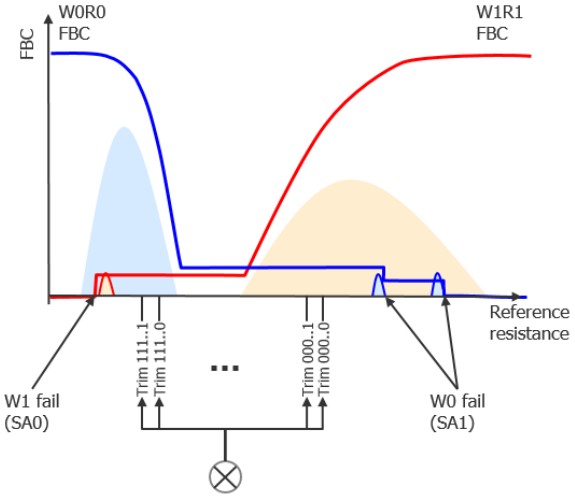

Fig. 3. Illustration of FBC relative to different reference resistance setting of an array which has some write 1 fails (stuck at 0) and write 0 fails (stuck at 1)

To draw the FBC graph like Fig 3 requires full FBC measurement at all R_{Trim} steps for W1R1 and W0R0. This is a very expensive solution since the entire memory test needs to be repeated as many as twice the number of trim steps. To make this evaluation more practical, the number of tests has to be reduced significantly. Eliminating unnecessary MBIST runs not only reduces the run time of the method but also reduces the impact on the wear-out of the MRAM.

The FBC is a cumulative curve, so the W1R1 FBC curve is monotonically increasing, and the W0R0 FBC curve is monotonically decreasing as the trim resistance is increasing. This is a very convenient behavior which allows us to perform the "binary-judge-based-search" that can eliminate unnecessary tests. If the current trim setting shows a non-zero fail count at the W1R1 test (FBC>0), we know the transient of no-fail to first-fail happens at a lower reference resistance value than the current trim setting. If there were no W1R1 fails at the current test (FBC=0), we know the fail transient happens at a higher reference resistance. Therefore we can select the middle trim value among remaining available trim resistance settings as the next trim value to evaluate, and discard either half of the trim setting. In other words, we are removing half of the trim setting after each test until we find the final trim setting.

The diagram in Fig 4 shows a binary-judge-based-search method in more detail. Here, the example of a 4-bit trim resolution search for W1R1 fail transient is shown. The search starts with a mid-trim value of the available range. This sets all trim bits to 0 except the MSB (most significant bit), which is set to "1". The example shows zero fails on the mid-trim value, so we know that all resistance values below the "1000" setting (i.e. trim value numbers larger than "1000") are no longer of interest to investigate. Rather we should continue searching for the first-fail transient from the resistance values higher than the "1000" setting (i.e. the trim value number lower than "1000"). Therefore, to create the next trim value to evaluate, the MSB is set to "0" and the "1" moves to the next bit of the trim, i.e. "0100". In this example, the FBC becomes

larger than 0 for the “0100” trim setting, so the next bit can stay “1”. This means the first two bits are fixed to “01”. We move on to the next mid-trim value which is “0110”. The search test will continue until the LSB (least significant bit) is determined. The example in Fig 4 judges 16 different trim settings but the test was performed only 4 times. In general, the binary-judge-based-search needs “n” number of tests to evaluate “2^n” trim steps. It is executed twice, once each for W0R0 and W1R1. The number of test saving exponentially increases with the number of trim bits used.

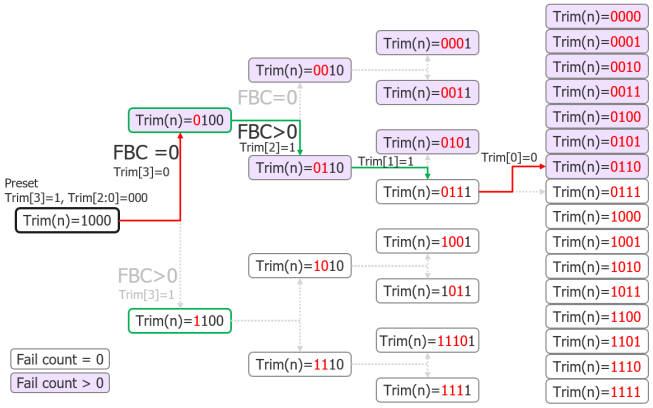

Fig. 4. The binary-judge-based-search for the first-fail boundary, n=4 trim

III. Write-Fail-Screen Mode

Searching for the first-fail transient is a good option to maximize the success rate of the read operation especially when the repair option is not available. But it may incorrectly compute the trim resistance setting if the memory array contains extrinsic fails. Those defective memory bitcells may have a resistance away from the normal distribution of resistance like the stuck-at fault write fail examples shown in Fig 3. Such a cell may actually be written with a much lower or much higher resistance than normal cells. Those cells fail during read operation at any available trim settings. We call these fails “write fail”. These write fails will extend the tail of FBCs and make the safe zone disappear. Although those fails may be fixable at the later manufacturing test steps using repair and/or ECC, during the trim search process, however, these defective bits will cause a reference setting into a fail zone. Therefore, the first-fail transient search algorithm stated above will not be applicable as-is and must be extended to correctly deal with defective memory bit cells.

One way to avoid the effect from these write fail is to eliminate these write fail from the FBC curve. We can easily identify those write fails as fails at the extreme trim condition. For example, to find a W0 fail, one can write 0 and then read the cells at the highest resistance reference setting (in our example trim bits (n=4) are set to 0000). This represents the best possible trim setting for the R0 operation. Any R0 fail with this trim setting cannot be recovered by further trim adjustments. Similarly, one can count W1 fail bits by reading the array with the lowest resistance reference setting (trim bits are set to 1111 in our example). The counted write failures need to be subtracted from the respective FBC to represent only the pure trim adjustment related fails. The pre-screening of write fails will allow the FBC curve to reach zero. This will allow us to use the simple first-fail transient search mentioned in the previous section to set the final trim value without complicated search cases introduced by those extrinsic fail level or complicated correctable bit budget calculations for multi-bit fails case. The example cumulative FBC graphs before and after the pre-screen of the write fails are illustrated in Fig 5. It is recommended to perform the trim evaluation test at the highest operating temperature where the read window is the smallest and write operation encounter fewer failures.

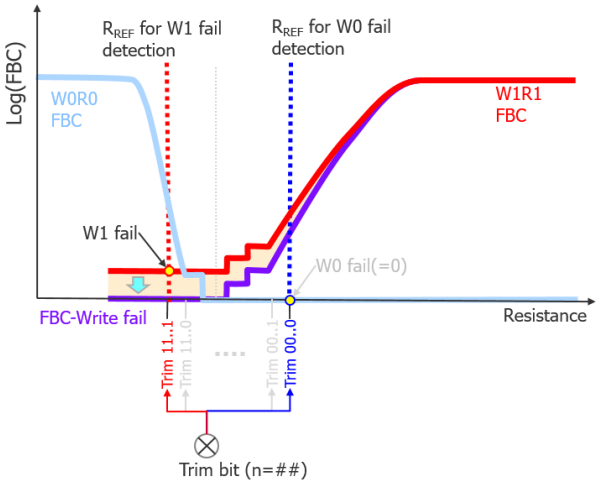

Fig. 5. Example Write-Fail-Screen.

During the Write-Fail-Screen test, the overall write fail count may reach a level beyond the correctable bit budget, which is the sum of repairable bits and ECC correction budget. In this case, the trim search test can terminate early, and the chip can be marked as defective. No further test needs to be performed for this chip since no combination of trim adjust and ECC/repair based correction can guarantee reliable read operation. The flow chart of the Write-Fail-Screen is shown below in Fig 6. The dotted lined box indicates the read operation at either extreme of the trim values, representing the best possible trim setting for read operation. As mentioned before, we called failing under these conditions as a write fail, and saved W1R1 test fail as W1 fail and W0R0 fail as W0 fail. The sum of the W1 fail count and the W0 fail count is compared against a saved correctable budget count to determine the early test termination.

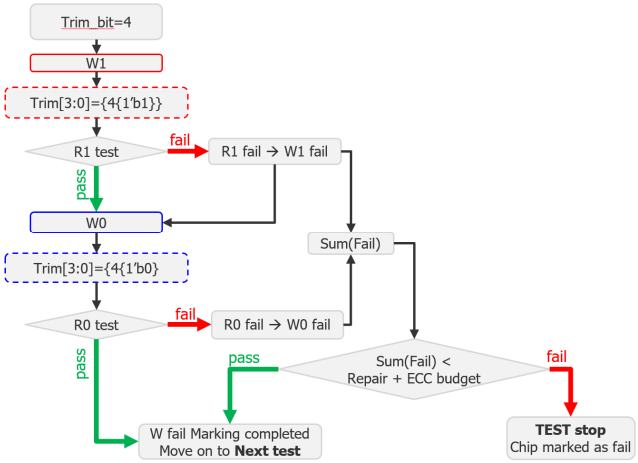

Fig. 6. Example diagram for Write-Fail-Screen.

IV. General Trim Setting Mode

The first-fail transient search method with the Write-Fail-Screen will give an optimum trim setting based on the distribution tail behavior of data 1 and 0. This is a good

method to set an independent trim setting for the individual chip because the reference value was customized for the weakest bit functionality. This reference value varies greatly chip by chip and is difficult to predict. However, the tail-end bit behaviors are based on rare bit properties and hard to capture unless the entire array is tested. For that reason, the first-fail transient search method is recommended to be used on small size memory arrays, which have limited repair solutions but capable to measure full array MBIST test several times for whole chips.

If a large memory array is used with a high repair / ECC budget, a few fails at the distribution tail ends are not a critical concern for the trim setting. Rather we should be more concerned about test time required to correct all the FBC from the large array. In this case, a generalized reference setting based on a sampling test would be a preferred solution to apply.

Sampling tests can be done in several ways. A test can be performed on sampled chips to set a single generalized trim setting for use in many untested chips in the wafer or lot. Or a test can be performed on sampled addresses on each chip to set the individual unique trim setting for each chip under reduced test time. The former one considers local variation information but it may be hard to capture global variations, which vary chip to chip, wafer to wafer, and lot to lot. The latter captures some of both, the local and global variations, and can roughly accommodate both variations for each chip. However, it would be expensive and inconvenient to test and set individual trim settings without an automated BIST based solution.

The first-fail transient search is not a suitable option for the general-trim setting. A distribution tail bit is a rare event. The behavior of those bits is unpredictable and different based on which array was tested. To find a general reference value, the reference value must be searched for more common events that follow generalized cell behavior. Trim value from a steep FBC slope is a good location to set a generalized trim setting. The slope of an FBC curve indicates how many more memory cells were failing compared to the neighboring trim resistance. The slope of the FBC will continue to increase until the reference resistance reaches the median of the distribution. Therefore, the higher slope, or cliff, of the FBC curve indicates that the tested trim value is close to the resistance value of the majority of the memory array. We need to set a fail boundary at the cliff of the FBC to set a generalized trim setting.

Direct FBC slope search is inconvenient because each slope data has to be calculated from two FBC data at neighboring reference settings. Also, we can't use the binary-judge-based-search method, rather we have to use the naïve linear trim scan. For simplicity and to use binary-judge-based-search we added a new input to the Trim Feedback Circuit, which we call "FSCR" (Fail Screen value). FSCR is an optional user input value which define the tolerable fail level. FSCR function will ignore the minor off normal distribution fail and search the boundary at the cliff of the FBC which based on normal distribution. The provided FSCR number will be stored in one of the registers in the new hardware component, Trim Feedback Circuit. The BIST will compare if the in-coming FBC is larger or smaller than FSCR and set the trim bit to "1" or "0" based on the comparison result. For R1 fail boundary search, the evaluating trim bit (MSB for the first test) should flip to "0" when the fail count is lower than FSCR to adjust the reference BL to have higher resistance for the next search. The evaluated trim bit can stay at "1" when the in-coming fail count is the same or more than FSCR. In this case, the next search test should use a lower reference resistance to find a lower fail point than the current. The trim boundary bits will be set from MSB to LSB until a final trim value is set where FBC is the smallest value exceeding FSCR.

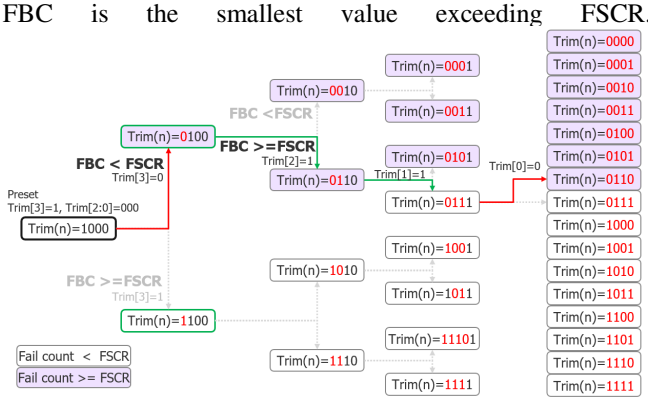

Fig. 7. The binary-judge-based-search for FSCR, n=4 trim

For the R0 test, the procedure will be similar except the final comparison is opposite because the fail count increases as the reference resistance gets lowered. An example of FSCR based binary-judge-based-search for the R1 case is illustrated in Fig. 7. The complete test must evaluate both R0 and R1 behavior. The block diagram of the generalized trim search method is shown in Fig 8. A User can use the search method with or without the Write-Fail-Screen mode option.

For the general-trim setting, FSCR should be set high enough to make sure the searched trim value is at the cliff of FBC and is not influenced by extrinsic fails such as a write fail. When we use a column or a row repair, massive fails in a single column or row can be fixed by a subsequent repair process and should not influence the reference setting. For example, a leaky BL or a BL with off-spec resistance will show massive fails with marginal parametric behavior on the specific BL. The FSCR value should be set high enough to cover the correctable bit budget to avoid the influence of those extrinsic fails during boundary search.

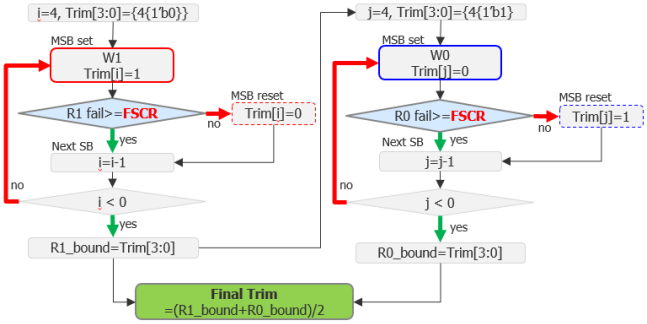

Fig. 8. Diagram for binary-judge-based-search for FSCR based trim setting

Fig. 9 compares the first-fail based search and the FSCR based search. When a first-fail transient search is used for the reference setting, the MBIST will search for the trim value for the first-fail beyond the write fail count (W1 / W0 fail count which is read at highest and lowest trim setting respectively). The final reference resistance setting will be set between two tails point named 'R0_tail' and 'R1_tail'. When a high FSCR value is used, our method will find two boundary trim values

on the FBC cliff of R1 and R0. The variation of sampling is smaller on the cliff than that of the tails because we read from higher population properties. The tail based search and the high FSCR based search may result in different reference settings for the same array. The first-fail transient based search accounts for more individual array properties. It will have less pre-repair read fail than a high FSCR based search. So it can work with a smaller repair and/or ECC budget.

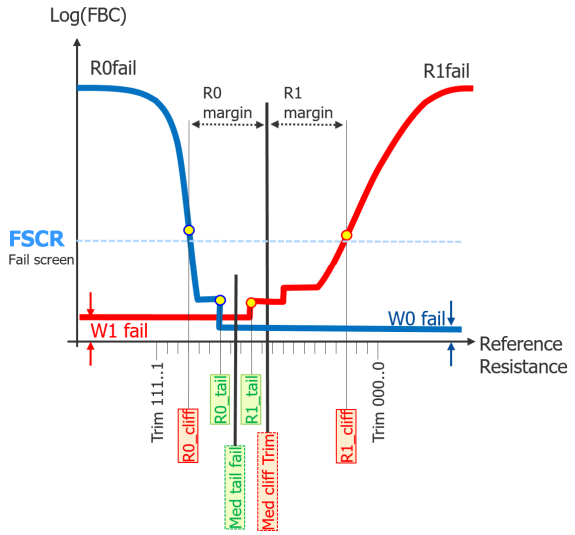

Fig. 9. Illustration of FBC relative to reference resistance setting.

Although the FSCR based trim setting has some extra fails compared to the first-fail transient search based setting, these fails however are within the correctable bit budget and expected to be fixed by repair and ECC. The FSCR based setting has other benefits. The FSCR based setting is a generalized trim setting, so we can use the result for the remainder of the memory array or for untested chips. This sampling reduces the overall test time and unwanted wear out of the memories.

In the current example, the final reference resistance value was set at the median of two boundaries of R1 and R0. This is to give an equal margin for R1 and R0 operation. However, when the sampling test is used on a small part of a memory array to judge the reference for the larger array, the FBC from the larger array will be spread wider and reduces the read margin for both R0 and R1. R_{AP} has inherently wider resistance variations relative to R_P. Add on to that, R_{AP} is more sensitive to temperature and read bias change [4][7][8]. Therefore, the tail of the R_{AP} distribution tends to extend more than that of R_P. As a result read margin for R_{AP} (separation between tails of R_{AP} to R_{REF}) will tends to get smaller relative to read margin for R_P. To re-balance the read margin, R_{REF} needs to shift towards R_P and the amount of shift depends on the temperature, the array size, the read voltage and the variance difference between R_P and R_{AP}. The user can calculate and compensate the shift by an external input, we call Trim_adj (not shown in the diagram).

FSCR can be set as low as "1". This turns the FSCR method into the first-fail transient search method. A low FSCR is good for trim setting for the individual device because the trim value would be customized for its specific weak bit functionality of individually tested arrays. As mentioned before, defective cells contributions can be measured by a functional test at the extreme trim condition and used to define an initial FSCR value. To save the test time of the defective cell pre-screening, one can pre-set the user-defined FSCR value high enough to screen any such expected fails.

V. BIST IMPLEMENTATION

All of the judgments are based on the number of fails at an operational reference trim setting. It is therefore required to store accumulated fail count data for each trim setting test. We are building a new Trim Feedback Circuit hardware component to operate in conjunction with the existing memory BIST engine and its components. At first, we describe the FBC accumulator, Fig. 10, and then move on to describe the entirely new hardware, shown in Fig. 11. The Trim Feedback Circuit implemented in the context of this paper is reasonable in size with about half the size of the Built-In Repair Analysis (BIRA) circuit deployed to compute a repair solution for hard faults. This means that the additional area required by the trim circuit is not a barrier to the adoption of this method. The relatively small size can be explained by the reuse of existing resources from the programmable MBIST controller and BIRA logic. Also, the main data path of the trim circuit is relatively narrow (5 bits in this example) and it does not change much with memory size.

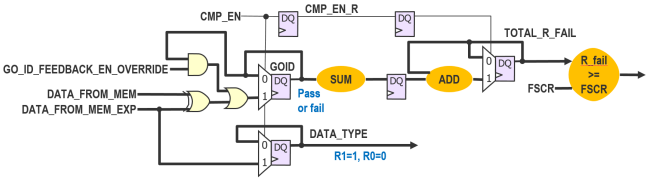

Fig. 10. Illustration of FBC accumulator.

The comparators, which compare the expected vs actual memory read values, and GOID registers, which save the result, already exist in the MBIST circuit. Each GOID bit indicates a fail event (mismatch between expected and actual) that occurred during a compare event (CMP_EN=1), see Fig 10. The state of the GOID registers is updated after each comparison event (GO_ID_FEEDBACK_EN_OVERRIDE=0). The number of fails of the compare event is added to the current total to accumulate a total fail count of the operation. The total size of the registers is proportional to the total number of bits of the memory macro. By default, it is set to accommodate errors on all bits. However, it is programmable to reduce the size so that only a subset of the memory bits can be tested during the sampling trimming procedure. The time complexity of the algorithm used for executing the trimming procedure of an array with five trim bits is 24N, where N is the number of memory locations that must be read. 4N of these 24N are attributed to the optional Write-fail-screen mode.

TABLE I. TRIM BIT UPDATE DURING THE TEST

	Test_initiate	incCountA	CountA	DATA_TYPE=1 (for R1 boundary)				DATA_TYPE=0 (for R0 boundary)			
				T3	T2	T1	T0	T3	T2	T1	T0
test initiate	1	0	0	1	0	0	0	0	1	1	1
during 1st test	0	0	0	1	0	0	0	0	1	1	1
1st test done	0	1	1	$R_{result1}$	1	0	0	$R_{result1}$	0	1	1
during 2nd test	0	0	1	$R_{result1}$	1	0	0	$R_{result1}$	0	1	1
2nd test done	0	1	2	$R_{result1}$	$R_{result2}$	1	0	$R_{result1}$	$R_{result2}$	0	1
3rd test done	0	1	3	$R_{result1}$	$R_{result2}$	$R_{result3}$	1	$R_{result1}$	$R_{result2}$	$R_{result3}$	0
4th test done	0	1	4	$R_{result1}$	$R_{result2}$	$R_{result3}$	$R_{result4}$	$R_{result1}$	$R_{result2}$	$R_{result3}$	$R_{result4}$

❖ Result# : #th comparison result of FSCR and FBC. 1, if fail count >= FSCR. 0, if fail count <FSCR

The full architectural view of the Trim Feedback Circuit is shown in Fig. 11. The accumulated fail count is compared to the FSCR value. The comparison result is then stored into the proper bit position of the R1 (R0) boundary trim register. The proper bit position is determined by the TrimMask register which shifts its data from MSB to LSB for each iteration of the binary search.

The data in the Final trim register will be delivered to the memory's reference trim port to update its current trim setting until the entire procedure completes. The register DATA_TYPE determines if the current search is for either R0 boundary or R1 boundary. The series of operations for the reference trim bit computation is shown in Table I. MSB trim bit "1" at the initial trim is shifting to LSB but the comparison result bit ($R_{esult\#}$) will stay at the tested trim bit position.

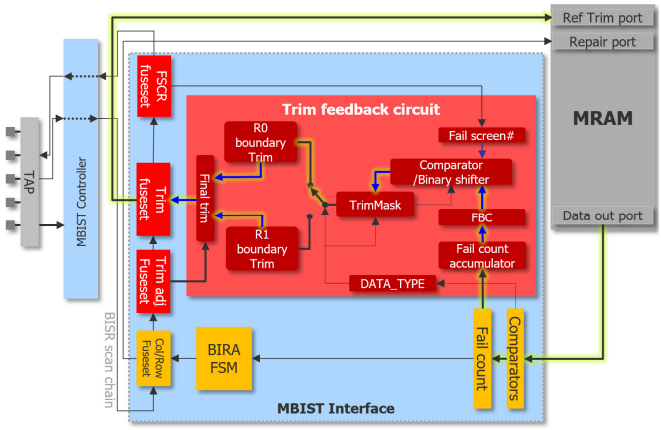

Fig. 11. Block diagram of Trim feedback loop

When both R1 and R0 boundary trim values have reached their individual LSB determination, the final trim register receives the calculated trim value between the R0 and R1 boundary values. The default calculation is averaging the two boundary values, but users may shift the value by external input, Trim_adj (not shown in Fig. 11). This optional setting will shift the final trim value for extra margin related to a sigma difference, array size, and temperature offset.

The goal of the trim adjust is to achieve post-process control of the reference resistance value to achieve maximum read margin for both "0" and "1". Due to different temperature sensitivities of the parallel and the anti-parallel state of an MTJ, the read margin is smaller at higher temperatures. Therefore, the reference resistance setting should consider resistance shifts at the highest operating temperature to make sure that the reference resistance works under the worst-case operation.

VI. Conclusion

In this paper, we described a fully automated on-chip trimming process leveraging existing memory BIST hardware. The MBIST circuit will collect FBC information of an array which will be used to judge two boundaries of a workable trim range. Each trim bits are determined bit-by-bit during the binary-judge-based-search. The search reduces the number of tests and improves the trim bit setting process. The Write-fail-screen mode is an added option to avoid the influence from extrinsic cell failures on the trim judgment process. It also provides the option of an early termination of the process in the case of fail rate beyond correctable bit budget.

The boundary will be set on the trim value where FBC exceeded the user input value FSCR. A low FSCR will set the boundary near the distribution tail of the FBC curve which is good for a small array with limited correction budget. A high FSCR setting will set the boundary at the cliff of the FBC curve where a much higher number of bits are failing. A high FSCR based trim setting is good for a general-trim setting on a large array, which has a high repair and ECC correction budget. In this case, test times of the full array could be considerable. The number of the test was reduced by the binary-judge-based-search and the sampling-test-based general-trim adjust. The introduced method will save large amounts of ATE test time without compromising the accuracy of the analysis.

References

[1] J. C. Slonczewski, Current-driven excitation of magnetic multilayers, J. Magn. Magn. Mater. 159, L1 (1996)

[2] W. F. Brinkman et al. "Tunneling conductance of asymmetrical barriers" Journal of Applied Physics Vol41. Num 5. p1915.

[3] Yi-Chun Shih et al., "Logic Process Compatible 40-nm 16-Mb, Embedded Perpendicular-MRAM With Hybrid-Resistance Reference, Sub-µA Sensing Resolution, and 17.5-nS Read Access Time", IEEE Journal of Solid-State Circuits, Vol. 54, No. 4, April 2019, pp.1029-1038.

[4] Artur Antonyan, Suksoo Pyo, Hyuntaek Jung, Taejoong Song, "Embedded MRAM Macro for eFlash Replacement," 2018 IEEE International Symposium on Circuits and Systems, pp. 529–551, May 2018.

[5] Kunal Korgaonkar, et al "To cache or to bypass? A fine balance in the emerging memory technology era" NVM 2019.

[6] V.B.Naik et al. "Superior Endurance Performance of 22-nm Embedded MRAM technology" IRPS 2C.2

[7] J.J.Kan "Systematic validation of 2x nm diameter perpendicular MTJ arrays and MgO barrier for sub-10 nm embedded STT-MRAM with practically unlimited endurance" 2016 IEDM 27.4.4

[8] Xiuyuan Bi. et al. "STT-RAM Cell Design Considering CMOS and MTJ Temperature Dependence" Trans on Magnetics Vol.48.No.11. 2012