MBIST Supported Multi Step Trim 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

Lori Schramm
Mentor, A Siemens Business
Atlanta, USA
lori_schramm@mentor.com

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

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

Khushal Gelda
ARM
Bangalore, India
khushal.gelda@arm.com

2020 IEEE International Test Conference (ITC) | 978-1-7281-9113-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/ITC44778.2020.9325218

***Abstract*—eMRAM (embedded Magnetoresistive Random Access Memory) has many attractive properties such as small size, fast operation speed, and good endurance. However, MRAM has a relatively small TMR (Tunneling Magnetoresistance) ratio, which means a small on-off state separation. It is a challenge to set an optimal reference resistance to reliably differentiate "1" and "0" states. Several trimming circuits were suggested in the literature to adjust a reference value and its search range. The trim setting can be controlled manually by user input; however, it consumes huge test time and requires off-chip engineering analysis to search and apply a trim setting for an individual memory array. In this paper, we will discuss the recent silicon results of fully automated trim process leveraging existing MBIST (Memory Built-in Self-Test) resources and new features to accommodate more complicated multi-step reference setting implementation through minor update of an existing MBIST circuit. The proposed MBIST solution uses a minimal number of tests to analyze massive array properties and automatically set complicated multi-step trim settings within a chip without the need for an external tester or manual adjustments.**

Keywords—MRAM, BIST, yield, reference trim, 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. The electron spin polarity of the layer switches based on the write-current direction. As a result, the MRAM cell can be switched to either a high resistance state (anti-parallel, R_{AP}) or a low resistance state (parallel, R_P) [1, 2]. During the Read operation, a sense amplifier latches a differential between the developed voltage or current level of the selected BL (bit line) relative to that of a reference BL. Therefore, the reference level (either resistance, voltage, or current) is one of the most important settings for reliable data sensing.

Due to process variations, On state and Off state memory has a certain range of resistance distribution similar to a normal distribution. An example of On and Off state resistance distribution of an MRAM array is illustrated in Fig. 1 (a). To read all memory data correctly, the reference resistance should be set between the distribution tails of the highest R_P and lowest R_{AP} values. The optimum reference setting is especially challenging in MRAM because the On and Off state resistance separation is relatively narrow compared to other types of resistive memories by an order of magnitude. The small separation of the on/off resistance makes reference setting even more sensitive to variations of resistances, which inevitably happen due to process shift, array configuration, total array size, temperature use case, etc. For that reason, using a fixed hardware preset R_{REF} for the entire array is quite risky to make large array applications. An incorrect reference setting leads to incorrectly interpreting the data, and it also limits the speed of read operation until a suitable differential level is achieved. The silicon measurement results summarized in section IV, allow you to visualize different optimum trim settings required for each individual array segment in the tested chips. To mitigate the reference setting issue, several trimming circuits were suggested in recent MRAM applications which allow post-process reference value control [3, 4]. The reference trim process allows for the update of reference values relative to accessed array properties and improve read margin for reliable sensing; thus ultimately improving yield.

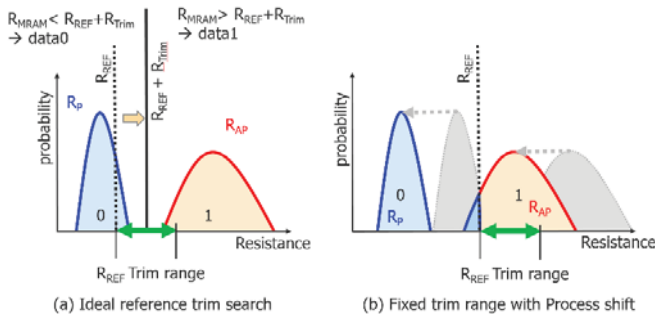


Fig. 1. Distribution of MRAM resistance for value "1" and "0" with sensing reference resistance setting

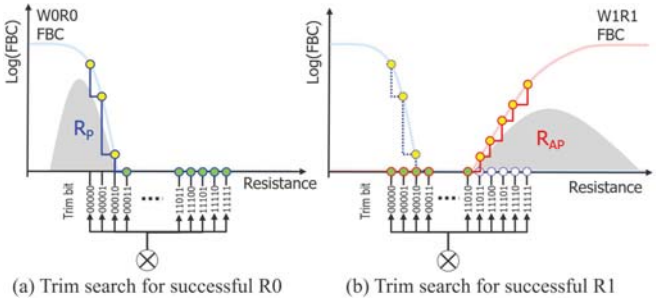
A trim search function is very efficient when it is executed as a part of MBIST. MBIST can test a chip without using expensive ATE (Automatic Test Equipment), and it is able to automatically evaluate the individual trim setting for each array based on a real-time measured test result. Although a trim search can be accomplished through external IO, collecting and analyzing the full set of memory data for each trim setting is a painful challenge. It requires an impractically large amount of

test resources to measure and analyze analog properties of each full array, including the resistance values of each cell in the "1" and "0" states. The functional test result can simplify the analog properties into binary pass/fail data. This simplified data analysis makes an MBIST-based trim search process even more effective.

In Section II, we briefly state our previous work about trim search procedures and various options to help search fidelity[5]. In section III, we will discuss the topic further with a new method to implement a multi-step reference setting adjustment from the MBIST circuitry. A multi-step reference search is needed when the optimal reference value is located outside the searchable trim range due to chip variation or process shift, etc. An example illustration of off-range optimal trim is shown in Fig. 1 (b). In this case, the optimum trim setting is not searchable within the range of available trims. Therefore, either the reference trim range or the memory resistance has to be adjusted so that the next step can detect the optimal trim search and setting. To understand memory distribution properties with smaller data set, we used FBC (fail bit count) information at each reference trim value. However, capturing a fail count collection for each individual trim setting for several ranges is still quite a lot of tests because it requires a series of tests for the entire memory array. In this paper, we are introducing a cost-effective test solution using a unique test sequence that can perform a multi-level trim search with minimum test insertion.

II. Reference Trim bit setting by MBIST

A memory functional test is performed by a series of Write and Read operations to a memory array. Output read data, "Q", will be compared with the previously written data by a comparator that gives a FAIL signal when the read data is unmatched to the expected value. The number of FAIL signal occurrences will be accumulated in a GOID register as the FBC of the tested array. The W0R0 (write0-read0) FBC will be zero when state "0" memory resistance (R_P) values for all memory bits are below the reference resistance value. In other words, when reference resistance gets lower, "0" state memory bits will have read fail start from the memory bits with the highest "0" state resistance.



(a) Trim search for successful R0 (b) Trim search for successful R1

Fig. 2. Illustration of FBC at different reference resistance settings

In the example of Fig. 2 (a), the trim settings 00011 to 11111 all have a reference resistance higher than the highest R_P resistance; so those trim settings are okay to use as the R0 reference. The FBC will increase when the reference value becomes lower than 00011, where some of the memories have an R_P resistance value higher than R_{REF}. Similarly, the FBC curve can be drawn for state "1" memory resistance (R_{AP}) vs. the reference trim setting as in Fig. 2 (b). In this example, we found the trim setting between 00011 to 11010 shows no-fail for both W0R0 and W1R1 (write1-read1) tests. We call this range of trim values the "safe zone". We will set the reference level in the middle of this zone to achieve the maximum success rate with the highest margin for both R1 and R0. The FBC curve follows the exact functional behavior of the array but with a much simpler data format. Therefore, optimal reference search analysis becomes much simpler than analyzing massive analog resistance values of the full array.

In reality, the FBC curve may not look as clean as we showed in Fig. 2. Some hard fails or circuit element fails will cause fails independent of the trim setting. A trim search test is done prior to a repair test, so the search result may contain some row/column fails. In some cases, this will cause complete removal of the safe zone in the search area and make search extremely difficult. In our previous work, we introduced an FSCR (Fail Screen) mode and a Write fail screen mode to overcome the issue by sorting out the unwanted fail behaviors from the FBC during the reference trim search[5]. To make the evaluation more practical, the number of tests has been reduced by eliminating unnecessary MBIST runs using a "binary-judge-based-search"

III. Trim Range Search

The search example shown in the previous section assumes the trim search range is correctly determined and the optimum trim setting is always searchable within the range. In the case of large arrays, it is extremely challenging to use a hardware preset trim range that always covers the entire safe zone between both tails of R_P and R_{AP}. A trim range that covers a partial safe zone will end up having a reference setting with reduced read margin, and a trim range with no overlap with a safe zone will always have fails for all available reference settings within the range. Similarly, if there is an intentional or unintentional process update, the optimal reference value for the new distribution may not be in the prefixed range of trim. Using more trim bits to expand the available reference range could be an alternative solution. However, adding more trim bits will require more search tests for each trim setting for an array, increase the data size to store for trim information, add a quality check for additional analog control step, and it may not be an adequate solution to address a significant distribution shift. Adding some quantized controllable current sources to the bitline is a good option to proportionally change the overall effect of trim behavior. Both trim resistance control with offset current control [3] and weighted additive current control on both active and reference bitline [4] offer two-step control of the trim setting. Although added control gives wider coverage of reference trimming, it increases the complexity of the search mechanism and required test time. In this section, we are suggesting a new search algorithm that can be used for these twostep trim settings with only a few selective evaluation tests.

The search algorithm will be separated into two separate steps. We first need to test the offset current setting or active bitline current setting prior to the reference trim bit search test to make sure the current range of trim covers the desired trim range. Since this first step test offers the range of trim, we refer

to this digitized control bit as “range bits”. The following reference trim bit search remains the same as we discussed in the previous study, which is also briefly discussed in Section II.

The suggested solution performs only a minimum number of evaluation tests to determine whether the selected trim range contains both tails of the distribution of the FBC curve or not. To detect the best range setting with a minimal test, we use a “FBC differential search”. The FBC differential (dFBC) is a fail bit count difference between two opposite data type tests performed at both edges of the trim range. It indicates how the trim range is overlapped with memory data distribution. When a trim range is overlapped more and more with memory data 1 distribution, reference to data 1 separation became smaller and cause more of read 1 fail while it will give more margin between reference and data 1 separation and increase the success rate for read 0. The dFBC is calculated by read 1 fail count at highest trim subtract from read 0 fail count at lowest trim. A ppositive dFBC indicates more overlap to data 0 and a negative dFBC indicates more overlap to data 1.

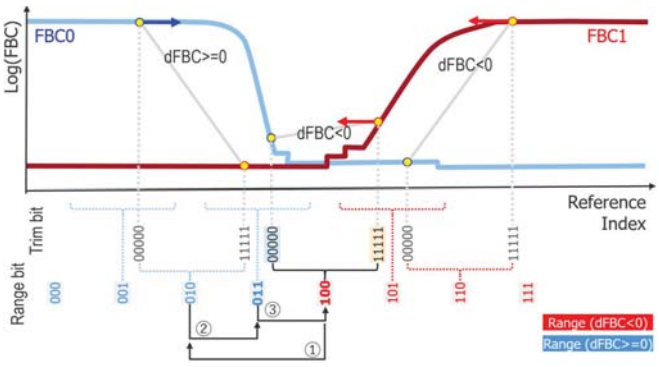


Fig. 3. An example range bit search, (① dFBC<0 next search left half ranges ② dFBC>=0 next search in right remainder ranges ③ dFBC>=0 compare and choose the smallest dFBC among Range011 vs Range100)

The differential FBC evaluation was combined with a binary search to reduce the number of tests. The binary search algorithm is similar to the one described in Section II, but it uses a differential FBC to judge each range bit and which range should be tested next. An example FBC curve and range bit evaluation is shown in Fig. 3. For each range bit setting, MBIST will compute the extreme R0 FBC (fail count at highest trim setting, in this example 00000) and extreme R1 FBC (fail count at lowest trim setting, 11111). MBIST will then compute the dFBC from these two extreme settings and set the examined range bit as 1 or 0, based on the polarity of dFBC. This process will be repeated from the MSB (most significant bit) to the LSB (least significant bit) sequence to judge which range bit setting is the best for the subsequent trim setting search. Fig. 3 illustrates the dFBC at each range bit setting. The range bits in Fig. 3 were colored red for a range bit with dFBC<0 and colored blue for a range bit with dFBC>=0. The binary search test for a range bit will start at the mid-value (in this example, the mid-value range bit is 100) and end at the transient range bit setting, where it changes its polarity of dFBC either to positive or to negative. These two dFBC trim ranges are the ones with their the dFBC value is closest to zero. A dFBC value close to zero means that the trim range setting has the most equally covered edge of data 1 and data 0 distribution and has a higher chance of containing a full safe zone. If one range bit to the other is tightly distributed with some overlap, both the smallest and second smallest dFBC would contain the entire safe zone. However, if each range bit is separated with small or no overlap of each range, the second smallest dFBC trim range may not contain some or all of the safe zone. This will cause misjudged reference settings later in the flow. To avoid that issue, we added extra analysis steps to evaluate which one of the two transient settings has a smaller differential fail count. This extra step is an analysis process based on data that has already been collected. Therefore, no additional memory test is required for the range search, and only the analysis process will be required by the MBIST circuit for a final decision.

For the case of trim control by additive current sources attached on both reference bitline and data bitline, the absolute value of reference current range is fixed by the trim current range on reference bitline and no additional control on reference bitline. However, the search would be similar if we treat the good active bitline trim current as the range bit and reference bitline current as a final trim setting. In Fig. 4, read 0 fail count is raised high on the right side of the graph because the x-axis is changed to current. This is the opposite placement of distribution compared to resistance-based distribution that we saw in Fig. 3. However, the cell current distribution still proportionally depends on the memory resistance distribution, so the active bitline trim search remains largely unchanged as a range search algorithm.

In order to find the optimal reference current, the good I_{REF} range should overlap both the low and high state of the I_{CELL} distribution tail. When I_{REF} does not cover both ends of the cell current distribution, as shown in Fig. 4 (a), we may not find the correct I_{REF} that sits in the middle of the “safe zone” (green arrow).

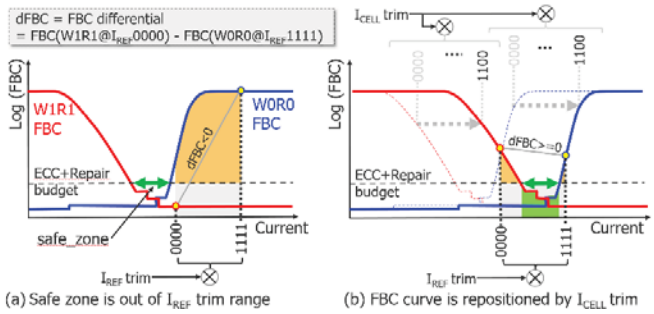


Fig. 4. I_{CELL} trim search by dFBC comparison

MBIST will evaluate the FBC difference between two extreme I_{REF} trims for each I_{CELL} trim bit to identify the optimum I_{CELL} trim condition. For example, dFBC is calculated as the FBC difference during W0R0 operation at I_{REF} trim 1111 and W1R1 operation at I_{REF} trim 0000. The dFBC-based search algorithm is very similar to what was showed in Fig 3. It will compare dFBCs to determine the I_{CELL} trim bit-by-bit from MSB to LSB until all I_{CELL} trim bits are defined. A I_{CELL} trim bit will be set to 1 (0) when its dFBC<0 (dFBC>=0). Once all I_{CELL} trim bits are settled, the trim setting has converged to where the polarity of dFBC changes.

The simplified flow for differential fail count based binary search is shown in Fig. 5. MBIST will search for the optimum range bit, where dFBC changes from positive to negative. In the example of Fig. 5, the transition occurs between range 011 and 100. MBIST will compare dFBC of these two transients to pick the range bit with the smallest differential as the best range bit setting. For this comparison, the measured dFBCs at previously reviewed range settings will be temporarily saved in registers. The search will converge to the range bit with the smallest dFBC or a dFBC close to zero, when fail counts at W0R0 at trim00000 and W1R1 at trim11111 are similar. This range bit captures the most balanced R1 and R0 fails among all available range bit settings. An example case with 3 bits range and 5 bits trim, requires 512 full array tests ($2^{(3+5)}$x2) to perform the full FBC curve based evaluation. However, the suggested "FBC differential search" and the "Binary-judge-based-search" can do the job with only 16 ((3+5)x2) full array test, which is only 3% of the test time of a full FBC-based evaluation. Because of the cumulative nature of the FBC, we do not lose search accuracy by doing this type of binary search.

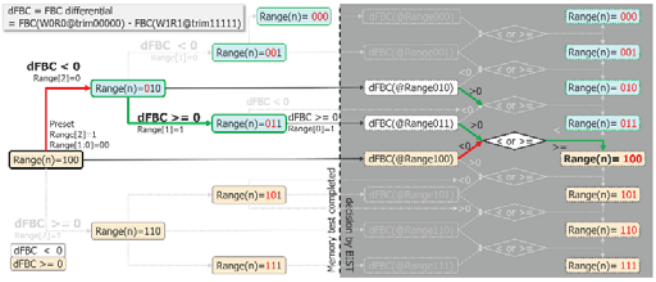


Fig. 5. Range bit search based on FBC differential at each extreme trim

IV. MBIST IMPLEMENTATION AND VALIDATION

The trimming hardware components were added to operate in conjunction with the existing MBIST engine. The full architectural view of the trim feedback circuit is shown in Fig. 6. All of the judgments are made from the number of failed operations at the tested reference trim setting. Therefore, it is required to store the accumulated fail count at each trim setting. Fail count information is collected through comparators which evaluate expected vs. actual memory cell content and generate the GOID signal whenever the result was not matched. Each GOID bit indicates a fail event that occurred during a compare event (CMP_EN=1). This process is already embedded as a part of the existing MBIST solution. The number of fails at the compare event is added to the current total in order to accumulate a total fail count of the operation. An accumulated fail count is then delivered to the trim range feedback circuit saved in a register for both edges of the trim range setting. The comparison result will be used to make a decision for each range bit setting from MSB to LSB. Once the complete range bit search process is done, the final trim range bit will fixed to the array and initiate trim feedback circuit to start searching for optimum trim value within the trim range. In this circuit, each accumulated fail count will be compared with the FSCR value. The comparison result is used to judge each trim bit and update the trim register value. The proper bit position is determined by the "TrimMask" register, which shifts its data from MSB to LSB at each test iteration. When both R1 and R0 boundary trim values have reached their individual LSB determination, this will trigger the final trim register to receive the calculated trim value between the R0 and R1 boundary values. The default calculation averages two boundary values which maximize the read margin for data "1" and "0" for the test condition.

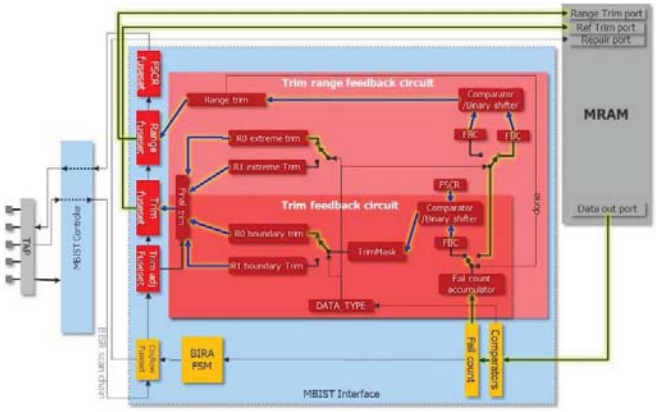


Fig. 6. Block diagram of trim feedback loop includes trim range bit search

Users may shift the computed trim value through an external input Trim_adjust (not shown in Fig. 6). This allows a bias of the final trim value based on other consideration factors such as a sigma difference, array size, or temperature use case, etc. Due to the different temperature sensitivity of the parallel and the anti-parallel state of MRAMs, the "safe zone" will shift with the temperature change. The reference resistance will also be shifted accordingly, following the temperature behavior, because the reference resistance is also set by one of the paired memory bits; and it will shift as R_P and R_{AP} shift [6-7]. However, the read margin may become smaller at higher temperatures due to the extra tail extended from some of the weak bits. Users may utilize the Trim_adjust option to offset the computed trim value from a mid-point of the R0 and R1 boundary to a different position, based on their data to make sure that the reference resistance works under the worst-case operational conditions. For example, if the trim search test was done at room temperature only to save test cost, users may have to shift the final trim value more towards the R0 boundary to compensate for extended R_{AP} tail at higher temperatures.

The trim feedback circuit functionality was emulated by Verilog simulation. The memory simulation model had to be improved in order to introduce a variable number of read faults as a function of the trim value applied by the MBIST circuit. At the beginning of the simulation, a number of memory cells are randomly selected to be faulty when reading a "0" and a different number when reading a "1". The number of faults of each type is selected to reflect a typical distribution of read faults. The model automatically applies the correct distribution based on the value of the majority of the cells being read during a clock cycle. This fault injection mechanism is independent of the conventional one used to inject hard faults during the verification of the diagnosis and repair features of the BISR circuit. This way, the entire manufacturing process can be simulated using a single verification pattern to perform trimming of the reference resistance, repair analysis, fuse

programming, and validation of the repair solution. Fig. 7 shows the portion of the simulation identifying the R1 and R0 boundaries. In this example, trim values vary from 0 to 31 encoded with five bits. The binary search of each boundary then requires six iterations, where the first iteration is used to count the number of hard faults. The first iteration for data value 1 (0) is identified by "WF1 (WF0) screen" in the figure. These faults are subtracted from the total number of faults found in the subsequent iterations as trimming cannot fix them. Hard faults are addressed either during repair using spare rows/columns or using ECC. The progression of the trim values during the binary search is shown by signal SATRIM. Here, R1_BOUND=23 and R0_BOUND=4 for a final trim value of '13'.

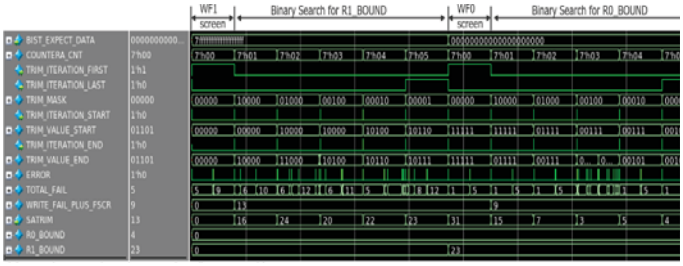


Fig. 7. Verilog simulation run result for trim analysis

Fig. 8 shows the silicon result of the trimming process. Each row in the table indicates an array of the tested die with 8Mb granularity, except for the 1Mb instance in the first row; and each column indicates different samples. Each tested array is colored based on its self-trimmed value, with white as the lowest trim setting and red as the highest trim. The color scale allows us to visualize the spread of trim values across the dies and even across the arrays in a die. It is therefore very inefficient to manually search and set the trim value. An on-chip solution is a great option for memories using the smaller granularity of the trim (more trim bits) and/or for large size arrays where diversity is even more of a concern.

Fig. 8. Die-to-Die self-trimming setting color maps

The trim circuit implemented in the context of this paper was reasonable, i.e. about half the size of the Built-In Repair Analysis (BIRA) circuit employed to repair 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 BIST controller and BIRA logic. Also, the main data path of the trim circuit is relatively narrow (five bits in this example), and it does not change much with memory size. The largest registers storing the total number of failures and the number of hard faults is currently 16 bits wide, but it can be reduced in the future to accommodate additional logic needed to identify the trim range values. The block diagram of Fig. 6 shows the components of this new capability to be implemented in the future. Many of these components are actually shared with the trim feedback circuitry.

The trim feedback circuit itself can be shared to reduce the area cost further. However, the test time will be increased due to the serial execution of the 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. Here, 4N of this operation was used for an optional Write-fail-screen mode. In the worst-case, where all locations must be read, the MBIST circuit allows reading a subset of locations distributed all over the array to speed up the procedure. The size of the subset is fully programmable before starting the MBIST controller.

V. Conclusion

Based on the silicon results shown in Fig. 8, a wide selection of optimum trimming settings are used even for array segments in a single chip. However, it would be extremely expensive and inconvenient if we manually search and set each individual array segment for a large array. Complexity is even greater if we consider to search trim ranges or multiple trim adjustments on different inputs. In this paper, we reviewed a fully automated on-chip trimming process leveraging existing memory BIST hardware to search multiple trim setting processes. A trim range search was performed prior to the reference trim search. Before we initiated the optimal trim search, a trim range search was performed. The search reduces the number of tests and improves the trim bit setting process. The suggested "FBC differential search" based range bit search combined with the "Binary-judge-based-search" saved 97% of test time relative to full trim FBC curve review on the example test case we used. The introduced test method will save the test budget 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] Artur Antonyan et al. "Embedded MRAM Macro for eFlash Replacement," 2018 IEEE International Symposium on Circuits and Systems, pp. 529–551, May 2018.

[4] 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.

[5] Jongsin Yun et al., "MBIST Support for Reliable eMRAM Sensing" ETS 2020.

[6] El Mehdi Boujamaa et al. "A 14.7Mb/mm2 28nm FDSOI STT-MRAM with Current Starved Read Path, 52Ω/Sigma Offset Voltage Sense Amplifier and Fully Trimmable CTAT Reference" VLSI 2020 CM2.1

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