PAPER I – ECP, not ECC

1) Summary

Resistive memories are promising and more scalable replacement of DRAM. But, the resistive memories such as PCM have lesser write endurance. Majority of the faults in such type of memories are hard faults (stuck-at faults) and not the soft faults. ECC codes are generally used to provide protection against soft errors with no limitations on effective lifetime of the cells considered. **The major contribution of this paper is a new correction technique, Error-Correcting Pointers, targeting the immediately detectable hard faults as found in resistive memories.** The read-write-read pattern can be used to perform write operations resulting in less number of cells needed to be modified and immediate detection of cell failures. ECC codes get modified on each write operation and thus, these cells wear out faster. ECP minimizes the wearout because of writes, handles permanent faults and improves overall memory lifetime while considering the lifetime variability of the memory cells.

ECP uses the concept of correction entries consisting of correction pointer to point to the failed cell and replacement cell to contain the correct value for the faulty cell. The technique is able to protect the data cells as well as its own data structures. The paper also presents a **good comparative analysis** of this technique with other existing techniques as well as some optimized ECP versions such as layered ECP and ECP with intra-row wear leveling.

2) Best Point

The technique seems to be really simple and might have got inspired by the Wilkerson technique discussed in the paper. But, I felt that the **best point is the comparative analysis presented** by the authors. They present the various existing techniques and a theoretical “Perfect_Code” technique. They present an **analysis that for a targeted overhead limit how effective each technique is**. The results show that the proposed method outperforms other existing techniques and is competitive with “Perfect_Code”. The interesting point is that **rather than just taking the existing techniques and comparing, authors try to enhance them to improve their results as per the available environment and then compare with the proposed technique**. This shows the fairness maintained in the study and thus, proving its effectiveness and superiority.

3) Worst Point

The approach is **too simple**. The **storage overhead is high for wide scale adoption** of this technique to cover maximum possible failures. This technique doesn’t provide any good approach to reduce the overhead. Moreover, it **assumes that the weakness level of all the lines is same**. But, many of the lines may require fewer or different number of ECP entries and thus, some entries are going to remain unused. So, **large overhead and per row unutilized correction entires contribute to make this scheme not suitable for practical adoption**. It also falls back to ECC like protection in case of transient faults (at high temperatures) and thus, adds extra overhead.

3) Relation

The basic concept is of providing **redundancy** to increase **lifetime reliability** of the chip (that we studied in the class). Faults are masked by providing redundant replacement cells to contain the values for the faulty cells. It is similar to **the concept of spares that we studied, though at very fine grain level.** We have replacement cells as spares which are used only when the data cells fail.

The basic motive of the proposed technique is same as the other **error detection/correction techniques** (hamming code, SECDED etc.) studied. But, most of these techniques try to fix the transient faults in place while ECP provides replacement cells to handle the permanent failure of data cells**.** Like ECC codes, ECP is able to protect its own data structures**. Parity based RAID techniques** also help to recover from permanent failures such as disk failure. So, we can say that this technique effectively masks the fault to facilitate error correction.

4) Improvement

I would like to take this scheme and try to **optimize it to reduce its performance overhead and make it more suitable for the purpose of error correction in practical systems**.

It would be good to explore some **approach to divide the correction entries among the lines more efficiently**. Finding better technique will help in reducing the overhead and thus, controlling the size and complexity. The proposed technique assumes that there are no transient faults. But, we should provide protection against transient faults for practical adoption. If we are able to reduce the overhead for correction entries, we may also provide protection against transient faults within the same targeted overhead limit. Therefore, complexity reduction and coverage enhancement would be my approach to improve this technique.

PAPER II – Proactive Wear-out Recovery

1) Summary

The chip lifetime reliability gets affected by the wear out failure mechanisms such as NBTI (Negative bias temperature instability), electro-migration etc. Micro architectural redundancy can be used to improve the chip lifetime reliability. This paper proposes **a proactive use of micro architectural redundancy in which non-faulty components can transition between active and recovery states to recover from certain wearout effects.** The key contribution is the introduction of a new **approach to exploit the wearout recovery properties of certain failure mechanisms to ward off the onset of wearout failures** improving the lifetime reliability.

The authors **limit the scope to cache SRAM arrays and NBTI** failures as the wearout recovery can be facilitated by completely eliminating the electric field (and thus, the stress) in wearout recovery mode. The **authors propose a new design for 6T SRAM cell array** to allow exploitation of NBTI wearout recovery properties. **Spatial redundancy is used to allow some of the components to be in recovery mode** at a time and scheduling is done to provide required recovery for all. The paper also provides the evaluation results showing that the proactive approach provides **better lifetime-performance and lifetime-area trade-offs** as compared to other approaches.

2) Best Point

The techniques suggested previously either try to manage cell failures using spares (reactive) or try to reduce the stress conditions (proactive). This paper guides us to look into **a new direction where we exploit the wearout recovery properties of failure mechanisms in an intensified way and thus, can delay the failure of all the micro-architectural components altogether** using some redundancy. This is the best point as it shows that if we try to study the failure mechanisms, how they happen and can they be reversed, we can exploit these properties and design the chips accordingly to provide better solutions to delay failures increasing the lifetime reliability of the chip.

3) Worst Point

The worst point is the **narrow coverage and much overhead**. This proposal focuses on SRAM cache arrays and NBTI induced wearout. So, it **can delay the failures only where NBTI recovery properties can be exploited** and thus, we would have to adopt some other techniques to enhance lifetime reliability of whole chip. Also, there is much overhead for this limited coverage. It requires a new 6T SRAM cell array design, forcing to **change the cache internals** and also **assumes configuration support** to allow the rotation using redundancy. More overhead gets added as we need to **drain the arrays repeatedly**, need to **block the cache lines** and can also **result in cache misses**. In addition, **some extra wiring and multiplexing** is also required in case of migration drain mechanisms.

In addition, I feel the **evaluation done to show the superiority of the scheme is unfair** as other techniques are more generic and may have less overall overhead. So comparing their performance with this scheme (which is only for NBTI induced failures) is not fair.

4) Relation

It uses the concept of **spatial/physical redundancy** and **lifetime reliability** studied in the class. It **is similar to the concept of error correction codes studied in the class in the sense that they also try to fix/mask the faults to delay failure**. Most of the techniques studied try to fix transient faults during intrinsic failure period. But, this paper tries to delay the permanent failures due to wearouts. So**, ECC is basically a detection and recovery approach while this is a prevention technique**. ECC codes are general with wider scope. RAID also uses redundancy but that is also a detection and recovery technique. **The concept of cache draining used here is also used in check pointing and recovery.**

It focuses particularly on SRAM cache arrays with NBTI failures which covers many devices and thus, targeting more vulnerable components. AVF also helps us to decide which components are more important. So, we can intuitively make a relation although no such thing has been discussed in paper. It is also distantly related to razor as razor tries to reduce the voltage and fixes timing errors also. This technique also tries to eliminate the stress in recovery mode by setting gate-source voltage to 0 and thus, avoids failures for longer time.

5) Improvement

I liked the concept of exploiting wearout recovery properties even though it has limited scope from the chip's perspective. I would like to **explore other failure mechanisms** leading to the wearout of devices. By studying them, we can figure out if they have some exploitable recovery properties allowing us **to target different parts of the chip**. We can **aim for a chip designed in a way to facilitate the exploitation of wearout recovery properties for different failure mechanisms in different parts and see how much enhancement we can provide in the lifetime reliability** of the chip. It would allow to get a real picture and usefulness of this approach.

Relation and Comparison between the Two Papers Read

Both the papers have been proposed with similar goal to **deal with permanent/wearout failures and thus, to increment the lifetime reliability of the chip.** The ECP paper targets the resistive memories such as PCM while the other paper targets the SRAM cache arrays**. ECP is a reactive approach** which provides a replacement cell for each failed one while the **other one is a proactive approach** which uses rotation and redundancy to keep some of the components in recovery mode. **ECP is a general approach for resistive memories** and can be used to mask permanent cell failures irrespective of the failure mechanisms. **Other technique is specific to NBTI** induced wearouts. **Both the techniques are not complete in themselves** and thus, may require other traditional techniques such as ECC to cover other faults.

ECP paper also proposes an optimization using intra-row wear leveling. This is the basic concept of rotation with redundancy used in proactive recovery but for arrays. So, they might got inspiration from the same source as no reference to proactive paper (published in 2008) has been found in ECP paper (published in 2010). ECP doesn't cover issues related to elevated temperature and suggests use of ECC. But, other paper focuses on NTBI which happens at elevated temperature. ECP seems to be more economical. Maximum failure count that ECP can mask is governed by number of replacement cells while other approach tries to delay failure of all components.

I would also like to explore an approach in which we can combine the reactive and proactive techniques to achieve better lifetime reliability.