Exploiting a Fast and Simple ECC for Scaling Supply Voltage in Level-1 Caches

Gulay Yalcin[1] Emrah Islek[2] Oyku Tozlu[2] Pedro Reviriego[3] Adrian Cristal[1,4] Osman S. Unsal[1] Oguz Ergin[2]
[1]Barcelona Supercomputing Center
[2]TOBB University of Economics and Technology
[3]Universidad Antonio de Nebrija
[4] IIIA - Artificial Intelligence Research Institute - CSIC - Spanish National Research Council

***Abstract*—Scaling supply voltage to near-threshold is a very effective approach in reducing the energy consumption of computer systems. However, executing below the safe operation margin of supply voltage introduces high number of persistent failures, especially in memory structures. Thus, it is essential to provide reliability schemes to tolerate these persistent failures in the memory structures. In this study, we adopt a Single Error Correction Multiple Adjacent Error Correction (SEC-MAEC) code in order to minimize the energy consumption of L1 caches. In our evaluations, we present that the SEC-MAEC code is a fast and energy efficient Error Correcting Code (ECC). It presents 10X less area overhead and 2X less latency for the decoder compared to Orthogonal Latin Square Code, the state-of-the art ECC utilized in the L1 cache under the scaling supply voltage.**

I. Introduction

Technology minimization and associated gains in performance and productivity are in jeopardy as the power density and energy consumption of modern computing devices increase. For instance, smaller transistors allow computer designers to pack more chips in each technology node in the same area. However, within the given power budget, not all portions of the chip can be powered. The area of the chip which are not powered is termed as *dark*, and this is a recent problem known as Dark Silicon Phenomena [4]. Thus, in order to keep the power under control, a dramatic improvement in the energy efficiency of microprocessors is required.

Downscaling the supply voltage (V_{dd}) close to the transistors' threshold (near-threshold execution) is a quite effective approach for minimizing the energy consumption of the computer systems [8]. Therefore, microprocessors provide ***high-performance*** (i.e. uses nominal voltage at a relatively high frequency) and ***ultra-low power*** (i.e lowers the V_{dd} to reduce energy consumption by trading off performance) operating modes [6], [9]. However, when the V_{dd} is reduced lower than the safe operation margin, it causes persistent failures especially in memory structures [3]. This is because the number and the location of the dopant atoms vary randomly (i.e. random dopant fluctuation). In a SRAM cell, a mismatch between the strength of neighbouring transistors can lead to failing of the cell. This cell failure occurs persistently in the low V_{dd}, and it disappears when the V_{dd} is increased back to the safe margin. More drastically, the number of failing cells increases exponentially by the reduction of V_{dd} leading to multiple faults in the same processor structures [10]. Thus, it becomes crucial to build reliability solutions in the memory structures to be able to take advantage of energy saving provided by scaling V_{dd}.

Utilizing Error Correcting Codes (ECC) in the low-power operating mode is an appealing solution for reducing the safe operating margin for V_{dd} of memory structures. ECC extends data lines with additional parity bits. The encoder of the ECC generates parity bits when the data line is updated. In the reading of the data line, the decoder regenerates the parity bits to check and correct any existing fault. However, the energy and time spent by the encoder and the decoder limits the energy reduction provided by a lower V_{dd}. More specifically, time-critical storage structures such as L1 caches can not tolerate several cycles error correction latency. However, many ECC schemes require a couple of cycles for the decoder when the fault rate is very high as in near-threshold voltage execution. Similarly, their complex encoders present high energy overhead when the error rate is high [3], [10].

In this study, our goal is reducing the supply voltage of L1 caches to near-threshold voltage in order to minimize the energy consumption of it without harming the access time and read/write energy of the cache in the high-performance mode. To this end, we propose adopting a Single Error Correcting - Multiple Adjacent Error Correcting ECC (SEC-MAEC code) - a fast and energy efficient ECC with a high error correction capability - in the faulty lines of L1 caches for below safe voltage operations. The SEC-MAEC code used [12] is designed for multi-bit soft errors occurring in the adjacent bits in order to correct them in one cycle. It can accomplish the error correction via only 4 level gate pass while presenting 100% area overhead for the parity bits.

Besides correcting errors in the low-power mode execution, it is also important not to sacrifice the useful cache capacity in the high-performance mode due to the area overhead of SEC-MAEC. For instance, in a naive approach all lines are protected with SEC-MAEC code, which shrinks the useful cache capacity into half even in the high-performance mode although the error rate is very low in this mode. Furthermore, although SEC-MAEC have been proposed previously, it is not presented how to apply it to memory structures such as 4-way L1 caches. In this study, we also present how to organize L1 caches with SEC-MAEC code in order to protect only the faulty cache lines after performing a memory test in the boot time. Therefore, our implementation does not sacrifice any cache capacity in normal operating mode.

We compare the selected SEC-MAEC code with the Orthogonal Latin Square Code (OLSC), a state-of-the-art fast ECC scheme utilized in L1 caches in order to lower the supply voltage [5]. We present that SEC-MAEC reduces the area

978-1-4799-5324-0/14/$31.00

overhead of decoder by 10X while reducing the encoding and decoding latency into half. We also show that the energy spent for encoding and decoding can be reduced up to 80%.

The main contributions of this study are followings:

- We propose utilizing the SEC-MAEC code in [12] for L1 caches in order to reduce the supply voltage to near-threshold voltage level and to provide substantial energy savings.
- We present the cache organization for applying SEC-MAEC codes into L1 caches without harming the full cache capacity usage in the high-performance mode.
- We demonstrate that the SEC-MAEC code reduces the area overhead, energy overhead and latency of encoder and decoder substantially compared to OLSC while presenting similar useful cache capacity.

In the following section (Section II), we explain the principles of the SEC-MAEC code that we utilized in this study and the OLS Code, the state-of-the-art ECC used for lowering V_{dd}. In Section III, we present how we extend the faulty cache lines with SEC-MAEC codes. In Section IV, we evaluate our proposal. In Section V, we present previous ECC schemes and the benefits of SEC-MAEC over them. We conclude in Section VI.

II. Background

In this section, we explain the principles of the SEC-MAEC code and the OLS Code.

A. Single Error Correction Multiple Adjacent Error Correction Codes (SEC-MAEC)

For Error Correcting Codes (ECCs) utilized in L1 caches, decoding latency is a key parameter since L1 is a time-critical component during the execution. Recently Reviriego et al. [12] have proposed Single Error Correction, Multiple Adjacent Error Correction (SEC-MAEC) codes whose encoders and decoders are very energy efficient and fast. These codes can correct single bit errors and also multiple bit errors as long as they affect adjacent bits. This is useful to correct multiple bit errors. We argue that they are also potentially attractive to use in the high-fault rate below safe voltage operation, which also produces multiple bit errors. Additionally, the low decoding latency makes these codes attractive to protect L1 caches. Thus, in this study we investigate using SEC-MEAC codes for scaling V_{dd}.

Figure 1 shows the encoder and decoder of proposed SEC-MAEC codes. The SEC-MAEC code with the ability of correcting s adjacent multiple errors in a data block consisting of k bits $\{d_0, d_1, d_2, ..., d_k\}$ generates k parity bits $\{p_0, p_1, p_2, ..., p_k\}$. s can be upto 2,5,10, ... depending on the number of data bits. In Figure 1, we illustrate the encoder and decoder of a SEC-MAEC code where s=2 and k=8. Depending on the size of the data block, the maximum value of s can be 2 (k = 8), 5 (k = 16), 10 (k = 32) and so on.

The encoder of the SEC-MAEC code computes parity check bits with the following equation:

$$p_i = d_i \oplus d_{mod(i-s,k)} \quad (1)$$

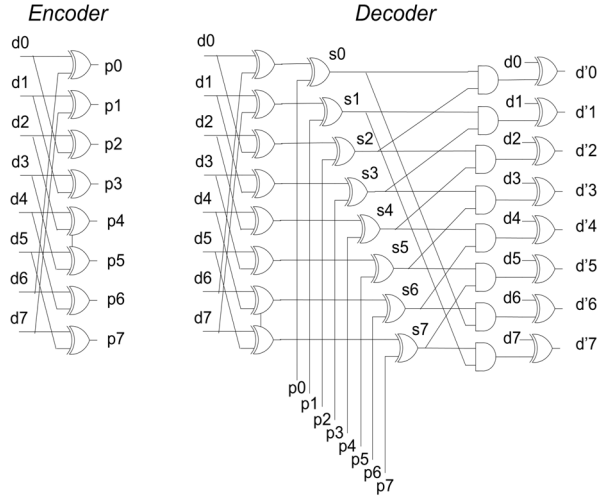


Fig. 1: Encoder and Decoder of the SEC-MAEC code

As it can be seen, the encode operation is accomplished for all bits in parallel, in one gate level. For instance, even single parity bit calculation in which all data bits are XORed, and only odd number of errors can be detected without correcting them, requires at least $log_2 k$ gate levels.

Parity bits and data bits compose a codeword. For a codeword (faulty or not) consisting of k data bits and k parity bits, SEC-MAEC decodes it in two stages: 1) SEC-MAEC generates k bit syndrome bits $\{s_0, s_1, s_2, ..., s_k\}$, 2) it calculates d', the corrected data after the decoding. Syndrome bits are calculated as:

$$s_i = p_i \oplus d_i \oplus d_{mod(i-s,k)} \quad (2)$$

Briefly, syndrome bits detect whether there is an error in the codeword or not. If all syndrome bits are zero, that means that the codeword (all parity bits and data bits) is error free. After the syndrome bits, the correct data d' is also computed with the following equation:

$$d'_i = (s_i \cap s_{mod(i+s,k)}) \oplus d_i \quad (3)$$

The decoding logic of SEC-MAEC code is also quite simple and fast. More precisely, each correct d' bit is obtained with 4 gates (i.e. 3 XOR gates and 1 AND gate), and all bits can be calculated in parallel. Therefore, SEC-MAEC provides error correction with fast and area efficient encoders and decoders which makes it appealing to be used for low-power mode executions in the cache. Moreover, SEC-MAEC can detect and correct faults occurring either in parity or data bits. Thus, two faulty words can be used together to become one functioning word.

B. Orthogonal Latin Square Codes

OLSC are based on the concept of orthogonal latin squares [5] and can be decoded using majority voting. An OLSC encodes "orthogonal" groups of bits to form check bits. At decoding time, each data bit generates the final value through a voting process from a group of orthogonally coded data and check bits. Thus, an OLSC does not need to generate a syndrome but can "correct" errors directly from majority voting. To perform t error corrections in a data block consisting of m×m bits, an OLSC requires 2×t×m check bits.

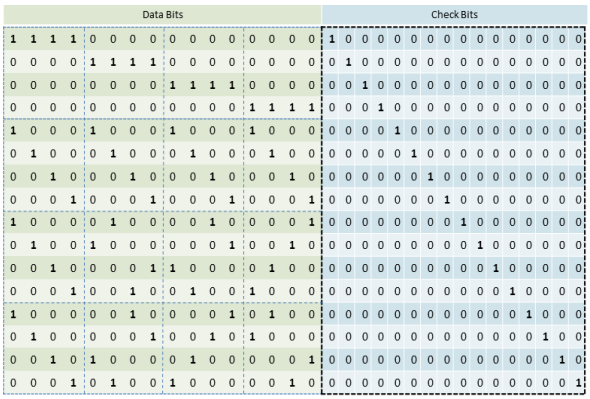


Fig. 2: Parity check matrix for the Orthogonal Latin Square Code with k = 16 that can correct two errors

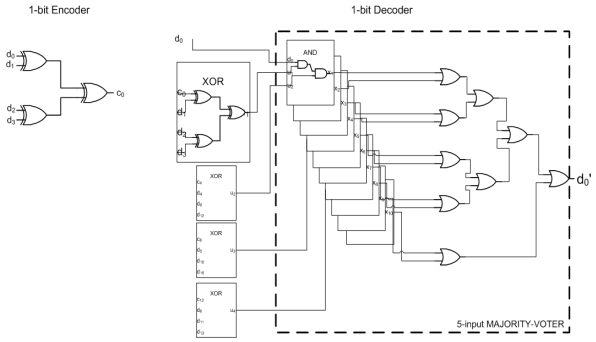


Fig. 3: OLSC 1-bit Encoder and Decoder

In Figure 2, we present the H-matrix of an OLS code for an 16-bit data correcting up to 2 bit errors (i.e. m=4, t=2).

The encoder computes each check bit C_i as the XOR over data bits corresponding to columns of the H-matrix that have a '1' , such that;

$$\begin{aligned} c_0 &= d_0 \oplus d_1 \oplus d_2 \oplus d_3 \\ c_1 &= d_4 \oplus d_5 \oplus d_6 \oplus d_7 \\ &... \\ c_{15} &= d_3 \oplus d_5 \oplus d_8 \oplus d_{14} \end{aligned}$$

Each row in the H-matrix has exactly m bits of '1's. Thus the calculation for each check bit requires an m-input XOR operation with the critical path being ceil($log_2(m)$) levels of 2-input XOR gates.

The decoder decides the correct value of d_i via (2t+1)-input majority voter. One input of the voter is the received d_i' itself, the other 2t are derived from check bits that contain d_i as its encoding variable, such that;

$$\begin{aligned} d_0' = Majority \quad & (d_0, \\ & (c_0 \oplus d_1 \oplus d_2 \oplus d_3), \\ & (c_4 \oplus d_4 \oplus d_8 \oplus d_{12}), \\ & (c_8 \oplus d_5 \oplus d_{10} \oplus d_{15}), \\ & (c_{12} \oplus d_6 \oplus d_{11} \oplus d_{13})) \end{aligned}$$

The critical path for the decoder is ceil($log_2(m)$) levels of 2-input XOR plus (2t+1)-input majority function. Figure 3 presents the circuit logic for encoding and decoding single bit in an OLS code for an 16-bit data correcting up to 2 bit errors. We implemented the majority voter with 5 inputs required by the defined OLSC in a way that if any of the three inputs are 1, the result becomes 1, otherwise 0. In this design, while the critical path of the encoder consists of 2XORs, it consists of 2XORs+2ANDs+4ORs for the decoder.

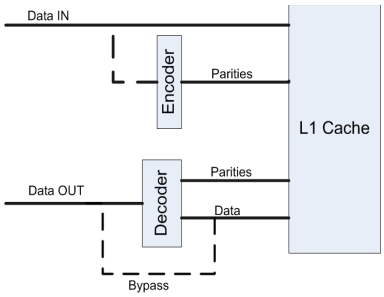


Fig. 4: High-level overview of the architecture

III. Architecture

In this section, we explain the architectural design of the first level caches which utilizes SEC-MAEC for saving energy in the low-performance mode in which V_{dd} is reduced.

ECC schemes used for near-threshold voltages require a high number of parity bits (e.g 100% area overhead) in order to increase the error correction capability. However, this high area overhead does not present any benefit in the high-performance mode in which failure rate is very low. Thus, adaptive and variation-aware ECC-based schemes selectively enables protection for only the faulty cache lines in the low-power execution mode [3], [10], [2]. In Figure 4, we present the high-level overview of the cache architecture protected by an ECC. All writes to the faulty cache lines go through encoder while reads from faulty cache lines go through the decoder. Also, for lines requiring no protection, encoding/decoding is bypassed, which reduces access latency and the energy overhead.

In order to enable the protection for the faulty lines dynamically, the hardware should have the ability of determining the faulty lines. Failures due to voltage variation do not present a dynamic behaviour, and they occur persistently in the given supply-voltage level. Thus, faulty lines can be detected easily via simple built-in self test (BIST) in postmanufacturing and can be saved to a storage structure. BIST writes and reads two test patterns to each line: one containing all 0's and one with all 1's, so that, it can determine the relative vulnerability of cache lines in each voltage interval. After the test, the cache configuration maximizing the capacity at each voltage interval is saved to some memory location (i.e. ROM or main memory) in the system.

It is not trivial to configure the cache when the supply voltage is lowered. In a naïve approach, the ECC-protection is disabled entirely in the high-performance mode, and it is enabled for all lines in the low-power mode after some V_{dd} level (i.e. safe V_{dd} level). This method has been utilized by MS-ECC in L1 cache [3]. However, due to the non-uniform distribution of errors, variation-aware protection algorithms that consider the relative vulnerability of cache lines are amenable in order to maximize the useful cache capacity. In this study, we present a variation-aware algorithm for the SEC-MAEC code when it is utilized in a 4-way L1 cache. In

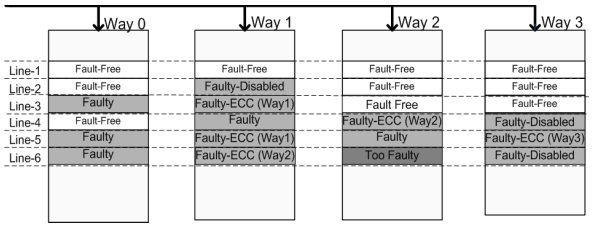


Fig. 5: The organization of cache ways for faulty lines and the ECC values

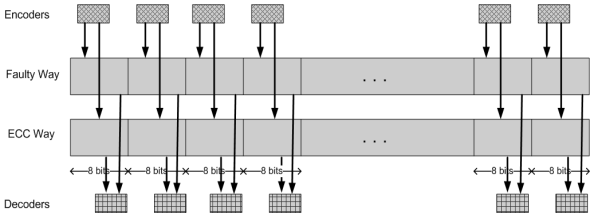


Fig. 6: Direct mapping between the data and ECC partitions

our algorithm, we inspired by the one utilized for 8-way L2 caches [10] in which higher number of ways in a cache line presents higher flexibility for the cache organization.

In Figure 5, we present an example for the organization of the cache ways for faulty cache lines and ECC values of the faulty lines. We do not extend fault-free cache ways with ECC so that in the high-performance mode execution, the entire cache capacity is available (Line-1 in Figure 5). When only one way is faulty in the cache line (Line-2), instead of allocating a fault-free cache line for ECC, we disable the faulty cache way. In this way, we present the same cache capacity (e.g. 3 out of 4 ways are utilized) without increasing the access time due to encoding/decoding logic. When there are 2 faulty cache ways in a line (Line-3), we combine these two ways. We save the data in one of them and the ECC to the other one. So that we can still provide 3 useful cache ways out of 4 ways. When there are 3 faulty-ways in a line (Line-4), we combine two of them and disable the third one. When all the ways are faulty (Line-5), we save data to two of them and ECC values to the other two. Finally, when the supply voltage is very low as in near-threshold execution, there can be a cache way which has more errors than it can be corrected by the utilized ECC scheme (Line-6). In this case, we disable this cache way as well. In order to accomplish this cache organization, we need to save two informations for each cache way: 1) is it fault-free, faulty-data, faulty-ecc or disabled (2 bits) 2) if it is faulty, the address of the pair cache way for the ECC or data (2 bits). Thus, this cache organization presents 16-bit (4-bit in a 4-way cache) area overhead per a cache line which is a negligible overhead for 2Kb cache lines (512-bit per way). Also, it is possible to save this information in the tag area of the cache. Note that, we left the protection of the tag area out of the scope of this study.

The second question for the SEC-MAEC usage in the L1 cache is the organization of the data-blocks. We use the SEC-MAEC with the configuration that k=8 meaning that the size of the data block is 8 bits. In order to apply it, we divide the cache lines into partitions consisting of 8 bits (i.e. 64 partitions in a 64B cache line). We also divide the ECC way into partitions with the same size (64 8-bit partitions) and

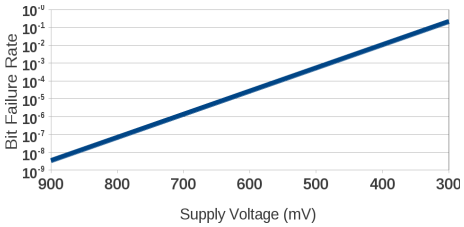


Fig. 7: Bit Failure Rate vs Supply Voltage

combine the data partitions with the ECC partitions in the same order. We present this direct mapping in Figure 6.

For the 8-bit data sizes, when s=2, SEC-MAEC can correct two adjacent bit failures. Similarly, when s=3 it can correct 2 failures that are 1-bit far from each other. Thus, due to the random distribution of failures, some lines can be corrected with s=3 although they can not be corrected when s=2 or vice versa. In order to take advantage of both configurations for different lines, we decide s parameter dynamically during the BIST for each cache way. In order to accomplish this dynamic decision, we extend each cache-way with a single bit determining s parameter used (s=2 or s=3).

In a cache way, when the V_{dd} is in the middle-low level (i.e. lower than the safe margin but higher than the near-threshold level), since the bit failure rate is not drastically low, there could be several error-free partitions in a faulty cache way. One can think of applying the optimization of extending only the faulty partitions with ECC values in order to reduce the ECC overhead. For instance if k of 64 partitions are faulty, parity bits for only those k partitions are calculated and saved. However, in this case, the ECC partitions need to include the address of the data partition that they are combined with. This addressing requires 8 additional bits (e.g. addressing 64 partitions in 4 ways) for each ECC partition which diminishes the benefit of ECC. Thus, in this study we avoid this optimization and we utilize direct mapping between the ECC and data values.

IV. Evaluation

In this section, we evaluate how much energy reduction in L1 caches can be provided by SEC-MAEC code. We compare SEC-MAEC with MS-ECC scheme [3]. MS-ECC uses Orthogonal Latin Square Code (OLSC) which is the state of the art, multi-bit correcting ECC used for the voltage scaling in L1 caches [5]. The complexity of OLSC scales well with the number of error corrections, thus, Chishti et al. [3] use OLSC for L1 caches. We utilized the OLSC code with a block size of 16 with the error correction capability of up to 2 failures in a block. Note that OLSC block size is optimized for the block size of m^2, and its encoder/decoder complexity increases when the larger blocks are used. Thus, we choose the best combination for OLSC. We analyse 1) useful cache capacity, 2) error correction latency, 3) area overhead and 3) energy minimization. We calculate the useful cache capacity as the ratio of the portion of the cache which can be used reliably (i.e. the lines which are not disabled after the memory test at the postmanufacturing time). We use fault injection methodology to measure the cache capacity under the bit failure rates (i.e. probability of a bit fails). We repeat each fault injection experiment 100 times for each failure rate. We empirically decide that 100 fault injection for each V_{dd}

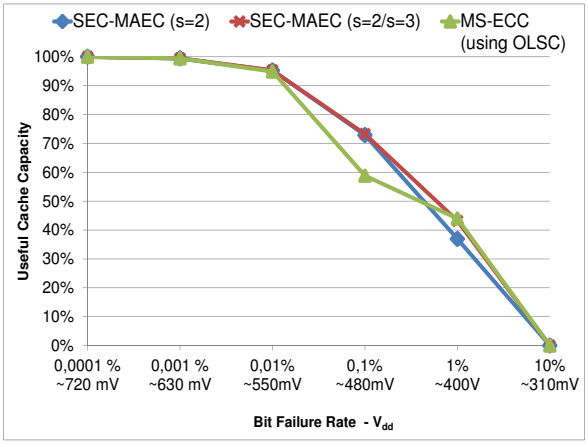


Fig. 8: Useful Cache Capacity

is enough since the standard deviation of our experimental results in each point is low. The mapping between the bit failure rate and the V_{dd} is examined by Miller et al [10] which can be seen in Figure 7. It is shown that as V_{dd} is lowered, the bit failure rate increases exponentially. We reference these previous results for our evaluations.

In Figure 8, we compare the useful cache capacity provided by SEC-MAEC and MS-ECC versus the bit failure rate. In order to present the benefits provided by cache-way-organization and dynamic-s-decision separately, we show two configurations of SEC-MAEC in the figure. SEC-MAEC (s=2) presents the benefit of cache-way organization on useful cache capacity. SEC-MAEC (s=2/s=3) presents the benefit of dynamic decision of s-parameter on top of cache-way organization (Note that, since we use 16-bit chunks for SEC-MAEC, s can take the value of upto 5). Both SEC-MAEC and OLSC provide similar cache capacity when the V_{dd} is close to the safe operating margin (i.e higher than 550mV). When the V_{dd} is middle low (i.e. between 550mV-430mV), SEC-MAEC extends some of the cache ways with ECC protection. On the other hand, MS-ECC does not activate the ECC protection for any cache way when the V_{dd} is higher than a threshold value. Note that, this threshold value should be determined as the V_{dd} level in which provided useful cache capacity without ECC protection is around 50%. Thus, when V_{dd} is middle low, SEC-MAEC presents higher cache capacity due to its cache organization in the architecture. In the near-threshold voltage execution (lower than 450 mV), while the cache capacity provided by SEC-MAEC (s=2) is lower than MS-ECC, SEC-MAEC (s=2/s=3) presents a high cache capacity similar to MS-ECC.

	SEC-MAEC		OLSC	
	Encoder	Decoder	Encoder	Decoder
Number of Gates in the Critical Path	1 XOR	3 XORs + 1 AND	2 XORS	2 XORs + 2 ANDs + 4 ORs
Total Number of Gates	512 XORs	1,5K XORs + 512 ANDs	1,5K XORs	6K XORs + 10K ANDs + 4,5K ORs
Latency	39 ps	182 ps	78 ps	324 ps

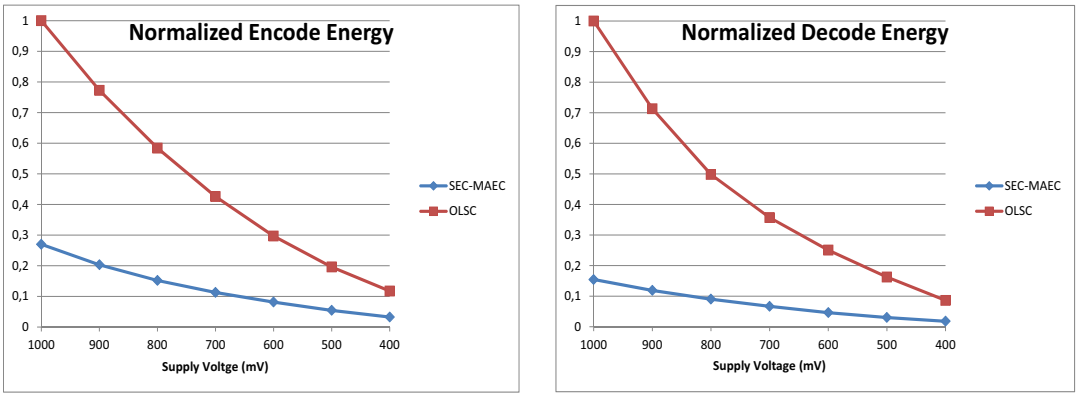
TABLE I: Number of Gates in Encoder and Decoder for a 64B cache lines

In Table I, we compare the area overhead and the latency presented by encoders and decoders in SEC-MAEC and OLSC. Note that we evaluate latency in the nominal voltage level when the frequency is 1 GHz. In the low-power mode, we reduce the frequency to 400 MHz. We first present the number of gates in the critical paths. SEC-MAEC has nearly half number of gates in the critical path compared to OLSC both for the encoder and decoder. This also affects the time spent in the encoder and the decoder, thus, the latencies of SEC-MAEC are much less than OLSC. Decoder latency of SEC-MAEC is much less than 10% of a clock cycle while it is more than 30% of a cycle for OLSC. Note that in this study, we only target correcting V_{dd} dependent, persistent failures, and soft errors are an independent issue. We assume that the memory structure is protected for soft errors by any other additional means such as parity or Single Error Correction Double Error Detection (SECDED) Code. The area overhead of parity/SECDED bits is relatively low, and they can be saved to the tag area of the cache. 'Obviously, when the V_{dd} is below the safe margin, the decoder of parity/SECDED can be activated after persistent failures are corrected. Nevertheless, the low latency of the decoder of SEC-MAEC codes leaves additional time for the decoder of soft error protection.

Similarly, the area overhead of the encoder and the decoder in SEC-MAEC is significantly smaller than the one in OLSC. While the encoder of SEC-MAEC is only one-third of the encoder of OLSC, the decoder is in the size of 10% of the one in OLSC. This saving in the area overhead is proportional with the static energy consumed by the encoders and the decoder. We also compare the dynamic energies of the encoders and decoders in each supply voltage level in Figure 9. In the figure, we normalize energies to OLSC at 1V. Obviously, SEC-MEAC presents substantially less energy consumption for both encoder and decoder.

V. Related Work

In this section, we present previous error correction schemes used for reducing supply voltage in the storage structures. To the best of our knowledge, Parichute is the ECC scheme with the highest error correction capability proposed for lowering supply voltage [10]. However, it has a complex encoding/decoding structure which presents several cycles error correction latency and high energy overhead under the high failure rate. Thus, it is not convenient for detection and correction of failures in L1 caches. 2D-ECC have been proposed by Kim, et al. [7], in which ECC values of the rows and columns are saved in order to provide a strong error correction capability for L1 caches. While 2D-ECC can not tolerate very high error rate as it occurs in the near-threshold voltage level, it also requires read-modify-write operations for each updates in the cache. Alameldeen et al. [2] proposed Variable-Strength Error-Correcting Codes (VS-ECC). VS-ECC uses a simple and a fast ECC for lines with zero or one failures while it uses a strong ECC for lines with multi-bit failures. Our study differs from VS-ECC since we utilize a new ECC code (SEC-MEAC) in the very low level of V_{dd}. Wilkerson et al. [13] proposed disabling faulty words and combining two consecutive cache lines in order to form a single, non-failing cache line. Similarly, Abella et al. [1] propose disabling sub-blocks instead of words in order to provide more capacity of the cache in the low-power mode. However, these disabling



(a) Encode Energy

(b) Decode Energy

Fig. 9: Encode/Decode Energy Normalized to the energy of the encoder/decoder of OLSC at 1V

scheme can not provide any useful cache capacity when the bit failure rate is higher than 0.01%. Wilkerson et al. also proposed bit-fix in the same study as an alternative to disabling [13] in which the location of defective bits are stored in one of the cache ways. However, this scheme can correct up to 10 failing bits in a cache line among 4 ways (i.e. bit failure probability is around 0.005). Besides architectural approaches, circuit-based hardening approaches have also been proposed such as using 8T SRAM cells [11] instead of 6T cells. Although 8T SRAM cells are more stable against parameter variations, they have high area overheads which does not provide any performance gains when operating in high power mode. Yoon et al. proposed Memory-Maped ECC in which error correction data is saved in the memory hierarchy to be accessed in case of an error detection [14]. This scheme is not applicable in the near-threshold voltage execution in which all cache lines present bit failures and require error correction.

VI. Conclusion

Although downscaling the supply voltage provides a substantial energy saving in computer systems, when the V_{dd} is reduced lower than the safe operation margin, it causes drastic increase in the number of persistent failures especially in memory structures. Utilizing Error Correcting Codes (ECC) in the memory structures is the most appealing approach to reduce the supply voltage below the safe operating margin. However, ECC schemes presenting high encoding/decoding overhead diminishes the performance of the system. In this study, we utilize a fast and low-complexity SEC-MAEC code in L1 caches under scaling supply voltage. We also demonstrate how to organize 4-way cache in the architecture level in order to maximize the cache capacity.

Acknowledgements

We would like to give special thanks to Salvatore Pontarelli for his guidance and help. This work is part of a collaboration in the framework of COST ICT Action 1103 “Manufacturable and Dependable Multicore Architectures at Nanoscale”. The research leading to these results has also received funding from the European Community’s Seventh Framework Programme [FP7/2007-2013] under the ParaDIME Project grant agreement n° 318693. This study is also partially supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under research Grants 112E004.

References

[1] J. Abella, J. Carretero, P. Chaparro, X. Vera, and A. González. Low Vccmin Fault-Tolerant Cache with Highly Predictable Performance. In *MICRO*, pages 111–121, 2009.

[2] A. R. Alameldeen, I. Wagner, Z. Chishti, W. Wu, C. Wilkerson, and S.-L. Lu. Energy-efficient Cache Design Using Variable-strength Error-correcting Codes. In *Proceedings of the 38th Annual International Symposium on Computer Architecture*, pages 461–472, 2011.

[3] Z. Chishti, A. Alameldeen, C. Wilkerson, W. Wu, and S.-L. Lu. Improving Cache Lifetime Reliability at Ultra-Low Voltages. In *MICRO*, pages 89–99, 2009.

[4] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger. Dark Silicon and the End of Multicore Scaling. In *Proceedings of the 38th International Symposium on Computer Architecture (ISCA)*, 2011.

[5] M. Y. Hsiao, D. C. Bossen, and R. T. Chien. Orthogonal Latin Square Codes. *IBM Journal of Research and Development*, 14(4):390–394, July 1970.

[6] S. Jain, S. Khare, S. Yada, V. Ambili, P. Salihundam, S. Ramani, S. Muthukumar, M. Srinivasan, A. Kumar, S. Kumar, R. Ramanarayanan, V. Erraguntla, J. Howard, S. R. Vangal, S. Dighe, G. Ruhl, P. A. Aseron, H. Wilson, N. Borkar, V. De, and S. Borkar. A 280mV-to-1.2V wide-operating-range IA-32 processor in 32nm CMOS. In *IEEE International Solid-State Circuits Conference*, pages 66–68, 2012.

[7] J. Kim, N. Hardavellas, K. Mai, B. Falsafi, and J. Hoe. Multi-bit Error Tolerant Caches Using Two-Dimensional Error Coding. In *MICRO*, 2007.

[8] J. Kulkarni, K. Kim, and K. Roy. A 160 mV Robust Schmitt Trigger Based Sub-threshold SRAM. *IEEE Journal of Solid-State Circuits*, 42(10):2303–2313, October 2007.

[9] B. Maric, J. Abella, and M. Valero. APPLE: Adaptive Performance-Predictable Low-Energy Caches for Reliable Hybrid Voltage Operation. In *The 50th Annual Design Automation Conference*, page 84, 2013.

[10] T. Miller, R. Thomas, J. Dinan, B. Adcock, and R. Teodorescu. Parichute: Generalized Turbocode-Based Error Correction for Near-Threshold Caches. In *MICRO*, pages 351–362, 2010.

[11] Y. Morita, H. Fujiwara, H. Noguchi, and Y. Iguchi. An Area-Conscious Low-Voltage-Oriented 8T-SRAM Design under DVS Environment. *IEEE Symposium on VLSI Circuits*, pages 256–257, June 2007.

[12] P. Reviriego, S. Pontarelli, J. Maestro, and M. Ottavi. Low-cost Single Error Correction Multiple Adjacent Error Correction Codes. *IEEE Electronic Letters*, 48:1470–1472, 2012.

[13] C. Wilkerson, H. Gao, A. Alameldeen, Z. Chishti, M. Khellah, and S. lien Lu. Trading off Cache Capacity for Reliability to Enable Low Voltage Operation. In *ISCA*, pages 203–214, 2008.

[14] D. Yoon and M. Erez. Memory Mapped ECC: Low-Cost Error Protection for Last Level Caches . In *Proceedings of the 36th International Symposium on Computer Architecture*, 2009.