# Bias programming in TLC Flash cells

## Introduction

Flash-based storage has become the standard storage in computing devices today. It has become popular thanks to its fast-random access, light weight, low energy consumption, and increasingly large capacity. However, flash memories have a much shorter lifespan than other storage devices. Furthermore, to modify a single page in a block the entire block must be erased and programmed over again. Each such program and erase (P/E) cycle wears the pages and increases their bit error rate (BER), up to the point where they can no longer be used reliably. A flash chip’s lifetime is measured by the number of P/E cycles its blocks can endure before reaching the critical BER level.

There are several approaches to extend the lifespan of an SSD device. One of the approaches is primarily done by protecting the data written in each flash page by an error correction code (ECC). The redundancy bits required for this code are stored in the page’s spare area. The number of bit errors the code can correct can be increased by increasing the size of the spare area and by employing stronger and more efficient codes, such as BCH and LDPC. In general, the spare-area size is determined by the expected BER at the end of the chip’s lifetime. As a result, the ECC is stronger than what is necessary when the chip is still ‘young’. Previous studies proposed to implement a weaker ECC in the beginning of the chip’s lifetime to reduce flash read and write latencies. Alternatively, the unused bits in the spare area can be leveraged to implement specialized codes that allow rewriting flash pages without erasing them first.

Previous studies correlate the bit values programmed on the flash cells with the extent of their wear, a phenomenon termed content-dependent memory damage. In a SLC chip each cell represents ‘1’ bit from 1 page. The ‘0’ is represented by a high voltage level while ‘1’ by a low voltage level. Therefore, increasing the number of ‘1’ reduces the average voltage level sustained by the flash cells, which has been shown to reduce their wear and increase their lifespan. Jagmohan et. al suggested to leverage this property for increasing SSD lifespan by employing biased programming. They proposed to encode the data written to the flash pages with enumerative (or endurance) codes, whose output is biased (shaped) - it includes either more ones or more zeroes.

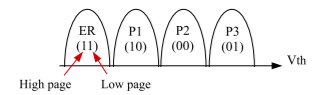


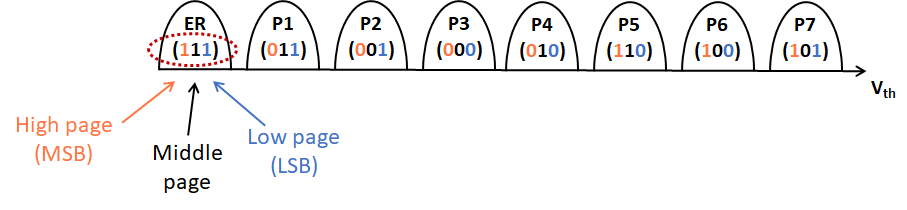
Figure 1: Standard mapping of voltage levels to bit values in MLC flash chips.

In a previous study they accommodated the extra overhead in unused bits of the page spare area. The benefit of this approach is threefold:

1. Does not depend on the compressibility (or other properties) of the incoming data.
2. Does not incur any storage or computational overhead
3. Can be fully implemented within the flash controller, at the chip level.

Thus, it doesn’t affect other device-level optimizations. The applicability of our biased programming approach depends on the flash chip characteristics: the size of the page spare area and the cells’ sensitivity to their voltage level. The latter is strongly influenced by proprietary flash-level optimizations applied by manufacturers for minimizing program and read disturbance. Nevertheless, our initial results demonstrate that biased programming can be applied successfully even without detailed information about these optimizations.

In that previous study this approach was successfully tested on two different MLC chips  designs that showed an increased flash lifespan by 24.17%. In this study we implement the same method on a TLC chip with the following voltage scheme:



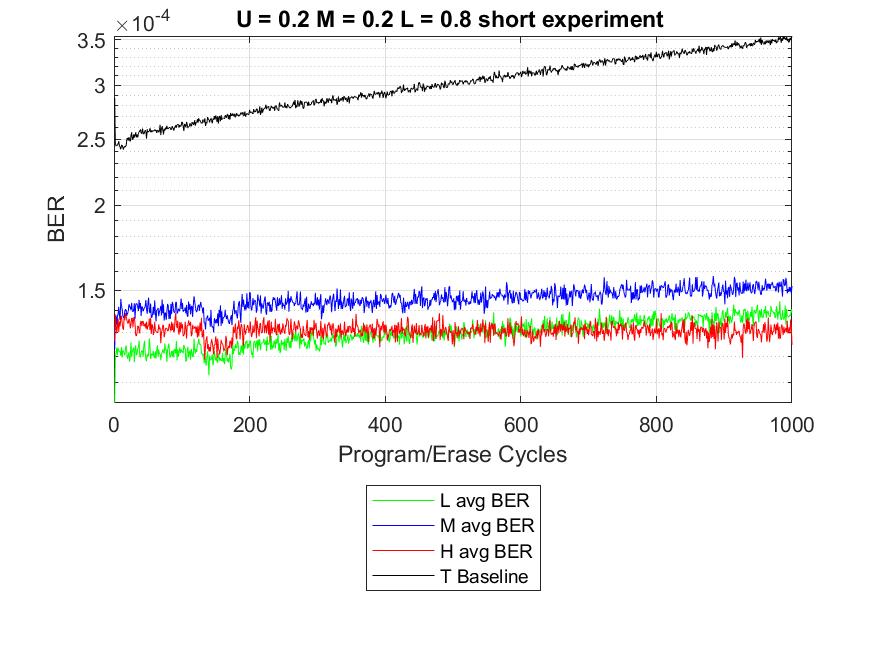
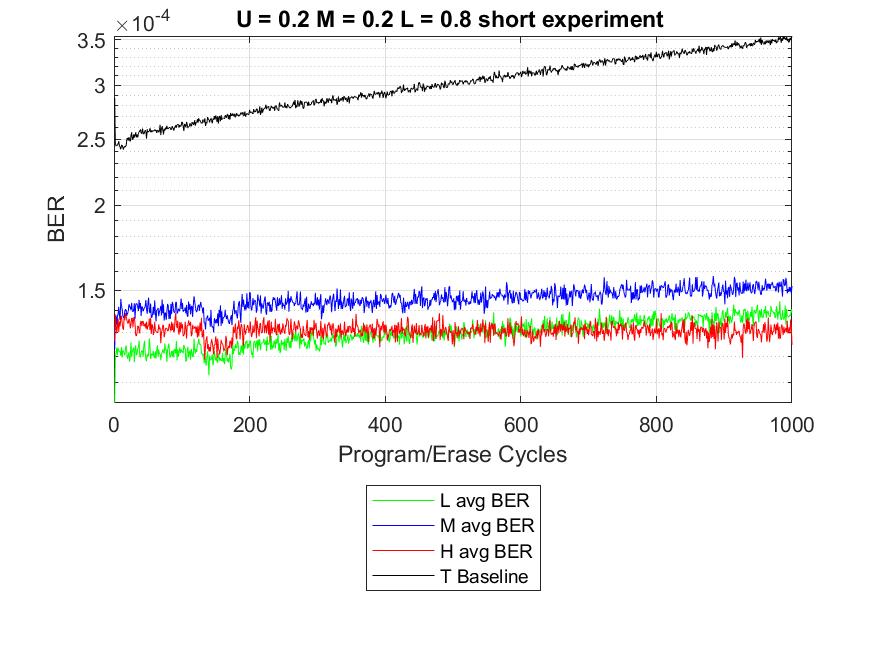
Each cell in the TLC can be one of 8 different values, which corresponds to 8 different voltage level sections, for each value. This means that the voltage range, which was previously split into 2 or 4 sections, is now split into 8, and consequently is more susceptible to errors.

While in 2 or 4 state chips (SLC & MLC) the effect of higher levels of voltage corresponds to higher levels of ware, this isn’t necessarily the case with 8 state chips (TLC) since now the BER is affected by any flakiness of the midrange voltage states behavior.

## Balancing between entropy & BIAS

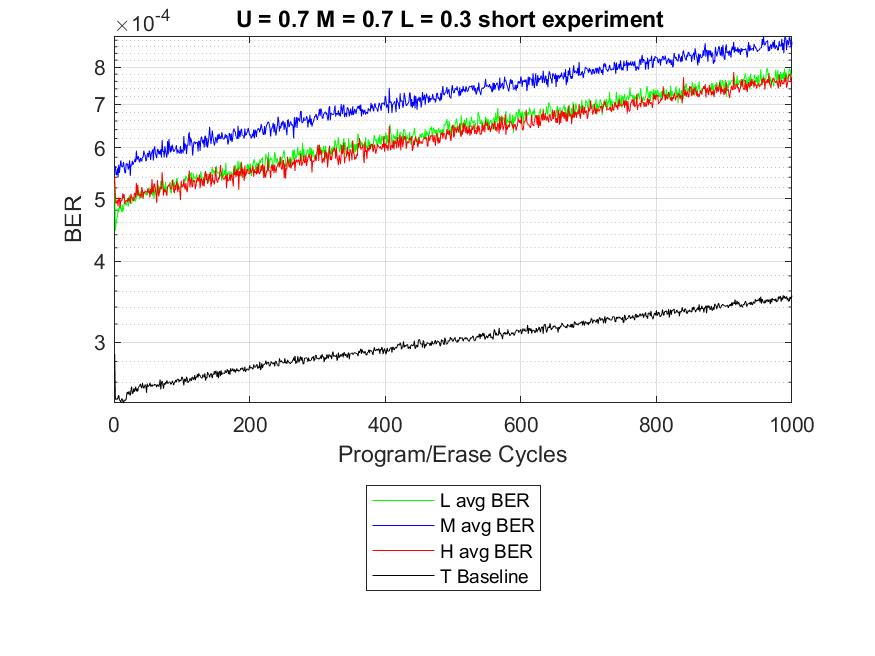
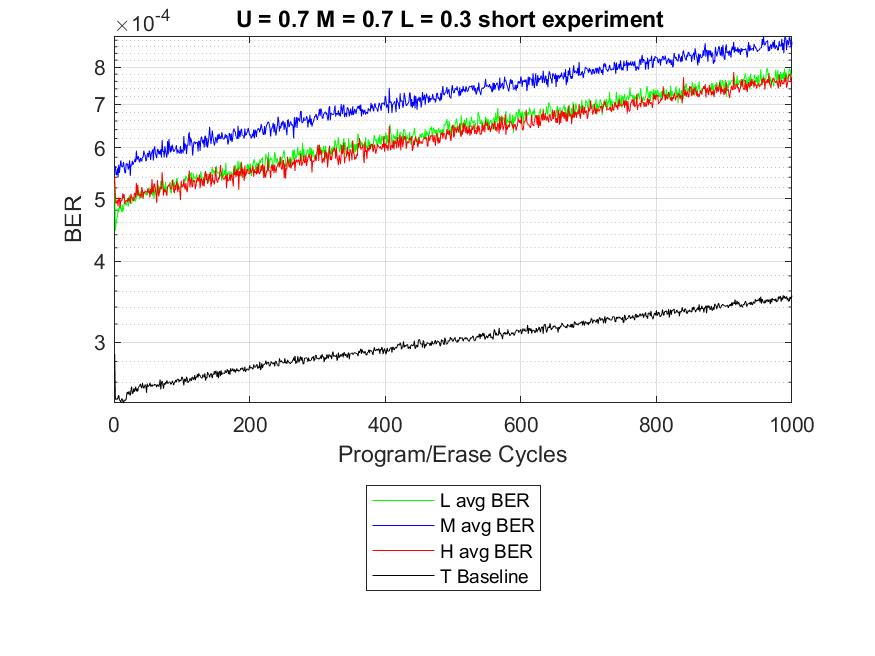
But there is a downside to reducing the Bias level in some states and increasing in others which has to do with the entropy of the data. In other words, the more we increase one state over the other the more we limit the data that can be stored onto the device and reduce its storage capacity.

## First shaping trial - Bias per page

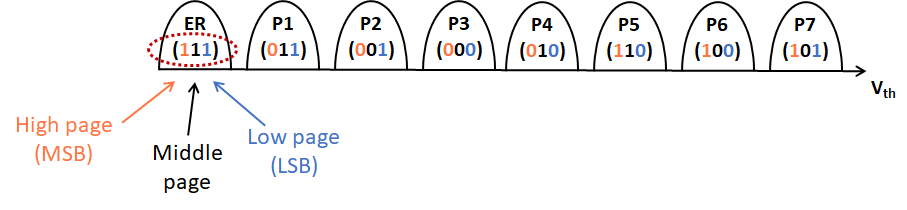
At first, we tried shaping the data using bias per page. We ran a series of short tests with different bias values for the lower, middle & upper page. The value of each page defines the percentage of zero in the data. The most successful tests are presented in the table below. And the best overall in the graph:

|  |  |  |
| --- | --- | --- |
| **Upper** | **Middle** | **Lower** |
| 0.2 | 0.2 | 0.8 |
| 0.3 | 0.3 | 0.7 |
| 0.4 | 0.4 | 0.6 |
| 0.25 | 0.25 | 0.75 |
| 0.3 | 0.3 | 0.3 |
| 0.4 | 0.4 | 0.6 |
| 0.35 | 0.35 | 0.65 |

The tests we ran with the lowest BER had low percentage of zero in the upper & middle pages and a high bias in the lower page. And the worst tests were characterized with the exact opposite values. For example:



But, in complex voltage schemes, such as in our TLC, you can't control the bias programming of each state with this method. For example, state 0, and state 7, which represent the lowest and highest levels, are different only in their middle page bit. To achieve the optimal bias combination, we need to use a different method of bias programming.



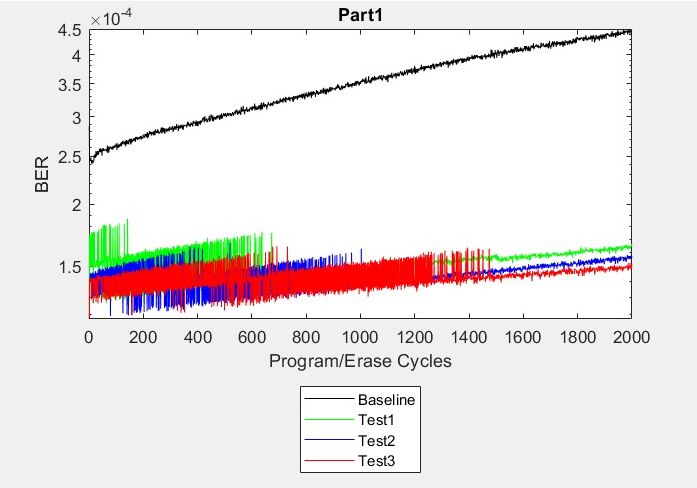
## Focus on Bias by state and not by page – “more power to the State”

In this study we took a different approach, by setting the bias according to the state. This means we have better control on the bias programming and can optimize the shaping of the data fully. We started by validating a previous study using this method.

## Assumption of previous tests

We were assigned three configurations to test with the following biases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| State | Bits | Test 1 Prob | Test 2 Prob | Test 3 Prob |
| 0 | 111 | 0.2 | 0.225 | 0.225 |
| 1 | 011 | 0.2 | 0.175 | 0.2 |
| 2 | 001 | 0.15 | 0.175 | 0.175 |
| 3 | 000 | 0.15 | 0.125 | 0.15 |
| 4 | 010 | 0.10 | 0.125 | 0.1 |
| 5 | 110 | 0.10 | 0.075 | 0.075 |
| 6 | 100 | 0.05 | 0.075 | 0.05 |
| 7 | 101 | 0.05 | 0.025 | 0.025 |
| **entropy** |  | **2.85** | **2.808** | **2.761** |



The assumptions of the tests were that, as in other cell architectures (SLC, MLC), The higher the voltage of a state the lower it’s bias probability should be in order to achieve reducing the cell’s wear, while trying to maintain a relevant entropy. There is a clear tradeoff here, the less wear on the chip the worse the entropy.

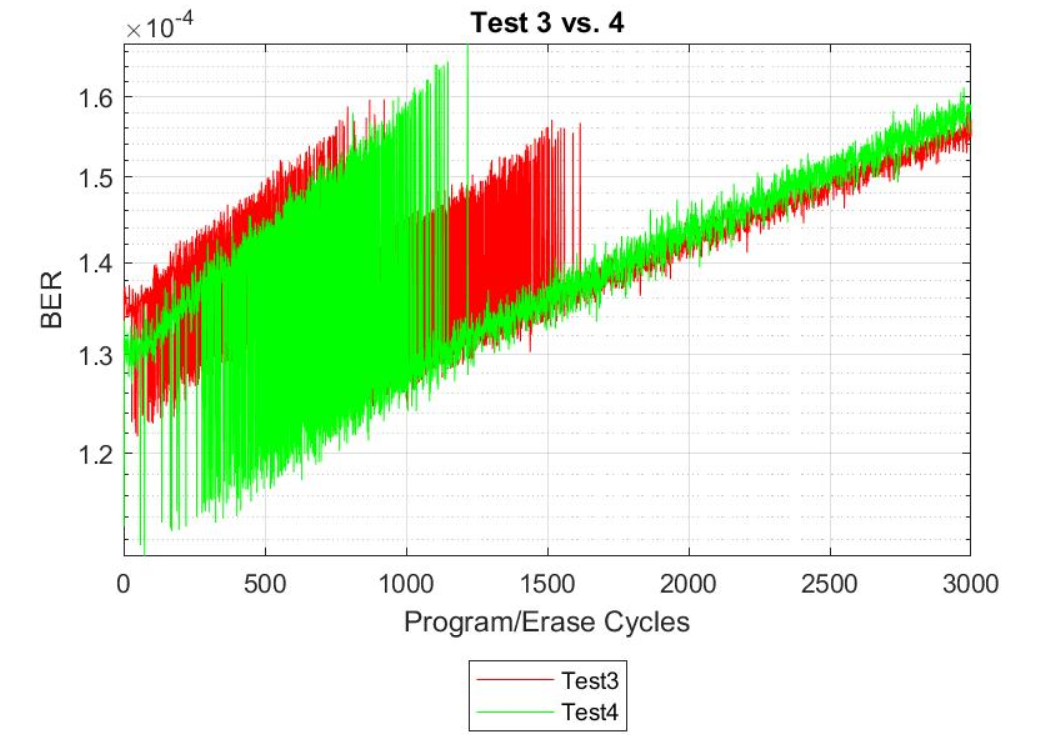
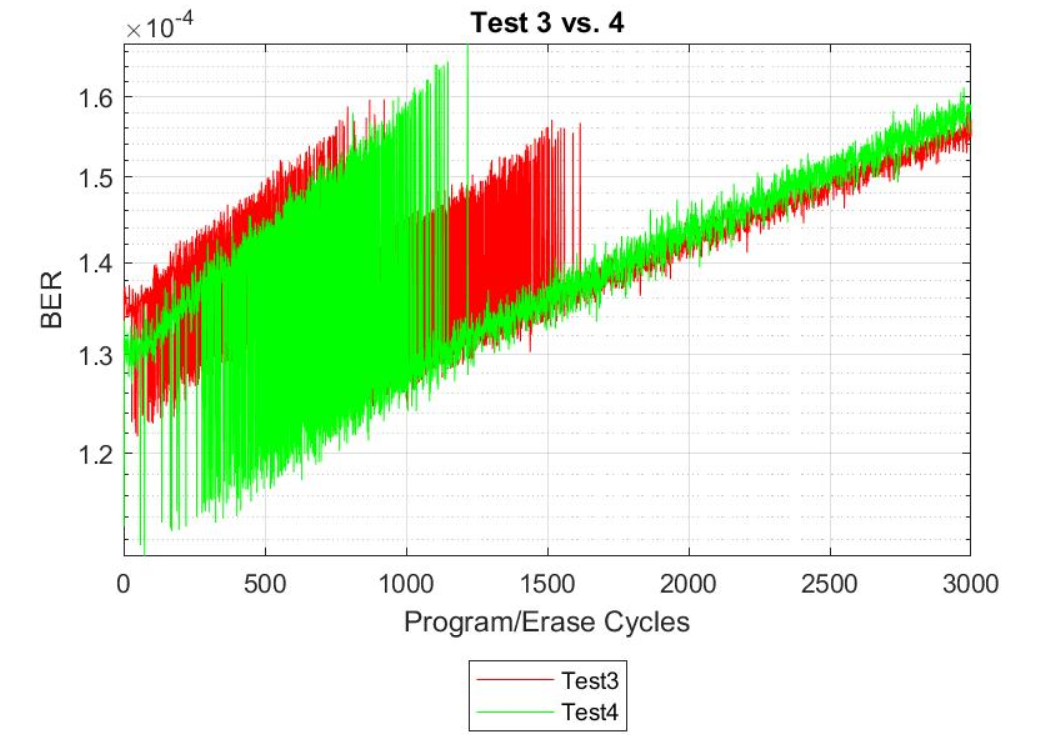
## Challenging the approach

To try and improve on the previous tests we first looked at the number of errors per mode and compared to the errors found in the Baseline:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| baseline | | | test3 | | |  |
| s1 | s2 | e | s1 | s2 | errors | diff |
| 5 | 4 | 3123423 | 5 | 4 | 7457347 | -4333924 |
| 3 | 2 | 686385 | 3 | 2 | 3189950 | -2503565 |
| 4 | 3 | 1231573 | 4 | 3 | 2440979 | -1209406 |
| 7 | 6 | 7134659 | 7 | 6 | 8155547 | -1020888 |
| 2 | 1 | 523218 | 2 | 1 | 1538071 | -1014853 |
| 6 | 5 | 1985982 | 6 | 5 | 2439775 | -453793 |
| 1 | 0 | 198094 | 1 | 0 | 469383 | -271289 |

We checked what states in test3 had an increase in the number of errors compared to the Baseline. We found that states 3 & 5 had the most significant increase. Therefore, we tweaked the bias numbers accordingly and arrived at test 4:

|  |  |  |
| --- | --- | --- |
| state | test3 | test4 |
| 0 | 0.225 | 0.175 |
| 1 | 0.2 | 0.225 |
| 2 | 0.175 | 0.2 |
| 3 | 0.15 | 0.125 |
| 4 | 0.1 | 0.125 |
| 5 | 0.075 | 0.05 |
| 6 | 0.05 | 0.075 |
| 7 | 0.025 | 0.025 |



Red – decrease in bias compared to test 3

Green – increase in bias compared to test 3

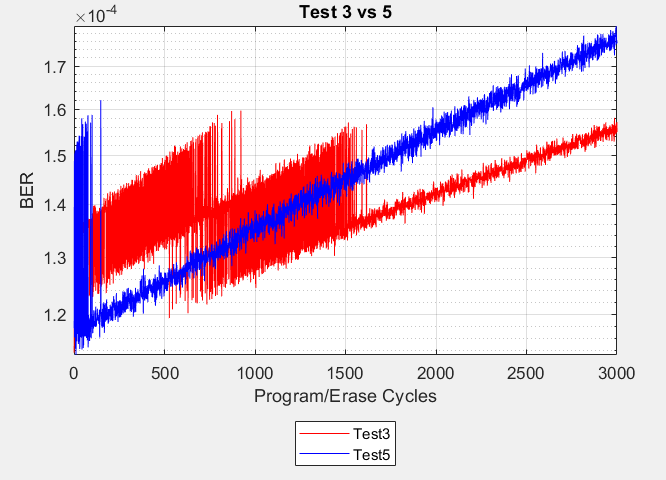
But test 4 had a higher error rate that test3 in the long run.

This only justified the assumption that increasing the bias on a high level of voltage as 6 would increase the wear of the chip. So again, we viewed the states with the largest diff in error numbers between test3 & test4:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test3 | | | Test4 | | |  |
| s1 | s2 | errors | s1 | s2 | e | diff |
| 4 | 5 | 2226947 | 4 | 5 | 3679791 | -1452844 |
| 2 | 3 | 1261783 | 2 | 3 | 1945263 | -683480 |
| 1 | 2 | 508532 | 1 | 2 | 943956 | -435424 |
| 5 | 4 | 7457347 | 5 | 4 | 5989268 | 1468079 |
| 0 | 1 | 4524725 | 0 | 1 | 3285201 | 1239524 |
| 3 | 2 | 3189950 | 3 | 2 | 2379709 | 810241 |

Even though we received a reduction in the expected states it was smaller than the increase in other states and has a worse wear rate. Since the bias is different now between each state, we had to find a more intelligent way to compare each state and analyze the effect each state could have on its neighbors.

So, to understand the results better we created a graph of the BER per state:

Strangely enough we saw an increase in the BER in state 5 even thought the Bias was dropped on that state. So, we decided to run an additional test while doubling the bias on state 5 & 6 and reducing on stats 2 & 3 so to create an extreme scenario so to find the correct graph to analyze the results. This is how we arrive at test 5:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **State** | **Baseline** | **Test3** | **Test4** | **Test5** |
| 0 | 0.125 | 0.225 | 0.175 | 0.225 |
| 1 | 0.125 | 0.2 | 0.225 | 0.2 |
| 2 | 0.125 | 0.175 | 0.2 | 0.15 |
| 3 | 0.125 | 0.15 | 0.125 | 0.075 |
| 4 | 0.125 | 0.1 | 0.125 | 0.1 |
| 5 | 0.125 | 0.075 | 0.05 | 0.15 |
| 6 | 0.125 | 0.05 | 0.075 | 0.075 |
| 7 | 0.125 | 0.025 | 0.025 | 0.025 |

The results of test 5 were much worse than test4. We generated the BER graph but saw that the BER on state 5 has decreased even though we double the bias.

It was clear a different graph is needed. We noticed that we only multiplied the denominator with the bias while also the errors were affected by the bias chosen. So, we generated a new graph displaying the true BER per state:

You can see straight away that the BER in reversed on state 5 and we see why we received an increase in test5 compared to test 4 & test 3.

Let’s look at the graph again while removing the Baseline to see more clearly the differences between the tests:

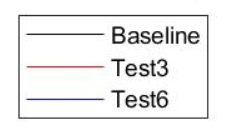
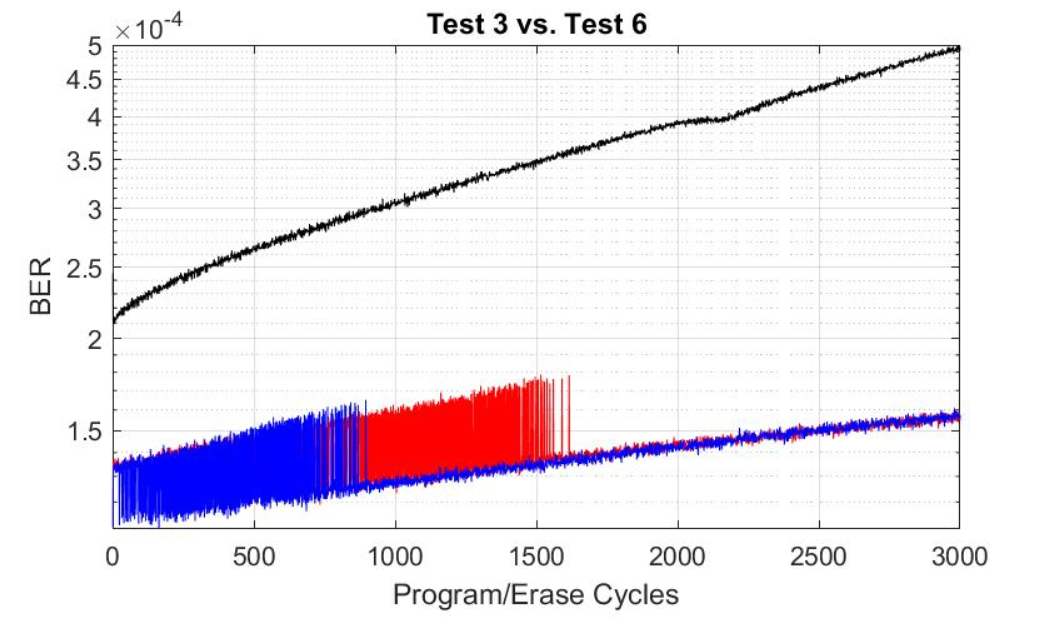
Since this test wasn’t successful, we decided to revisit our new approach by comparing the bias levels to our ‘BER per state’ graph to understand the impact of each bias number to the chips wear.

## Rethinking the new approach – test6

The assumption of test 3 was just to take a linear bias with a decrease of 0.025 between each state and a drop of 0.05 between state 3 & 4. **The assumption is that the higher voltage affects the BER of the TLC chip in an ascending order.** Comparing the bias graph to the BER we can see that the V curve between states 7 to 5 (seen in yellow in the graph below) is ignored completely by the bias in test3 (and test 1 & 2) as seen in the “Test3 Bias” graph. Assuming state 5 & 6 have a similar wear affect it might make sense. But if the ware affect isn’t according to an ascending order, we propose a test like test 3 with only a “V curve” between states 5 to 7 biases.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **State** | **Baseline** | **Test3** | **Test4** | **Test5** | **Test6** |
| 0 | 0.125 | 0.225 | 0.175 | 0.225 | 0.225 |
| 1 | 0.125 | 0.2 | 0.225 | 0.2 | 0.2 |
| 2 | 0.125 | 0.175 | 0.2 | 0.15 | 0.175 |
| 3 | 0.125 | 0.15 | 0.125 | 0.075 | 0.15 |
| 4 | 0.125 | 0.1 | 0.125 | 0.1 | 0.1 |
| 5 | 0.125 | 0.075 | 0.05 | 0.15 | 0.05 |
| 6 | 0.125 | 0.05 | 0.075 | 0.075 | 0.075 |
| 7 | 0.125 | 0.025 | 0.025 | 0.025 | 0.025 |

Furthermore, state 0 has a higher Ber than the other tests which could be another option to optimize test 3 since test 5 has the same bias on stages 0, 1 but the other bias values manage to reduce the BER on state 0 significantly.

the results of test 6 were very similar to test 3 in the number of errors and the wear rate. 

The “BER per state” graph shows the large improvement in state 5 & 0 between test 6 vs. test 3:

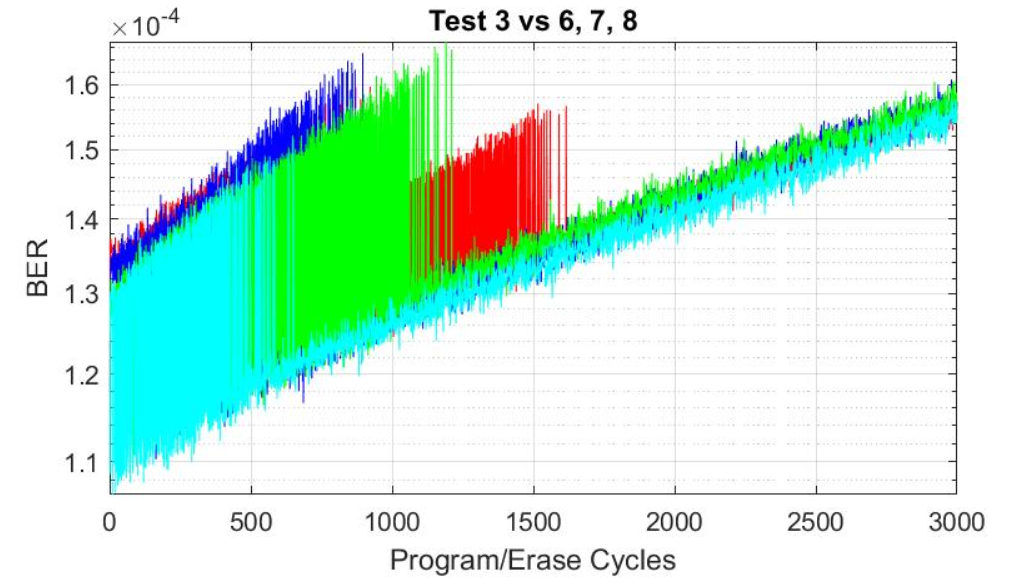
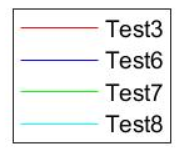
|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Baseline** | **Test3** | **Test6** |
| 0 | 0.125 | 0.225 | 0.225 |
| 1 | 0.125 | 0.2 | 0.2 |
| 2 | 0.125 | 0.175 | 0.175 |
| 3 | 0.125 | 0.15 | 0.15 |
| 4 | 0.125 | 0.1 | 0.1 |
| 5 | 0.125 | **0.075** | **0.05** |
| 6 | 0.125 | **0.05** | **0.075** |
| 7 | 0.125 | 0.025 | 0.025 |

The mothed now was working. To continue this thought further we noticed the effect of test 6 on state 0 doesn’t exploit the “v curve” on the low end of the voltage which brings us to a series of tests:

## Applying the same approach to the lower voltage states – “same same, different voltage”

Keeping the same biases as test 6, we permutated the biases of states 0-2 to test the effect of different bias levels on the BER levels of the chip. We came up with the following tests 7 & 8:

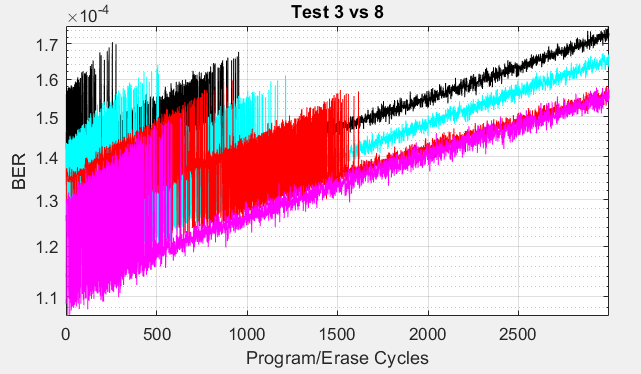
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bias** | **Baseline** | **Test3** | **Test6** | **Test7** | **Test8** |
| 0 | 0.125 | 0.225 | **0.225** | **0.2** | **0.175** |
| 1 | 0.125 | 0.2 | **0.2** | **0.225** | **0.225** |
| 2 | 0.125 | 0.175 | **0.175** | **0.175** | **0.2** |
| 3 | 0.125 | 0.15 | 0.15 | 0.15 | 0.15 |
| 4 | 0.125 | 0.1 | 0.1 | 0.1 | 0.1 |
| 5 | 0.125 | 0.075 | 0.05 | 0.05 | 0.05 |
| 6 | 0.125 | 0.05 | 0.075 | 0.075 | 0.075 |
| 7 | 0.125 | 0.025 | 0.025 | 0.025 | 0.025 |



Observing the “BER per state” graph we can notice several phenomena:

* When changing the bias numbers of the lower voltage end the effect on the BER levels is only local. In other words, a change in bias on stages 0-2 doesn’t seem to affect the BER numbers of the higher voltage levels.
* Test 8 shows that reducing the bias on state 0 and shifting the weight to state 1 (one state away from the edge like we did with stat6) has little to no affect on the BER and is still slightly better than test 3. This might show that the “higher the voltage the higher the BER” isn’t always true.

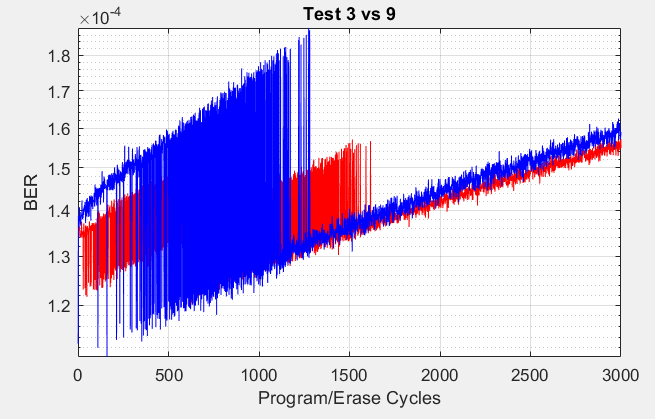
Overall, only test 8 offers less BER but yet to be seen what it install in the long run.



## Testing the limits of state 6 – “make or break”

The previous tests further proved our point but didn’t improve over test 3. We decided to push state 6 to the limit of its capacity with test 9 by increasing the bias on state 6 to 0.1 while reducing state’s 4 bias to see if reducing the bias from the mid-range could improve the BER:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bias** | **Baseline** | **Test3** | **Test6** | **Test9** | **Test10** |
| 0 | 0.125 | 0.225 | 0.225 | 0.225 | 0.175 |
| 1 | 0.125 | 0.2 | 0.2 | 0.2 | 0.225 |
| 2 | 0.125 | 0.175 | 0.175 | 0.175 | 0.2 |
| 3 | 0.125 | 0.15 | 0.15 | 0.15 | 0.15 |
| 4 | 0.125 | 0.1 | 0.1 | 0.075 | 0.075 |
| 5 | 0.125 | 0.075 | 0.05 | 0.05 | 0.05 |
| 6 | 0.125 | 0.05 | 0.075 | 0.1 | 0.1 |
| 7 | 0.125 | 0.025 | 0.025 | 0.025 | 0.025 |

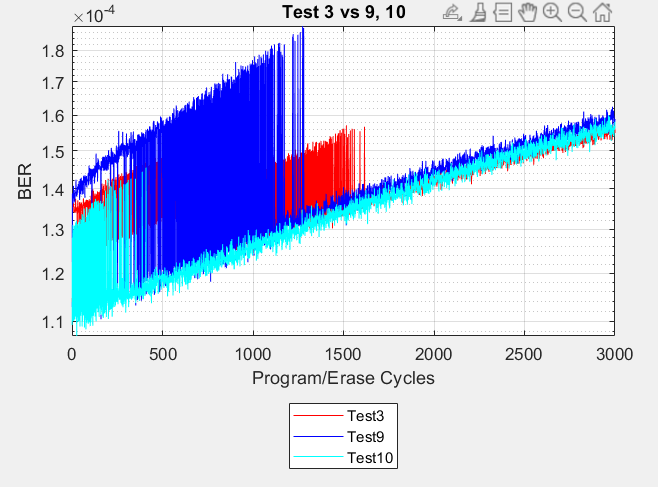


Viewing the results of test 9 we can see that the increase in state 6 bias has passed the sweet spot. Between test 3 & 6 the increase of 0.25 in the bias didn’t affect the BER much. But when adding an additional 0.25 the BER increased dramatically in both state 6 & state 0. Test 10 has a large reduction in state 0 but still the wear is worse that test 3.

## Testing both ends of the voltage spectrum

The previous tests of 7-9 tested different ends of the voltage spectrum. The following test came to combine test 8 by moving the highest bias to state 1 while increasing the bias on state 6 to test if by decreasing the bias on the lower end of the spectrum we could reduce the effect of test 6 on the lower states.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bias** | **Baseline** | **Test3** | **Test6** | **Test9** | **Test10** |
| 0 | 0.125 | 0.225 | 0.225 | 0.225 | 0.175 |
| 1 | 0.125 | 0.2 | 0.2 | 0.2 | 0.225 |
| 2 | 0.125 | 0.175 | 0.175 | 0.175 | 0.2 |
| 3 | 0.125 | 0.15 | 0.15 | 0.15 | 0.15 |
| 4 | 0.125 | 0.1 | 0.1 | 0.075 | 0.075 |
| 5 | 0.125 | 0.075 | 0.05 | 0.05 | 0.05 |
| 6 | 0.125 | 0.05 | 0.075 | 0.1 | 0.1 |
| 7 | 0.125 | 0.025 | 0.025 | 0.025 | 0.025 |

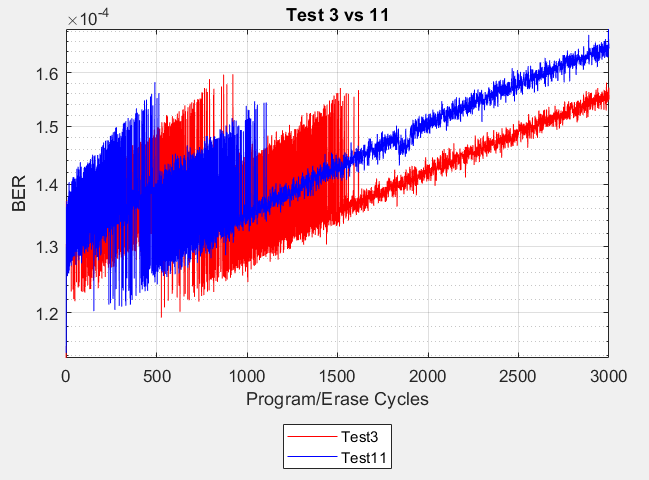


Yet again it seemed like we have reached the point where the small increase in bias of states 1 & 6 results in a large jump in the BER numbers. This test brings us to conclude that we can’t shift too much of the bias weight on state 1 & 6 at the same time.

## Relaxing the lower end voltage

After the previous attempts to imply the same theory that worked on the higher end of the voltage to the lower end, we decided to take another change of thought. We did see an improvement of state 0 in test 8 and wanted to further explore the option of improving the lower end using a different method. Maybe the notion that changing the bias for each individual state according to its BER doesn’t work on the lower end but rather a more regional approach should be taken. We decided to average the bias on states 0-2 at a rate of “0.2”. but this is definitely not the correct direction.

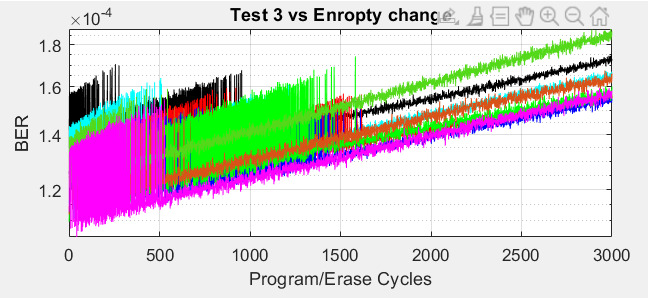
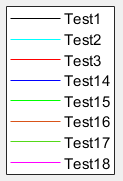
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bias** | **Baseline** | **Test3** | **Test6** | **Test11** |
| 0 | 0.125 | 0.225 | 0.225 | **0.2** |
| 1 | 0.125 | 0.2 | 0.2 | **0.2** |
| 2 | 0.125 | 0.175 | 0.175 | **0.2** |
| 3 | 0.125 | 0.15 | 0.15 | 0.15 |
| 4 | 0.125 | 0.1 | 0.1 | 0.1 |
| 5 | 0.125 | 0.075 | 0.05 | 0.05 |
| 6 | 0.125 | 0.05 | 0.075 | 0.075 |
| 7 | 0.125 | 0.025 | 0.025 | 0.025 |



we can see here that state 2 is increasing significantly:

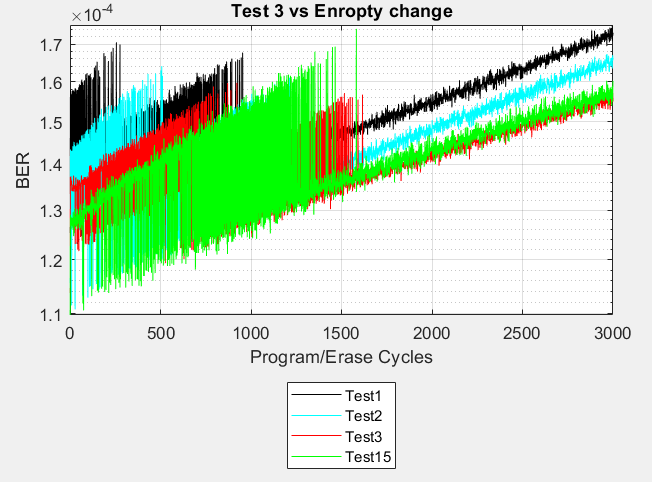
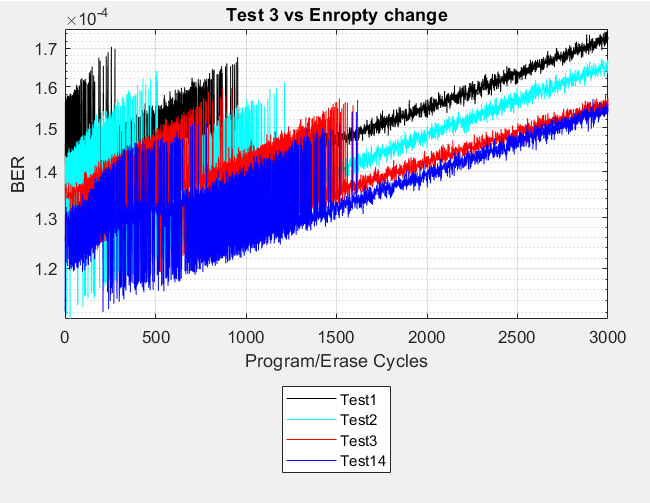
## Improving the entropy

After trying several permutations, the following tests focus on improving the entropy while trying to improve the number of errors. We tried a series of tests and finally we found a test improving of test3 and we slightly more entropy!



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Test 1 | Test 2 | Test3 | Test14 | Test15 | Test16 | Test17 | Test18 |
|  | 0.2 | 0.225 | 0.225 | 0.2 | 0.175 | 0.175 | 0.125 | 0.18 |
|  | 0.2 | 0.175 | 0.2 | 0.225 | 0.225 | 0.225 | 0.25 | 0.23 |
|  | 0.15 | 0.175 | 0.175 | 0.175 | 0.175 | 0.15 | 0.165 | 0.18 |
|  | 0.15 | 0.125 | 0.15 | 0.15 | 0.175 | 0.15 | 0.125 | 0.14 |
|  | 0.10 | 0.125 | 0.1 | 0.1 | 0.1 | 0.15 | 0.125 | 0.1 |
|  | 0.10 | 0.075 | 0.075 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 |
|  | 0.05 | 0.075 | 0.05 | 0.06 | 0.065 | 0.065 | 0.12 | 0.08 |
|  | 0.05 | 0.025 | 0.025 | 0.03 | 0.025 | 0.025 | 0.03 | 0.03 |
| entropy | **2.85** | **2.808** | **2.7608** | **2.7702** | **2.77** | **2.79** | **x** | **2.7944** |

From the graph above we can conclude that tests 14, 15 & 18 are the ones

****

## 

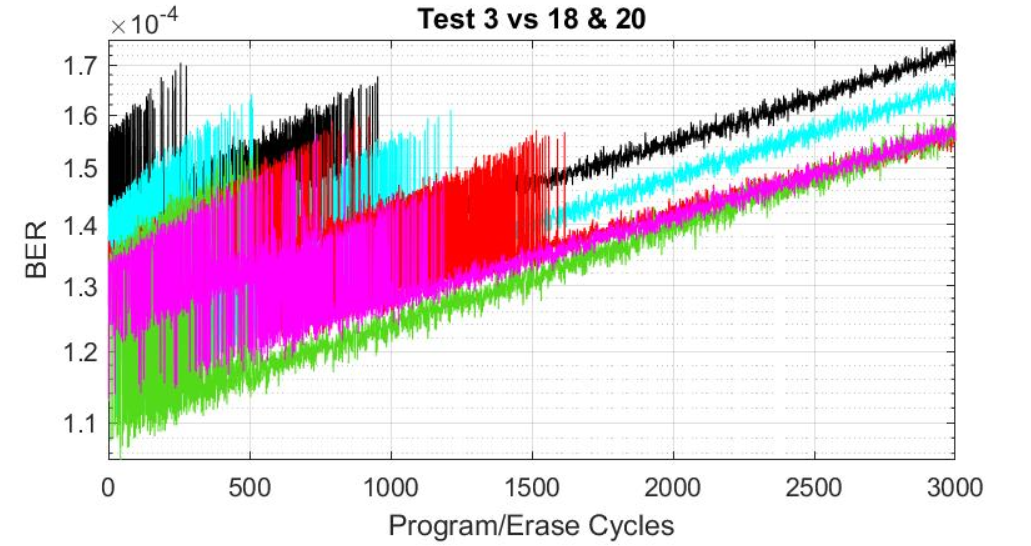
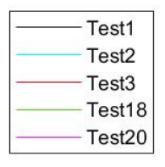
**Out of all the tests we ran test 14 is the most promising, offering a better entropy that test 3 with yet a lower wear on the TLC chip.** Although test 18 starts in a better staring position its wear rate is worse that test 3. But when compring test 18 to test 2, we see that the entorpy is almost the same althouh the wear is much better than test 2.

## The sweet spot

Once shifting the focus to the entropy, we started to look for the best sweet spot combining the entropy, BER & wear. Test 14 presents a good sweat spot, but test 18 presents a good balance offering an entropy close to test2 while its BER & wear are close to test 3.

The issue we find with test 18 is that the wear is worse than test3. So shifting the bias a little to the higher end of the voltage we arrive at test 20 with better entropy and lower wear that test3.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Test 1 | Test 2 | Test3 | Test14 | Test15 | Test16 | Test17 | Test18 | Test20 |
|  | 0.2 | 0.225 | 0.225 | 0.2 | 0.175 | 0.175 | 0.125 | 0.18 | 0.19 |
|  | 0.2 | 0.175 | 0.2 | 0.225 | 0.225 | 0.225 | 0.25 | 0.23 | 0.23 |
|  | 0.15 | 0.175 | 0.175 | 0.175 | 0.175 | 0.15 | 0.165 | 0.18 | 0.19 |
|  | 0.15 | 0.125 | 0.15 | 0.15 | 0.175 | 0.15 | 0.125 | 0.14 | 0.14 |
|  | 0.10 | 0.125 | 0.1 | 0.1 | 0.1 | 0.15 | 0.125 | 0.1 | 0.1 |
|  | 0.10 | 0.075 | 0.075 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.07 |
|  | 0.05 | 0.075 | 0.05 | 0.06 | 0.065 | 0.065 | 0.12 | 0.08 | 0.06 |
|  | 0.05 | 0.025 | 0.025 | 0.03 | 0.025 | 0.025 | 0.03 | 0.03 | 0.03 |
| entropy | **2.85** | **2.808** | **2.7608** | **2.7702** | **2.77** | **2.79** | **x** | **2.7944** | **2.7814** |



## Conclusions

Our initial challenge was to maintain the same entropy of test3 and decrease the BER level. We found great difficulty in achieving a major reduction compared to test 3 but managed to get bet entropy numbers while maintaining the same or lower BER. In order to reach this goal, in the course of our research, we tried to characterize the 3D TLC chip behavior to the changes in the different bias levels.

Here are our conclusions regarding the 3D TLC chip behavior:

1. The BER levels of the 3D TLC chip aren’t optimal when sorting the bias levels in an ascending voltage level order - in contrast to the assumption in tests 1-3 that the higher the voltage the greater the BER of the TLC chip is. We observed that each state should be analyzed individually to achieve the optimal sweet spot between entropy & BER.
2. Changing the bias per state according to the BER level of the state isn’t always correct:
   * Increasing the bias in the higher states affects the BER levels in the lower states as well (test 7).
   * In the lower states of 0-2 setting the same bias number is not the way to go since those states behave very differently from each other (test 11).
3. There is a breaking point for each state. beneath the breaking point an increase in the bias doesn’t affect much the BER (test 6) but above that point a small increase in the bias will result in a jump in BER numbers (test 18 on state 6 and other).
4. The main conclusion has to do with the trend of the bias which is optimal which can be seen in this graph below comparing test3, test14 & 18