DUO: Exposing On-chip Redundancy to Rank-Level ECC for High Reliability

Seong-Lyong Gong[†] Jungrae Kim[‡] Sangkug Lym[†] Michael Sullivan[*] Howard David[⋆] Mattan Erez[†]

[†]UT Austin [‡]Microsoft [*]NVIDIA [⋆]Huawei

[†]{sl.gong, sklym, mattan.erez}@utexas.edu [‡]jungrkim@microsoft.com [*]misullivan@nvidia.com [⋆]howard.david@huawei.com

***Abstract*—DRAM row and column sparing cannot efficiently tolerate the increasing inherent fault rate caused by continued process scaling. In-DRAM ECC (IECC), an appealing alternative to sparing, can resolve inherent faults without significant changes to DRAM, but it is inefficient for highly-reliable systems where rank-level ECC (RECC) is already used against operational faults. In addition, DRAM design in the near future (possibly as early as DDR5) may transfer data in longer bursts, which complicates high-reliability RECC due to fewer devices being used per rank and increased fault granularity.**

We propose dual use of on-chip redundancy (*DUO*), a mechanism that bypasses the IECC module and transfers on-chip redundancy to be used directly for RECC. Due to its increased redundancy budget, DUO enables a strong and novel RECC for highly-reliable systems, called *DUO SDDC*. The long codewords of DUO SDDC provide fundamentally higher detection and correction capabilities, and several novel secondary-correction techniques integrate together to further expand its correction capability. According to our evaluation results, DUO shows performance degradation on par with or better than IECC (average 2–3%), while consuming less DRAM energy than IECC (average 4–14% overheads). DUO provides higher reliability than either IECC or the state-of-the-art ECC technique. We show the robust reliability of DUO SDDC by comparing it to other ECC schemes using two different inherent fault-error models.

***Keywords*-DRAM reliability; scaling errors; ECC;**

I. Introduction

As DRAM technology continues to scale, new reliability challenges are emerging—*scaling-induced inherent faults* occur during manufacturing and persist such that they may later compound with *operational faults* that accumulate during operation. Although inherent faults already exist in current DRAM technology, their rate is low enough to be tolerated with traditional row and column sparing techniques [1, 2]. However, sparing cannot tolerate the high inherent fault rate expected as DRAM technology continues to scale [3, 4, 5].

In response to this new challenge, both industry and academia are exploring alternatives to row and column sparing. One such alternative is *In-DRAM error checking and correcting* (IECC) codes [4, 6, 7, 8, 9, 10, 11]. With IECC, storage and ECC logic are embedded in each DRAM device to encode data written into DRAM and correct it before transmission back over the memory channel. IECC presents a (mostly) unchanged DRAM external interface and it does not expose scaling errors outside of the DRAM itself. This is why IECC is appealing despite the additional on-chip redundancy with 6.25% or 12.5% capacity overheads.[1] However, IECC presents inefficiencies because of how it interacts with internal DRAM operations and, because highly-reliable systems must also rely on *rank-level ECC* (RECC) with its own redundancy for tolerating severe operational faults such as device failures. For some DRAM organizations, the IECC codeword length does not match the granularity of data transfer, thus requiring more data to be fetched within the DRAM than the same design without IECC (Section III-A). Furthermore, this same mismatch turns every write operation into a read-modify-write to update the entire IECC codeword. When IECC is combined with an independent RECC, overall redundancy is higher than necessary for a given level of reliability.

We propose **d**ual **u**se of **o**n-chip redundancy (*DUO*), an alternative to IECC. DUO introduces an IECC *bypass mode* to directly transfer on-chip redundancy over the memory channel, exposing this redundancy to the RECC. DUO can be implemented without any significant changes to current DRAM design, while avoiding the performance and energy overheads of IECC (encoding/decoding, overfetch, and read-modify-write). DUO provides higher performance, energy efficiency, and reliability than prior work—reliability improves because instead of investing in weak in-DRAM protection, DUO invests all the available DRAM redundancy in a single powerful code. DUO presumes that DRAM vendors will include on-chip redundancy and IECC because it is necessary for low-cost systems and because array designs are common across segments. The DUO techniques are applicable to configurations without on-chip redundancy that have increased rank-level chip redundancy instead. However, due to space constraints we are unable to discuss this scenario.

We discuss DUO protection for different DRAM configurations, including for lower-reliability systems (i.e., non-ECC DIMMs), high-reliability systems and future interfaces with narrower ranks as expected with DDR5. For high reliability, combining the on-chip redundancy with rank-level chip redun-

Appears in HPCA 2018.

[1]Although this is a design choice, 6.25% ECC redundancy is preferred because of the cost sensitivity of the DRAM industry; 12.5% redundancy results in >10% area overhead [3].

	IECC +SDDC [14]	XED [13]	DUO
IECC (6.25%)	SEC	SEC	Bypass
Encoding/decoding	yes	yes	no
Overfetch	yes	yes	no
Read-modify-write	yes	yes	no
Cooperation by exposing	N/A	IECC result	IECC redundancy
RECC	RS(18,16)	Parity	RS(76,64),BCH
SDDC method	SSC / codeword	RAID-3	Burst erasure
Coverage overlap w/ IECC	yes	yes	no

Tab. I: Comparison with prior work. IECC provides weak protection (single-bit error correction) and degrades performance. DUO bypasses IECC and uses longer codewords and novel decoding techniques. DUO requires extra bus beats, similar to optional CRC in DDR4 [15]. RS and SSC stand for Reed–Solomon and *single symbol correction*.

dancy across a rank enables a novel and strong RECC—*DUO SDDC*. By carefully utilizing the full available redundancy in a coordinated manner, DUO SDDC simultaneously tolerates both inherent and severe (chip-level) operational faults.

DUO effectively unifies IECC and RECC by exposing on-chip redundancy in order to achieve higher reliability while avoiding the overhead of IECC (Tab. I). This is very different from some recent work that proposes cooperation between IECC and RECC instead. Son et al. [12] show that a poor combination of RECC and IECC offers lower reliability than expected from the RECC. XED [13] suggests that an efficient *single device data correcting (SDDC)*[2] using the IECC primarily for detection, and a parity-based RECC along with complex diagnostics procedures for correction. XED exposes the outcome of IECC detection to the RECC but not the redundant information itself. The error-coverage of XED against simultaneous errors in multiple devices is hampered by the weak IECC, which offers weak per-device correction with a single-error-correcting (SEC) code. In terms of runtime overhead, XED requires additional checks for handling overlapped errors. These procedures are time-consuming and rely on untested, yet very logical, assumptions about the distribution of errant bits.

In summary, we make the following key contributions:

- We propose DUO for stronger and more performant protection than solutions that rely on in-DRAM error checking and/or correcting. DUO bypasses IECC mechanisms, exposes on-chip redundancy to the memory controller, and combines the internal redundant bits with the redundancy of the RECC. DUO enables the RECC to efficiently handle both inherent and coarse-grained operational faults simultaneously. This is unique from all prior approaches that propose some awareness between the IECC and RECC.
- We describe and evaluate a novel decoding scheme for DUO SDDC that incorporates burst-erasure decoding based on standard RS code to better tolerate scenarios with overlapping coarse-grained operational faults and bit-level inherent faults with a novel half-symbol correction scheme for fully utilizing all redundant bits to further increase fault coverage.
- We introduce a new reliability challenge of strong protection for future DRAM. We show that DUO SDDC achieves far higher reliability than previously-published ideas that also use a single redundant device per rank (up to 1650× lower DUE and 15300× lower SDC probabilities than XED at high inherent fault rates).
- We evaluate the performance impact and the improved reliability of DUO. DUO SDDC exhibits both better performance and better power efficiency when compared to IECC [3, 4, 9, 11] and the state of the art XED [13]. This efficiency advantage is a result of avoiding the mismatch between IECC codeword length and the DRAM interface width; at the same time, DUO SDDC offers higher reliability than either IECC or XED.
- We evaluate a range of DRAM organizations, including ones without ECC DIMMs, and show that DUO is far superior to alternatives in all cases.
- We show the reliability robustness of DUO by evaluating it and comparing to other ECC schemes with two different models for inherent faults; one model follows prior work and treats all inherent faults as permanent while the other is inspired by research studies of scaling faults [11, 16, 17, 18] and it models inherent errors as a combination of permanent and intermittent faults.

[2]SDDC is also referred to as chipkill, extended ECC, and ChipSpare by IBM, Oracle, and HP, respectively.

II. Background

A. Current DRAM Reliability Challenges

Terminology. An *error* is a discrepancy between the intended and actual state of a system, and a *fault* is a physical phenomenon or a defect that may cause an error [19]. A *transient* fault is introduced by some temporary environmental impetus (e.g., a high-energy particle strike), whereas a *permanent* fault is an irreversible physical defect that continually produces errors (e.g., a stuck-at bit) [20]. Permanent faults that manifest errors at irregular intervals are called *intermittent*. A severe error that exceeds the correction capability of an ECC code can be flagged as a *Detectable but Uncorrectable Error (DUE)*. In the case of miscorrection or missed detection, incorrect data escapes from the error control system, resulting in a possible *Silent Data Corruption (SDC)*. DUE and SDC will typically result in disruption or failure to the systems. For computing systems, *reliability* indicates the continuity of service without a failure [19], often measured in *Failures In Time (FIT*—1 failure in 1 billion hours).

Inherent faults. DRAM scaling induces more inherent faults within a DRAM array by causing a greater fraction of cells to not operate within their design parameters. There are several reasons why scaling the already small one-transistor one-capacitor DRAM cell array increases the incidence of manufacturing faults. For example, increased aspect ratio due

to scaling makes maintaining the required charges for correct operation more difficult [4, 5]. In addition, the narrower pitch due to scaling can result in more disturbance errors [21]. According to a recent industry study, the inherent fault rates in 20nm and sub-20nm DRAM technologies may already be within 10^{-4}–10^{-6} [3]. Recent studies [5, 12] have shown that it is challenging to efficiently remap this increasing number of inherently faulty cells, even assuming that all inherent faults are permanent and can be identified during test. Specifically, a dramatic increase in the area overhead (from <10% to about 40%) is expected when the inherent fault rate increases from 10^{-7} to 10^{-6} if the inherent faults are handled by a remapping scheme [3]. This prohibitive area overhead will likely prevent inherent faults from being handled by sparing alone.

While the details of inherent faults are proprietary information, prior studies have observed that they may manifest as random bit errors [17, 18, 22] due to the *variable retention time* (VRT) phenomenon. Some DRAM cells occasionally discharge too rapidly for a fixed refresh interval, generating retention errors. These intermittent inherent faults generate errors randomly and infrequently, so they could not be detected during the testing phase of manufacturing. If faults are not identified before they cause errors, sparing becomes ineffective. Retention errors due to VRT are already a source of random DRAM errors and they are expected to increase in finer-pitch DRAM technologies [3]. If future scaling trends exacerbate VRT and induce more frequent intermittent inherent faults, we can expect their impact on DRAM reliability to be even more severe then their increased rates imply [23].

IECC. Academics and DRAM vendors are considering incorporating ECC in the DRAM chips themselves (IECC) [4, 6, 7, 8, 9, 10, 11]. When writing data, IECC generates ECC check bits internally within each DRAM chip and stores them into redundant array storage. This redundant storage resides within each chip and is likely associated with each memory mat/tile (though other organizations are possible). When reading data, IECC also reads the check bits and attempts to correct any errors. The check bits are not transferred out of the chip, and used to correct single-bit errors and (hopefully) transfer data that is free from scaling-induced errors to the memory controller. Despite its costs and limitations, IECC has been considered for low-power commodity DRAMs to improve yield and decrease operational voltage [3, 6, 7, 8, 10, 11], as well as for refresh reduction [8, 24].

Operational faults. Operational faults are also a significant concern for highly-reliable systems [20, 25, 26, 27, 28]. Field measurements on the 1.75 PFLOP Jaguar supercomputer, for example, show one DRAM operational error every 10 seconds [20]. Larger systems that are used in datacenters and future supercomputers will likely suffer from even more frequent DRAM errors as such systems aggressively increase memory capacities to scale with growing workloads [29]. While most operational faults manifest as single bit errors (e.g., 91% in IBM servers [30]), high-performance computers require stronger protection against coarse-grained faults such as device failures. SDDC can restore the data from a completely failed device within a rank and improve memory reliability significantly by correcting 99.94% of all errors [30] and achieving 42× better uncorrectable error rate than common bit-level ECC [20]. As a result, SDDC has become the de facto memory reliability standard and most high reliability systems employ SDDC protection to meet their reliability goals [14, 27, 30, 31, 32, 33, 34, 35, 36].

B. Related Work

Long codeword ECC. In general, an ECC with a longer codeword provides higher reliability than one with a shorter codeword. For example, BambooECC [37] uses a RS code with a codeword at the cache-line granularity. Long codewords provide far safer detection (flagging virtually 100% of errors). Because of a symbol layout that matches DRAM internal structure, BambooECC can handle up to 4 independent pin faults or a coarse-grained fault such as a device failure. For example, BambooECC can correct up to 4 error symbols from different devices that cannot be corrected by conventional SDDC schemes. Thus, the overall reliability of BambooECC is higher than that of other SDDC schemes. Besides inheriting longer codeword idea, DUO incorporates novel decoding techniques to efficiently handle coarse-grained operational faults as well as multiple inherent faults.

Coordination with IECC. XED [13] proposes cooperation between IECC and RECC to provide efficient SDDC protection against device failure. XED repurposes the IECC to detect errors from each device so as to correct any detected errors by using a RAID-like parity code at the rank-level. In the common case, only a single device reports an error, making RAID correction efficient and simple. Unlike DUO, XED does not expose the redundant information directly to the rank level. Instead, when the basic correction flow above fails, for example, if more than a single device reports an error, XED resorts to quite-complex protocols because the rank-level parity correction that relies on a single known error location will fail. First, if multiple devices report errors, XED *blocks memory traffic to the rank* and switches it to In-DRAM single-bit error correction (SEC) mode. Each device attempts its own correction, which is then verified by the rank-level parity. Second, achieving SDDC-level correction in the presence of scaling errors is more complex with XED because the IECC cannot correct, and sometimes not even detect, such errors. To solve this, XED enters a diagnostics mode that attempts to discern which chip (or row) failed; the diagnostics reads, at least, the entire row buffer and takes a minimum of $128 \times 5ns(tCCD_L) = 0.64ms$.

To summarize, XED aims at efficient SDDC protection utilizing IECC with small changes in memory interface and protocols, such as *catchword* and *XED-enable*. Our evaluation shows that the effectiveness of XED's decoding strategy, with

IECC[3] detecting failed chip or correcting only single bit errors, is insufficient to achieve high reliability in important scenarios. Rather than relying on IECC, DUO rearchitects on-chip redundancy and adds novel decoding ideas to achieve high reliability; rearchitecting on-chip redundancy leads to extra transfers, which can be easily incorporated within memory interfaces; e.g., DDR4 already supports multiple burst lengths for optional write CRC.

III. Motivational Challenges

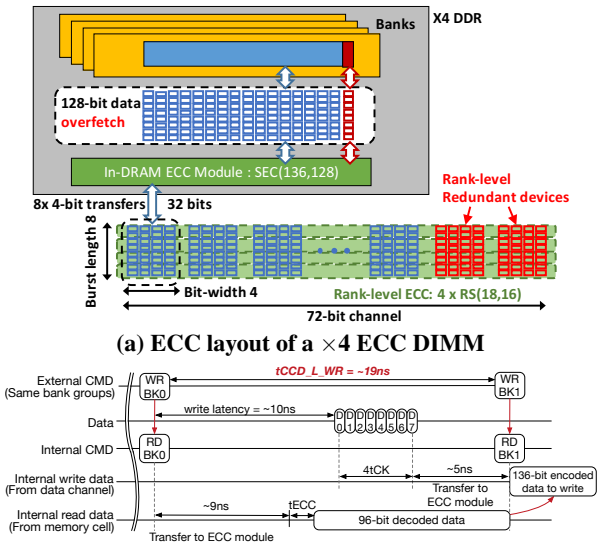
A. Costs and Limitations of IECC

Costs. While IECC is appealing in that the DRAM interface is essentially free from any scaling-induced errors, it suffers from significant overheads from encoding and decoding the on-chip ECC check bits. Redundant check-bit storage and ECC logic blocks are necessary within each chip; more importantly, IECC decoding degrades performance and energy efficiency [4, 8]. First, IECC decoding increases memory column command timing parameters to provision time for the ECC decoding logic.

Second, a mismatch between the DRAM interface width and the IECC codeword length can exist for commonly-used narrow chips. Each DRAM chip contributes only a small number of bits to each memory transfer—32 bits for a ×4 DDR4 DRAM and 64 bits for a ×8 device, whereas a longer 136-bit IECC codeword has a low 6.25% storage redundancy. For a (136,128) single-bit error correcting (SEC) code, therefore, 136 bits need to be fetched for every 32-bit access in a ×4 DRAM chip (Fig. 1a).[4] This internal overfetch consumes more power by activating and reading more cells for both read and write commands [4].

Lastly, the internal overfetch imposes additional delays for write commands due to read-modify-write operations. For a write operation, first the old data must be decoded by IECC after fetching via an internal read command, which takes several nanoseconds (Fig. 1b). This is because 96 bits must be fetched from the old codeword to encode and write the new 136-bit codeword (*read-modify-write*). Hence, the latency for consecutive writes to the same bank group ($tCCD_{L\_WR}$) roughly doubles. Due to the above reasons, IECC reduces performance of memory-intensive applications by 5-10% on average, even with optimistic DRAM timing parameters [3].

Limitations. It is hoped that IECC is an effective solution to address inherent faults, assuming that RECC can still be used for operational faults. For highly-reliable systems, however, IECC can amplify errors by miscorrecting them, potentially increasing reliance on the RECC. Note that IECC is innately weaker than RECC because of its per-device operation and short codewords. In fact, for some choices of IECC and weak RECC, protection may actually worsen [12].



(a) ECC layout of a ×4 ECC DIMM

(b) Timing diagram

Fig. 1: IECC. (a) IECC results in overfetch. For each 32-bit data, a chip should read and write a 136-bit codeword due to IECC encoding/decoding (data size mismatch). At the rank-level, a memory transfer block (of 512-bit data) can be encoded as four 8-bit RS(18,16) codewords (green dotted boxes) for SDDC [14]. (b) IECC results in read-modify-write overhead. Since a 32-bit data write modifies IECC redundant bits, the remaining IECC codeword should be read first. Thus, a longer column-to-column delay ($tCCD_{L\_WR}$) is needed.

Error accumulation. Scaling errors may be intermittent (e.g., VRT errors) and can thus accumulate over time, unless they are corrected in a timely manner. Memory scrubbing is a common technique for periodically reading memory and correcting any observed errors to prevent such accumulation [38]. Since IECC can correct single bit errors when reading data, it can provide partial scrubbing. However, the effect of this in-DRAM scrubbing is limited to the read and overfetched data during operation and it is unlikely that all error accumulation can be eliminated by reads and writes alone. While it appears that refresh operations are similar to reads, refresh cannot be used with IECC for scrubbing memory. For cost reasons, IECC decoders are likely placed at or near the I/O slice with likely one decoder per bank, or possibly one per bank group [3, 11]. Refresh operations are handled entirely within the mats (DRAM sub-arrays) and do not move data to the bank edge. In-DRAM scrubbing requires this extra quite-costly movement. Furthermore, each refresh is for multiple rows (2M bits per refresh for 16Gb DRAM) but decoder throughput is at most 128b per access, and thus scrubbing simply cannot be done at refresh rate. Instead, any In-DRAM scrubbing is likely to have a period of $2Mb/128b * 64ms = 1048s$. RECC, including DUO, can scrub at such a period with <0.1% impact on memory throughput and power.

[3]The IECC used by XED is the ATM8-CRC code, which can detect errors well and correct a single error. While Nair et al. [13] use it as a (72,64) code with 12.5% on-chip redundancy, our evaluation uses the same code design with 6.25% as a (136, 128) code.

[4]Each subarray contributes 8 (or 4) bits [37] and 136-bit overfetch can be done by accessing 16 (32) data and 1 (2) redundant subarrays.

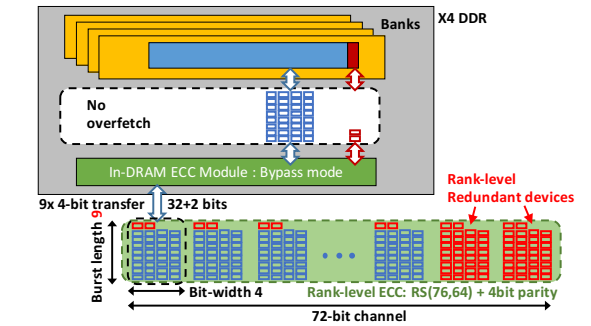
B. Reliability Challenges for Future Interfaces

The maximum data transfer rate of DRAM interfaces has continuously increased over time, whereas the maximum internal array frequency has remained fixed (at about 200MHz) [15, 39, 40]. Unless addressed, this frequency disparity results in a performance penalty as a result *bubbles* between data transfers. These bubbles can be avoided by increasing the data transfer granularity. More bits are accessed internally in parallel and then transferred as a burst over the fast interface. The burst length, also called the DRAM prefetch depth, doubled between DDR and DDR2 as well as between DDR2 and DDR3. An alternative solution to coarser access granularities is bank grouping, which was adopted in the GDDR5 and DDR4 standards. Bank grouping enables bubble-less data transfer so long as back-to-back transfers target banks in different bank groups. However, bank grouping can degrade performance considerably by requiring a longer column-to-column delay ($tCCD_L$) within the same bank group.

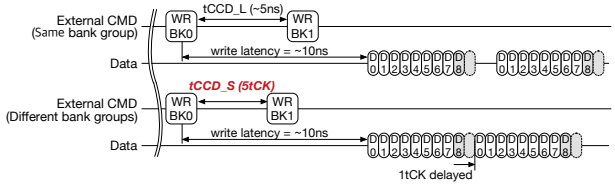
Because the DDR3 to DDR4 transition did not involve deeper DRAM prefetch, it is quite possible that the DDR4 to DDR5 transition will. Assuming prefetch size is doubled and that others parameters remain the same (e.g., a cache line size of 64B and the same width for each DRAM device), the number of data devices in a rank will be halved. This implies that the number of redundant devices in a rank must also be halved to maintain the 12.5% redundancy level; only a single ×4 redundant device will be available with a 32-bit wide rank and a 64B access granularity. This poses a challenge in achieving high reliability as seen today with ×8 DDR4-generation ranks, which cannot provide true SDDC protection. For example, current *chipkill-level*, SDDC protection techniques will require more than 12.5% rank-level chip redundancy if a rank is only composed of 8 burst-16 data chips [14, 30, 31, 32]; in fact, the necessary redundancy for this organization grows to 25% (10 x4 chips per rank overall). Alternatively, the access granularity may increase to 128B, but such coarse granularity results in poor bandwidth utilization for many applications [41, 42, 43]. In the rest of the paper, we refer to this speculative DDR5 configuration as DDR5*.

IV. Dual Use of On-chip Redundancy

We present **d**ual **u**se of **o**n-chip redundancy (*DUO*), which repurposes the on-chip redundant bits thus incorporates them within a strong RECC. DUO enables higher reliability than IECC and other state of the art ECC protection schemes (e.g., XED[13]). DUO achieves this high reliability with three unique features: (1) DUO bypasses IECC so that the on-chip redundancy is exposed directly at the rank-level ECC, where it can be most efficiently used, (2) DUO utilizes long codewords that guarantee high detection capability and offer unified corrections at the cache-line granularity, which are unconstrained by error patterns, and (3) DUO supports novel ECC designs to expand its correction capability.



(a) ECC layout of a DDR4 ×4 ECC DIMM



(b) Timing diagram

Fig. 2: DUO. (a) Bypassing IECC avoids overfetch. Instead, an extra transfer burst is necessary to transmit the on-chip redundancy (32+2 bits are read and transferred from each chip). At the rank-level, the transferred on-chip redundant bits enable to encode a longer RS(76,64) codeword (details in Section 4.3). (b) Bypassing IECC results in an extra transfer, which corresponds to one additional cycle (two additional beats). This extra transfer degrades performance only for consecutive accesses to the different bank groups.

DUO is not limited to a specific ECC scheme and the DUO concept itself is about how to use on-chip redundancy in an alternative way. We discuss how DUO can be used for SDDC protection with, and without, rank-level chip redundancy and how DUO provides flexibility in the use of redundancy even in configurations that only aim for a low level of reliability. For high-reliability systems that aim to tolerate severe errors, including device failure, DUO SDDC provides SDDC protection with symbol-based Reed–Solomon (RS) codes. For lower-reliability systems that only aim to tolerate scaling errors (i.e., similar to configurations today that cannot tolerate operational faults), DUO VRT (VRT is a primary cause of scaling errors) focuses on single-bit correction with Bose–Chaudhuri–Hocquenghem (BCH) codes [44] (Section IV-B).

A. IECC Bypass

Prior work uses the on-chip redundant bits for internal single-bit error correction (or internal detection for XED). Hence, data is transferred to the memory controller after only weak correction/detection. Instead of investing in weak protection within each DRAM device, DUO uses the same amount of on-chip redundancy as an integral part of the RECC. This alternative use of on-chip redundancy can be obtained by adding an IECC *bypass mode* to the DRAM interface.

In bypass mode, IECC is disabled and the on-chip redundant bits associated with each memory access are read or written directly by the memory controller. Since only a few redundant

bits are transferred (Fig. 2a), the bypass mode requires a 9-beat burst rather than the original 8-beat burst or a 17-beat rather than a 16-beat burst with DDR5*. The double data-rate (DDR) bus design may not be compatible with an odd beat count. Back-to-back reads or writes need to be on whole clock boundaries, so only turn-around cycles (R-W, W-R, rank-to-rank) with a half clock of slack time will benefit from an odd burst length. Hence, we evaluate DUO a bit pessimistically with a 10-beat burst (Tab. VI).

Implementation. Bypass mode is implementable and the vast majority of design details (including DRAM addressing) remain the same. DRAM internal implementations (mats and datapath to IECC encoder) must already support on-chip redundancy, and thus only a small extension to the I/O serializer datapath (with data masking) is needed for DUO. Subarrays can be selectively accessed by a simple MUX and redundancy can be masked. Instead of IECC encoder output, this masked data in bypass mode is transferred through the I/O serializer. We expect that these implementations are straightforward though the details strongly depend on internal DRAM designs. Note that current DDR4 DRAM already supports 10-beat bursts with write CRC, and DUO bypass is conceptually similar. A DUO-enabled system does not rely on IECC for reliability. We therefore expect that DRAM vendors will choose to fuse Bypass vs. IECC mode in the factory; IECC is still important for consumer-oriented systems and as an alternative solution and the differences between the two chips are minimal. This practice is common in the DRAM industry [45, 46, 47].

Performance trade-offs. DUO requires additional bus transfers, but also saves overheads associated with IECC encoding and decoding. The evaluation results we present later suggest that the performance benefits outweigh the bandwidth overheads. Although every DRAM read or write requires one additional bus cycle, the additional cycle rarely has a significant impact on performance (Fig. 2b). This is not only because bandwidth does not always bound performance, but also because consecutive accesses to the same bank group have no overhead (i.e., $tCCD_L$ remains the same). The additional cycle only affects consecutive accesses to different bank groups ($tCCD_S = 5tCK$). While not always adding overhead, DUO avoids the decoding latency of IECC, which takes several nanoseconds [3]. More importantly, the bypass mode solves the size mismatch problem between data transfer and IECC codeword granularities, which is the reason for its internal overfetch and performance and power degradation.

Wide I/O interface IECC bypass is the general concept of redirecting on-chip redundancy to be used at the rank level. For a DDR-like off-package memory interface, we propose and evaluate transferring on-chip redundancy using an additional bus transfer (lengthening the transfer burst). However, for an on-package interface, like HBM, extra TSVs could be used, which is preferable as HBM burst lengths are short. Thus, bypassing IECC does not result in prohibitively high overheads even in a wide I/O interface. Note that IECC does not suffer from overfetch with a wide I/O interface. However, exposing the on-chip redundancy allows for more flexible and customized ECC at the memory controller.

redundancy		6.25%		12.5%	
# of errors	result	In-DRAM 4·SEC	DUO BCH TEC	In-DRAM 8·SECDED	DUO BCH HEC
1	CE	100.00%	100.00%	100.00%	100.00%
2	CE	74.99%	100.00%	87.49%	100.00%
	DUE	2.93%	0.00%	12.51%	0.00%
	SDC	22.08%	0.00%	0.00%	0.00%
3	CE	37.43%	100.00%	65.60%	100.00%
	DUE	7.63%	0.00%	33.52%	0.00%
	SDC	54.94%	0.00%	0.88%	0.00%
4	CE	9.33%	0.00%	41.04%	100.00%
	DUE	13.09%	99.99%	55.84%	0.00%
	SDC	77.58%	0.01%	3.12%	0.00%
5	CE	0.00%	0.00%	20.48%	100.00%
	DUE	18.65%	98.75%	73.70%	0.00%
	SDC	81.35%	1.25%	5.82%	0.00%
6	CE	0.00%	0.00%	7.68%	100.00%
	DUE	24.14%	99.99%	84.95%	0.00%
	SDC	75.86%	0.01%	7.37%	0.00%

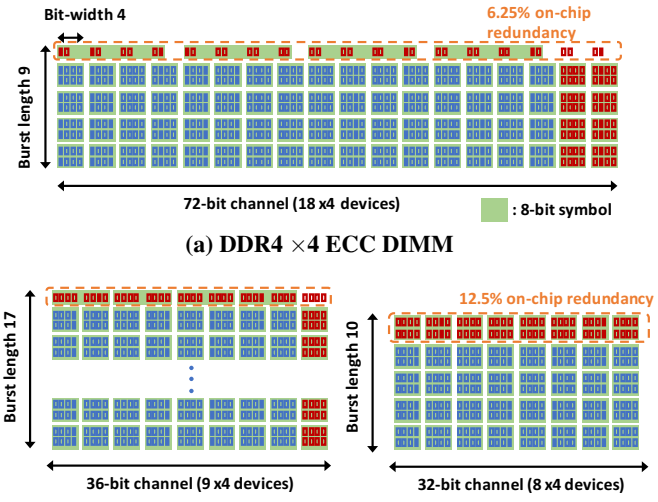
Tab. II: Error coverage of DUO VRT. (CE: corrected error)

B. DUO VRT

DUO can achieve higher reliability than IECC even for lower-reliability systems where no rank-level redundancy exists. DUO enables unified multi-bit correction for each memory transfer block. We proposed to use BCH codes [44] and call this ECC option DUO VRT because it is intended primarily for scaling errors. Suppose a DDR4-like $\times 16$ or DDR5*-like $\times 8$ non-ECC DIMM is used. If IECC is used, in terms of reliability, up to 4 single-bit errors can be corrected (single-bit error correction per 136b IECC codeword). However, these 4 errors must be distributed across the SEC codewords; double-bit errors within a codeword are uncorrectable (and some may even be miscorrected). That is, a maximum of 4 bit-errors can be corrected under the constraint that no two errors are located in the same codeword. With DUO, a *triple-bit error correcting* (TEC) BCH code is possible with the same 32 redundant bits for every 512 data bits transfered. In general, a t-correcting BCH code of length upto $2^m - 1$ requires $m \times t$-bit redundancy. Hence, 30 out of the 32 redundant bits can be used for the TEC BCH code. The TEC correction is more flexible and any three bit-errors within a memory transfer can be corrected. The remaining 2 bits can be used to improve detection coverage outside the primary BCH correction (Section V-B). While it may seem that TEC DUO VRT is inferior, as it can only correct three errors while IECC can correct 4, the flexibility in where those errors occur makes DUO VRT far more reliable. Tab. II compares the error-coverage against multi-bit errors of DUO VRT and IECC, assuming scaling errors are uniformly randomly distributed.

C. DUO SDDC

An even more compelling use of DUO is for highly-reliable memory. We design DUO SDDC, which uses RS codes combined with novel decoding techniques to expand error



(a) DDR4 ×4 ECC DIMM

(b) DDR5* X4 ECC DIMM (c) DDR4 ×4 non-ECC DIMM

Fig. 3: DUO codeword layouts for a memory transfer block. (a-b) With 6.25% on-chip redundancy, DUO SDDC uses 8b RS(76,64) and 4b parity between on-chip redundancy. (c) With 12.5% on-chip redundancy, DUO SDDC for DDR4 ×4 non-ECC DIMM uses 8b RS(72,64).

coverage. In the rest of this section, we describe how the codewords are organized and discuss the capabilities of the standard RS codes used as the *primary correction*.

DUO SDDC codeword layout. DUO SDDC efficiently protects DRAM against both fine-grained inherent faults and coarse-grained operational faults. For this purpose, DUO SDDC uses a long single ECC codeword for each memory transfer block. In the case of 6.25% on-chip and 12.5% rank-level chip redundancy, this corresponds to 512-bit data and 100-bit redundancy (Fig. 2a). For example, DUO SDDC for a DDR4 ×4 ECC DIMM can use an RS(76,64) 8-bit symbol code with a single codeword spanning the entire transfer block; each codeword is comprised of 64 data symbols and 12 check symbols (Fig. 3a). Each codeword does not include an additional 4 bits (6.25% on-chip redundancy from the redundant chips) that does not form a full symbol. This 4-bit half-symbol redundancy is utilized to improve correction coverage (Section V-B). This DUO SDDC codeword organization is flexible and applies to DDR4 channels with ×4 or ×8 devices as well as DDR5* ×4 configuration (Fig. 3b). A similar layout with, an RS(72,64) code, applies to DDR4 ×4 non-ECC DIMMs with 12.5% on-chip redundancy (Fig. 3c).

Primary correction. The RS decoding is the primary correction of DUO SDDC. In general, RS(n,k) can correct up to $\lfloor \frac{n-k}{2} \rfloor$ error symbols—simply, two check symbols are needed to correct one errors symbol. That is, an RS(76,64) codeword has 12 check symbols and a conventional RS decoder can correct up to 6 erroneous symbols. The primary correction of DUO SDDC can handle almost all error patterns from inherent and fine-grained operational faults. Even for DDR4 ×4 non-ECC DIMM with 12.5% on-chip redundancy, the redundancy corresponds to 8 check symbols and hence up to 4 independent error symbols can be corrected.

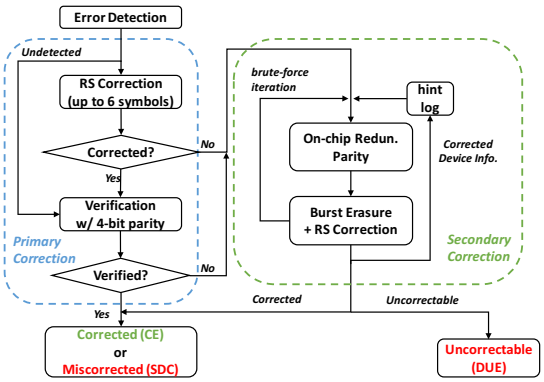


Fig. 4: Overall decoding flow of DUO SDDC. Primary correction relies on the conventional decoding of RS(76,64), and secondary correction relies on both burst decoding and on-chip redundancy parity.

However, the correction capability of primary decoding may not be enough to efficiently handle also device failures in addition to scaling errors. For a DDR4 ×4 ECC DIMM, we can correct only one more error symbol after correcting single device failure: only 2 check symbols remain if 10 check symbols are used to correct the 5 error symbols from the failed device. Furthermore, the primary correction cannot directly handle a single device failure in a DDR4 ×8 ECC DIMM or a narrow-rank DDR5* ECC DIMM (Fig. 3b) because primary correction cannot correct the 9 erroneous symbols from a failed device using its 12 check symbols (18 redundant check symbols are needed). Similarly, 8 check symbols are not enough to handle device failure with DDR4 ×4 non-ECC DIMM (Fig. 3c). We propose a novel secondary decoding to expand this limited correction capability below.

V. Expanding Correction Capability

To expand the correction capability of the conventional RS decoder, we develop a new decoding scheme for DUO. This scheme combines two techniques: (1) *burst erasure* decoding to efficiently handle coarse-grained operational faults and (2) *on-chip redundancy parity* to handle more inherent faults after a coarse-grained operational fault occurs. Secondary correction is activated when primary correction fails due to coarse-grained operational faults. Fig. 4 depicts the overall decoding flow of DUO SDDC.

A. Burst Erasure Decoding

The primary correction of DUO SDDC can correct up to six independent symbol errors. However, it cannot efficiently handle coarse-grained operational faults such as a device failure when multiple scaling errors also occur. We propose to use a *burst erasure* decoding technique for RS codes that can correct both *errors* and *erasures* efficiently [48]—in coding theory, *errors* refer to erroneous symbols at unknown locations, while *erasures* are errors whose location is known. Erasure decoding needs only one check symbol for correcting each

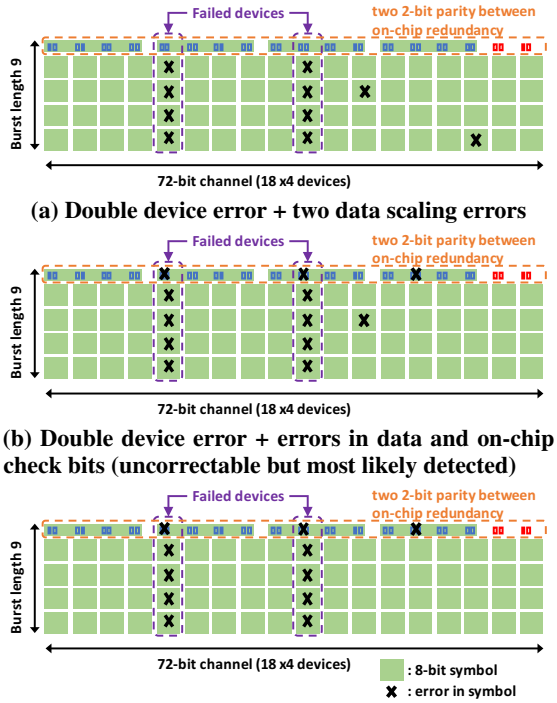
erasure. Thus, if the location of the failed device is known, all the errors from that device can be efficiently corrected as erasures. For a DDR4 ×8 ECC DIMM or DDR5* ×4 ECC DIMM, a single device failure can be handled with only 9 check symbols. Three check symbols are then available to tolerate an additional scaling error—two check symbols are enough for correction with the third used to improve detection coverage. Even for the DDR4 ×4 non-ECC DIMM with 12.5% on-chip redundancy, 5 out of 8 check symbols can be used to handle a single device failure. Again, 3 check symbols are available for tolerating scaling errors. Erasure decoding is even more effective for DDR4 ×4 ECC DIMMs; 10 check symbols can be used to correct all errors from two failed devices, and the remaining 2 check symbols can be used to additionally correct one error symbol.

To use burst erasure decoding, the locations of all erroneous symbols that are treated as erasures must be known. However, when an operational fault occurs, the location is not known. We propose to use a *brute force* decoding search: burst erasure decoding is serially attempted, with each attempt assuming a different device has failed. Intuitively, attempting erasure RS decoding [48] with a wrong device is the same as trying to correct more errors than the RS code is capable of correcting. The decoder detects such cases with extremely high probability and incorrectly "succeeds" only when the code miscorrects. The impact of such wrong locating results on reliability is negligible, as shown in our evaluation.

RAID and other parity codes blindly erase and reconstruct data based on the reported error locations and do not attempt a search. This is faster, but leads to miscorrections when an error location is wrong. As a result, prior approaches that rely on a separate detection code, including XED ([13, 43, 49]), have error-coverage holes. The magnitude of these holes depends on the strength of the code used for detection, which is often limited by short codewords and limited redundancy.

The erasure RS decoding we propose for DUO SDDC [48] is not susceptible to such miscorrections because it performs an exhaustive search on all possible erasure locations (assuming these are constrained to failed chips). Unless the tested location is correct, the decoder will report an error with high probability, eventually yielding the correct failure location. The exhaustive search fails if multiple attempts succeed (i.e., an attempt actually miscorrects), but such cases can be detected and handled as DUEs. Although not analyzed separately, all our simulation results this scenario.

Both the DDR4 ×8 ECC DIMM and DDR5* ×4 ECC DIMM require only 9 trials. The DDR4 ×4 ECC DIMM requires 18 trials to diagnose a single failed device. In the rare event that the DDR4 ×4 ECC DIMM suffers from two failed devices, it requires up to $\binom{18}{2} = 153$ trials in order to diagnose the faulty device locations. Because operational faults are very rare and the primary RS decoding of DUO SDDC suffices for scaling errors, this brute-force decoding search happens very infrequently. DUO SDDC avoids repeated searches by logging the location of the previously identified failed device. This information is used as a *hint* for where to start the erasure decoding. A wrong hint leads to an initial decoding failure, after which the full search is immediately performed. The hint requires just 5 bits per rank and is stored in the memory controller. There is no need to retain this information between boots.



(a) Double device error + two data scaling errors

(b) Double device error + errors in data and on-chip check bits (uncorrectable but most likely detected)

(c) Double device error + single scaling error in on-chip redundancy (corrected by burst erasure)

Fig. 5: Decoding scenarios with DUO SDDC. (a) a common correctable case of ORP where errors at on-chip redundancy of failed device are corrected by ORP and two inherent faults in data are corrected with saved check symbols. (b) an uncorrectable case of ORP with two scaling errors when one or both are in the check bits; such an error is uncorrectable but it is still likely detected. (c) another common correctable case where we correct two device errors and an inherent fault, even though this fault is in the check bits.

B. On-chip Redundancy Parity

DUO SDDC, as it is defined so far, cannot correct simultaneous coarse-grained operational faults and more than one inherent fault (Tab. III). Given expected future inherent fault rates, it is desirable to expand DUO SDDC to handle two inherent faults in the same codeword that also exhibits a coarse-grained operational fault. We do so in the common case by utilizing the remaining half-symbol (4 bits) redundancy in each codeword. We develop a scheme that uses these 4 bits to protect only the on-chip check bits. We call this mechanism *on-chip redundancy parity (ORP)*. ORP takes a checksum of the on-chip redundancy information across whole memory block. For the DDR4 ×4 ECC DIMM, ORP computes two 2-bit sums, each across the 4 2-bit blocks of an RS symbol stored in the in-

DRAM redundancy bits.[5] For the DDR4 ×8 and DDR5* ×4 ECC DIMMs, ORP uses the 4 bits as 4 slices of bitwise parity, with each chip contributing one bit to each slice. Before each trial of burst erasure decoding, we first correct the on-chip redundancy of the corresponding device with a RAID-like scheme using this ORP checksum. If this check-bit correction succeeds, it reduces the amount of redundancy needed to correct the rest of the failed chip(s) from 5 check symbols to 4 (DDR4 ×4 ECC DIMM), or from 9 check symbols to 8 (DDR4 ×8 or DDR5* ×4 ECC DIMM). This frees up an extra check symbol (per failed chip), allowing DUO SDDC to correct up to two inherent faults in the common case in addition to coarse-grained operational faults.

One caveat of ORP is that it guarantees that up to two inherent faults can be corrected *only as long as the scaling errors are located in the data region of the memory block being decoded* (Fig. 5a). If any scaling error occurs in the on-chip redundancy, the RAID-like correction procedure fails and it is not possible to correct both scaling errors (Fig. 5b). In this case, the error is uncorrectable, and the high detection capability of DUO SDDC ensures there is only a minimal chance of silent data corruption. Then, correction without ORP is attempted (Fig. 5c). ORP does not spoil the correction capability of burst erasure decoding; DUO SDDC maintains the ability to correct any single inherent fault along with a coarse-grained operational fault. In summary, ORP is an optimization that expands the correction capability by saving check symbols in the common case, yet maintains high detection capability in the worst case; overall DUO SDDC are summarized in Tab. III.

C. Decoding Latency

Expected primary decoding latency. The multi-error RS correction decoding uses the Berlekamp–Massey (BM) algorithm [50], or a derivative of it. This algorithm is iterative and effectively computes information for one error per iteration (in technical terms, generates the error polynomial one degree per iteration). The algorithm is serial and its runtime is proportional to the number of corrected error symbols [37]. While the worst-case primary decoding of DUO SDDC is thus 6 times higher than current ECC schemes (taking up to 12 memory cycles), the expected decoding latency is small because multi-symbol errors are very rare. With inherent fault rates of 10^{-5}–10^{-6}, 99.4%–99.94% of all 64B data blocks are expected to not have any inherent faults. Furthermore, most erroneous memory blocks have just one or two symbol errors since the likelihood that multiple errors occur within the same memory transfer block decreases exponentially as the number of simultaneous errors increases. Tab. IV shows the estimated expected overall decoding latency of DUO SDDC, which is effectively equivalent to that of current SDDC.

[5]Reconstructing two 2-bits needs exhaustive 4 trials to infer an error value for some cases (e.g., two 2-bits from the same RS symbol).

	×4	DDR4 ×8 or DDR5* ×4	×4 non-ECC
RS decoding	**6E** or **1C + 1E**	**6E**	**4E**
+ Burst erasure	**2C + 1E**	**1C + 1E**	**1C + 1E**
+ ORP	**2C + 2E***	**1C + 2E***	**N/A**

* for most errors (2-symbol common but 1-symbol worst-case correction)

Tab. III: The correction capability of DUO SDDC. C and E stand for chip failure and independent errors, respectively.

Primary		inherent fault rate 10^{-6}		inherent fault rate 10^{-5}	
# of errors	decoding overhead (*ns*)	rate	expected overhead (*ns*)	rate	expected overhead (*ns*)
0	**1.67**	**9.99e-1**	**1.67e+0**	**9.94e-1**	**1.66e+0**
1	**3.33**	**6.12e-4**	**2.04e-3**	**6.08e-3**	**2.03e-2**
2	5	1.87e-7	9.36e-7	1.86e-5	9.31e-5
3	6.67	3.80e-11	2.54e-10	3.78e-8	2.52e-7
4	8.33	5.78e-15	4.83e-14	5.75e-11	4.80e-10
5	10	7.03e-19	7.05e-18	7.00e-15	7.01e-13
6	11.67	7.12e-23	8.32e-22	7.08e-17	8.27e-16
overall expected decoding latency		1.67*ns*		1.68*ns*	

Secondary	burst erasure	brute force	
# of errors		9 devices	18 devices
Common case	3.33*ns*	109.64*ns*	2.03μs
Worst case	13.33*ns*	121.64*ns*	2.04μs

Tab. IV: The expected overhead of DUO SDDC. We assume 2 cycles (1.2Ghz I/O freq.) for detection (syndrome generation) and encoding, and 2 cycles per symbol (two iterations of BM) during RS decoding.

Expected secondary decoding latency. The decoding latency of DUO SDDC's secondary correction is also small. The combined erasure and random error RS decoding algorithm [48] has a small additional latency overhead because much of the calculation can be already performed using erasures in the initial iteration. The secondary correction can handle up to 2 scaling errors in addition to the erasures. If no scaling errors exist and the failed chip is known, the algorithm terminates after one iteration (common cse), which corresponds to a total of two DRAM cycles (3.33*ns*). Similarly, after at most another two iterations (6.67*ns*), the algorithm can terminate because any further iterations will not yield correctable errors. In the rare case that ORP fails because of a scaling error within the check bits of a functioning chip (fewer than 6.25% of scaling errors occur in this region), correction is reattempted without ORP (worst case), requiring at most another 6.67*ns*. In other words, the worst-case correction requires 13.33*ns* but nearly all require just 3.33*ns*.

Identifying the failed-chip location using the brute-force search requires 9 decoding attempts for the ×8 DDR4 and ×4 DDR5* ECC DIMMs, because each has 9 chips per rank. The initial decoding step is common to all attempts and takes 1.67*ns*. Each failed attempt takes the worst case additional 13.33*ns*. A successful attempt takes at most 13.33*ns* and 3.33*ns* at best. Hence, the brute-force search requires 109.64*ns* (or at most 121.64*ns*). This only happens once per operational fault because the location of the failure is then logged at the memory controller and used as a hint for future secondary decodings. A 122*ns* bubble that is expected to occur just once every several years for each DIMM has no impact on system operation. Similarly, the worst case latency of brute force

decoding for a $\times 4$ DDR4 ECC DIMM with one erroneous device is 18 trials and one with two erroneous devices is 153 trials—a maximum decoding latency that is just $2.04\mu s$. In contrast, the diagnostics steps proposed for XED take far longer. In addition to being untested, the expected bubble when encountering and operational fault is at least $0.64ms$—a number high enough to possibly affect some latency-sensitive operations.

VI. Evaluation Methods

We evaluate the reliability and performance impact of DUO and compare it with state-of-the-art ECC schemes. For reliability, we use a two-stage Monte Carlo simulation framework (Section VI-A) to measure the DUE and SDC probabilities over 10 billions iterations. For our performance evaluation, we use gem5 [51] integrated with DRAMsim2 [52]. We measure the relative IPC and DRAM energy consumption of 100M instructions (after fast-forwarded 500M-2B instructions) for different ECC organizations.

A. Reliability Evaluation

Our reliability evaluation framework is based on a two-stage Monte Carlo simulation, which is designed to simultaneously consider both inherent and operational faults. The first simulation stage is similar to Roberts et al. [53]; we randomly inject operational faults into a DRAM channel based on published fault rates [20]. We double the DRAM banks and quadruple the rows to accommodate 16GB DIMMs. To better evaluate ECC schemes against operational faults, we use detailed fault modes broken down by the number of failing DRAM I/O pins. Once faults are identified by the first stage, they are mapped onto bit/pin/word/chip faults. A bit fault has a single bit-error within the block. A pin/word/chip fault constrains errors to a single data pin/single chip and single beat/single chip, respectively. These models represent faults at different DRAM structures (e.g., cells, mats, or sense-amplifiers), and match the fault modes in recent DRAM field studies [20, 27, 54].

The second stage of our Monte Carlo simulation generates random errors based on the underlying fault models, assuming that each bit within a corrupted region has a 50% flipping probability (but error-free pattern is excluded). When a fault is injected, we first check whether the fault can be overlapped with older injected faults or not, based on addresses and coverage of fault modes. The generated error is then passed through an ECC decoder to determine whether the errors are correctable, detectable, or miscorrected. Undetected errors and miscorrected errors are categorized as SDC, whereas uncorrectable (detectable) errors are categorized as DUE.

To show the robustness of DUO, we use two error models for inherent faults. The first error model is based on a simple assumption that the inherent faults are permanent; prior work evaluates reliability based on this error model. To increase fidelity, our model also has a simple manufacturing yield component that introduces some non-linearity by sparing or scrapping highly faulty DRAM chips before they are deployed. As the second error model, we employ an error model based on VRT modeling [23]. The main benefit of this model is that intermittent inherent faults can be modeled in addition to permanent inherent faults. Inherent faults are parameterized by an error activation probability, a—permanent inherent faults can be modeled with $a_p \approx 1$ since they almost always generate errors, while intermittent inherent faults may have $a_i \ll 1$. We set the rate of permanent inherent faults p to be higher than or equal to 10^{-6}. We also assume periodic memory scrubbing with a 83-hour interval to prevent the accumulation of transient and inherent errors.

Operational fault rates [20]		
mode	transient	permanent
single bit	14.2	18.6
single word	1.4	0.3
single column	1.4	5.6
single row	0.2	8.2
single bank	0.8	10.0
multiple bank	0.3	1.4
multiple rank	0.9	2.8

Inherent Faults Parameters [23]		
	intermittent	permanent
rate	10^{-5}	$10^{-6} - 10^{-4}$
activation prob.	$10^{-7} - 10^{-5}$	1

Tab. V: Summary of parameters, fault rates, and target ECC schemes. Operational faults are in FIT / device.

Processor		
Number of cores	4	
Clock speed	3.4GHz	
Issue width	8	
Number of ROB entries	192	
Cache Hierarchy		
L1D	32KB/ 8-way / 4 cycles	
L2	1MB / 8-way / 12 cycles	
L3	8MB / 16-way / 36 cycles	
DRAM Memory $\times 4$ 16GB DIMM (DDR4-2400)		
Timing (cycles)	$tCL - tCCD_L(WR) - tFAW - tBL$	
w/o IECC	16 – 6 – 16 – 4	
w/ IECC	18 – 24 – 36 – 4	
DUO SDDC	16 – 6 – 16 – 5	
ECC latency (cycles)	RD FIFO	WR FIFO
SDDC [14]	+3	+2
DUO SDDC	+8	+2
Current (mA)	IDD0 - IDD3N - IDD4R - IDD4W - IDD5	
w/o IECC	43 - 38 - 110 - 103 - 250	
w/ IECC	53 - 54 - 191 - 274 - 251	
DUO SDDC	43.2 - 38.3 - 111.6 - 104.2 - 251.1	

Tab. VI: Simulation parameters for the performance and power evaluation. The parameters are based on DDR4-2400, the I/O bus frequency of which is $1.2GHz$.

B. Performance Evaluation

To quantify performance, we use gem5 [51] integrated with DRAMSim2 [52] and run homogeneous multi-programmed SPEC CPU 2006 workload mixes with one program per core [55]. The parameters for the 4 CPU cores are based on the Intel E7-4470 (Haswell), and DRAM timing and power parameters are based on Micron DDR4-2400 memory [56].

Due to a lack of publicly available information regarding changes in DRAM timing parameters due to IECC, we estimate these changes based on the DDR4 specification [15] and the expected internal operations to the best of our knowledge.

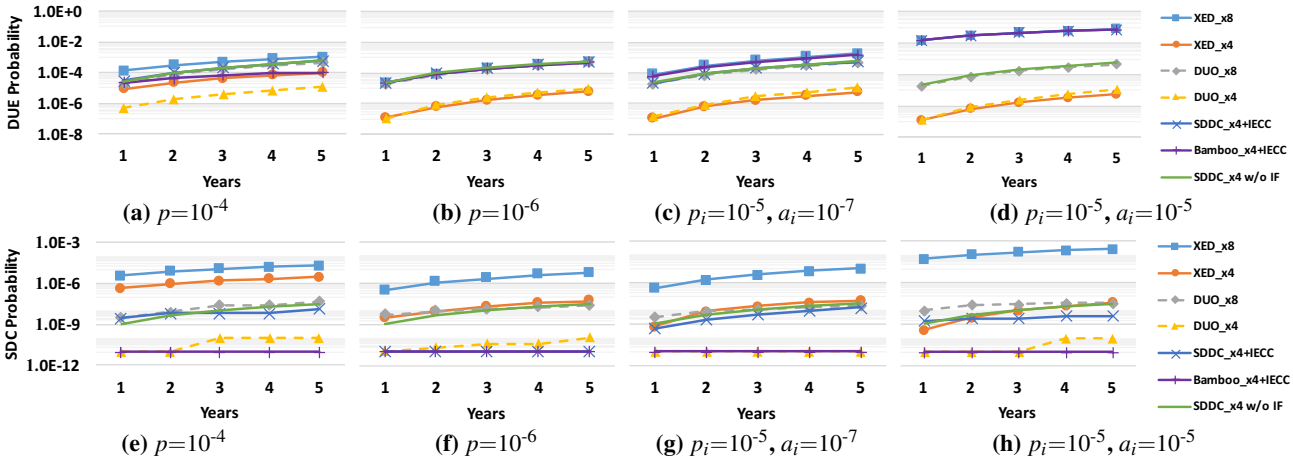


Fig. 6: The DUE and SDC probabilities of the different ECC schemes over time. We sweep the permanent fault rate (p), the intermittent fault rate (p_i) and the inherent fault activation probability (a_i). p=10^{-6} for (c), (d), (g), and (h).

For example, a longer column-to-column delay ($tCCD_L$) is expected for write commands within the same bank group. One more cycle is added to the burst length duration (tBL) of DUO. To simulate ECC performance overhead, we delay memory responses with RD/WR FIFO queues. We assume that both encoding/decoding of IECC and encoding/syndrome generation of rank-level ECC take 2 cycles [3]. We configure the decoding latency of RECC (RS codes) based on the number of correctable symbols as summarized in Tab. IV.

DRAM operations consume more power with in-DRAM ECC due to the 6.25% redundant cells, overfetching, and read-modify-writes. Similar to the DRAM timing parameters, we project the current parameters based on the differences between $\times$4 and $\times$8 devices. For example, while $\times$4 devices require an IDD0 of 43mA to activate a 4Kb row buffer, a $\times$8 device requires 46mA for activating a 8Kb row buffer.[6] That is, activating 4Kb more cells requires only 3mA-higher current, when assuming the same background power consumption level. Since IECC needs to activate bigger row buffers (of 17Kb) because of on-chip redundancy and overfetch, We estimate that a $\times$4 device with IECC will require about 53mA for IDD0. To measure DRAM energy, we use the Micron power model [57]. The parameters used for our performance and power evaluation are summarized in Tab. VI.

VII. Evaluation Results

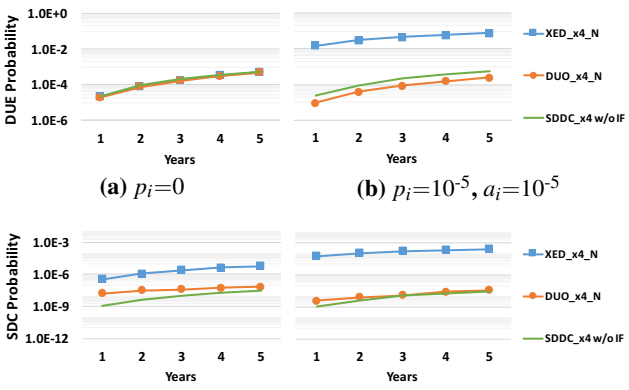
A. Reliability

The correction capability of DUO SDDC is on par with or better than the state of the art ECC schemes in the presence of both operational and inherent faults. Figs. 6a and 6b show the DUE probabilities of the ECC schemes over time, varying the rate of permanent inherent faults p from 10^{-4} to 10^{-6}. The DUE probabilities of XED increase more rapidly than with other schemes as p increases as multi-bit errors occurs more frequently. This is because XED relies on weak IECC correction against the multi-device multi-bit error cases. Unlike XED, DUO SDDC maintains high correction capability by correcting multi-device errors with both its primary and secondary correction mechanisms: primary correction handles the co-located multiple inherent faults (with no coarse-grained operational faults), and secondary correction handles the coarse-grained faults co-located with up to two inherent faults.

The detection capability of DUO SDDC is appropriate for highly-reliable systems because of its low SDC probability. Figs. 6e and 6f show the SDC probabilities of the ECC schemes over time. SDDC +IECC has weak correction capabilities, but it maintains strong detection and provides lower SDC probabilities than XED. Due to its longer codeword, Bamboo shows much stronger detection capability than normal SDDC, whereas its correction capability is not significantly better than SDDC. While XED suffers from an increasing SDCs as p grows, DUO SDDC maintains a superior level of error detection.

DUO SDDC is robust against intermittent inherent faults. Figs. 6c and 6d show the DUE probabilities of the ECC schemes when considering the intermittent inherent faults in addition to permanent inherent faults. Similarly, Figs. 6g and 6h show the SDC probabilities of the ECC schemes. We sweep the activation probability (a_i) of intermittent inherent faults while maintaining a modest impact from permanent inherent faults by setting $p = 10^{-6}$. Here, we evaluate the ECC schemes in harsher conditions and observe more frequent intermittent inherent faults. While DUO SDDC does not show dramatic reliability degradation, the other schemes show significantly increased DUE and SDC probabilities as the impact of intermittent inherent faults increases (by increasing a_i). This is mainly because multiple devices simultaneously suffer from multi-bit errors as more intermittent inherent faults are injected and generating errors. While the primary

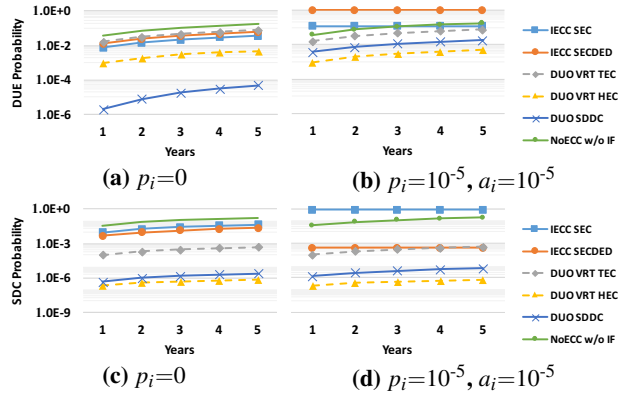
[6]IDD0, IDD3N, IDD4R/IDD4W, and IDD5 are the DRAM activation current, active standby current, operating burst read/write current, and burst refresh current, respectively. (See [56] for details.)



(a) p_i=0 (b) p_i=10^{-5}, a_i=10^{-5}

(c) p_i=0 (d) p_i=10^{-5}, a_i=10^{-5}

Fig. 7: DDR5* ×4 DRAM. We sweep the intermittent fault rate (p_i) and the inherent fault activation probability (a_i). The permanent fault rate (p) is held constant at p=10^{-6}.



(a) p_i=0 (b) p_i=10^{-5}, a_i=10^{-5}

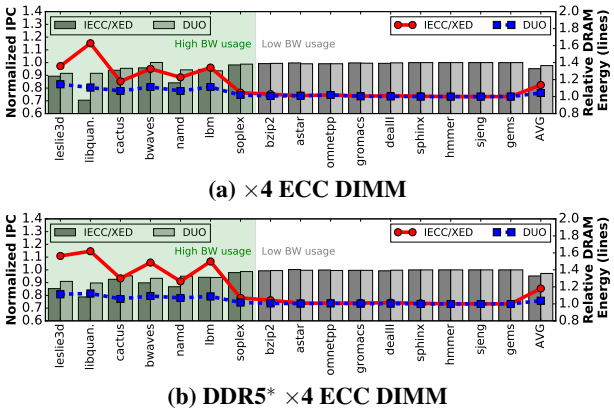
(c) p_i=0 (d) p_i=10^{-5}, a_i=10^{-5}

Fig. 8: Non-ECC DIMM (same parameter setup as Fig. 7).

correction of DUO SDDC is very strong against the multi-bit multi-device errors from the increased inherent faults—it can correct up to 6 error symbols regardless of error location—the other ECC schemes rely on weak correction capability when multiple devices have severe errors.

DUO offers high reliability for future DRAM and non-ECC DIMMs. Fig. 7 shows the DUE and SDC probabilities of DUO SDDC and XED for DDR5*. The permanent inherent fault rate is set to $p = 10^{-6}$, similar to Fig. 6. With only permanent inherent faults, DUO SDDC's DUE and SDC probabilities are on par with SDDC without any inherent faults (Figs. 7a and 7c). Even in a harsh condition with intermittent inherent faults ($p_i = 10^{-5}$ and $a_i = 10^{-5}$), DUO SDDC's DUE and SDC probabilities are only slightly increased, and are still on par with the SDDC-level baseline (Figs. 7b and 7d). XED for DDR5* experiences significant reliability degradation with respect to both DUE and SDC.

Fig. 8 shows the reliability evaluation of non-ECC DIMMs. Due to coarse-grained operational faults, all the ECC schemes show relatively low reliability. Compared to IECC, however, DUO ECC substantially lowers both DUE and SDC probabilities: DUO VRT for 12.5% redundancy (HEC) shows impressively low SDC probability as it is extremely unlikely to



(a) ×4 ECC DIMM

(b) DDR5* ×4 ECC DIMM

Fig. 9: Performance & DRAM energy comparison. BW use higher than 1GB/s is "high" (shaded).

miscorrect, even with device failures. Although DUO SDDC shows much lower DUE probability with lower inherent fault rates, because it can tolerate a failed device (Fig. 8a), its aggressive correction results in more miscorrections than DUO VRT (Figs. 8c and 8d). Note that with a high rate of intermittent faults, IECC has a 100% failure probability: IECC SEC almost certainly exhibits an SDC, which is why its DUE rate appears lower than that of IECC SECDED.

B. Performance and Energy

DUO degrades system performance little for most applications and it outperforms IECC. Fig. 9 compares the performance between DUO and IECC-based schemes (we expect that XED shows roughly the same performance as IECC). Although both schemes show on average 2–5% performance degradation for all three configurations, variations exist in the performance degradation among benchmark applications. While detailed information is omitted here, we sorted and divided applications into high and low BW usage to show intense memory usage degrades performance. For IECC, the overheads discussed in Section IV-A (read-modify-write and increased CAS latency from IECC decoding) degrade performance of memory-intensive applications by up to 30%. On the other hand, DUO shows less performance degradation than IECC (up to about 10% for memory-intensive applications) despite longer bursts higher decoding latency at the rank-level, because the overheads of IECC are avoided.

DUO can save DRAM energy consumption relative to IECC. Fig. 9 also compares the DRAM energy consumption of DUO to that IECC-based schemes. DUO offers significantly better DRAM energy efficiency. For memory-intensive applications, IECC spends 13–20% more DRAM energy on average (and up to 63% with libquantum at ×4 ECC DIMM) than ECC DIMM without IECC, due to overfetching and read-modify-write operations. On the other hand, DUO shows a more modest increase of 4–14% on average, akin to the normal expectations for an ECC-protected system.

VIII. Conclusions

We present an alternative use of on-chip redundancy (DUO) and a novel ECC design that utilize the benefits of DUO in order to achieve high reliability. We conclude that IECC, a current solution for DRAM scaling, not only uses on-chip redundancy inefficiently but also degrades performance and DRAM energy efficiency because of a granularity mismatch between codeword and access data. In contrast, DUO avoids the mismatch issue and achieves a slightly better performance and significant DRAM energy saving for memory-intensive applications. Moreover, DUO can efficiently handle the increasing inherent faults due to continued scaling, and its robustness is proved by considering two different inherent fault-error models and narrow channel future DRAM.

Acknowledgements

The authors acknowledge the support of Huawei Technologies and Texas Advanced Computing Center for providing HPC resources.

References

[1] M. Horiguchi, "Redundancy techniques for high-density drams," in *1997 Proceedings Second Annual IEEE International Conference on Innovative Systems in Silicon*, pp. 22–29, Oct 1997.

[2] C.-L. Su, Y.-T. Yeh, and C.-W. Wu, "An integrated ECC and redundancy repair scheme for memory reliability enhancement," in *Proceedings of DFT*, 2005.

[3] S. Cha, S. O, H. Shin, S. Hwang, K. Park, S. J. Jang, J. S. Choi, G. Y. Jin, Y. H. Son, H. C. J. H. Ahn, and N. S. Kim, "Defect analysis and costeffective resiliency architecture for future dram devices," in *Proceedings of HPCA*, 2017.

[4] U. Kang, H. soo Yu, C. Park, H. Zheng, J. Halbert, K. Bains, S. Jang, and J. S. Choi, "Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling," in *The Memory Forum*, 2014.

[5] P. J. Nair, D.-H. Kim, and M. K. Qureshi, "Archshield: Architectural framework for assisting DRAM scaling by tolerating high error rates," in *Proceedings of ISCA*, 2013.

[6] H. L. Kalter, C. H. Stapper, J. E. Barth, J. DiLorenzo, C. E. Drake, J. A. Fifield, G. A. Kelley, S. C. Lewis, W. B. v. d. Hoeven, and J. A. Yankosky, "A 50-ns 16-Mb DRAM with a 10-ns data rate and on-chip ECC," *IEEE Journal of Solid State Circuits*, vol. 25, no. 5, pp. 1118–1128, 1990.

[7] K. Arimoto, Y. Matsuda, K. Furutani, M. Tsukude, T. Ooishi, K. Mashiko, and K. Fujishima, "A speed-enhanced DRAM array architecture with embedded ECC," *IEEE Journal of Solid State Circuits*, vol. 25, no. 1, pp. 11–17, 1990.

[8] S.-H. Kim, W.-O. Lee, J.-H. Kim, S.-S. Lee, S.-Y. Hwang, C.-I. Kim, T.-W. Kwon, B.-S. Han, S.-K. Cho, D.-H. Kim, J.-K. Hong, M.-Y. Lee, S.-W. Yin, H.-G. Kim, J.-H. Ahn, Y.-T. Kim, Y.-H. Koh, and J.-S. Kih, "A low power and highly reliable 400mbps mobile DDR SDRAM with on-chip distributed ECC," in *Proceedings of the Asian Solid-State Circuits Conference*, pp. 34–37, 2007.

[9] S. Kwon, Y. H. Son, and J. H. Ahn, "Understanding DDR4 in pursuit of In-DRAM ECC," in *Proceedings of ISOCC*, pp. 276–277, Nov 2014.

[10] Joint Electron Device Engineering Council, *Low Power Double Data Rate 4 (LPDDR4), JESD209-4A*, Nov. 2015.

[11] T. Y. Oh, H. Chung, J. Y. Park, K. W. Lee, S. Oh, S. Y. Doo, H. J. Kim, C. Lee, H. R. Kim, J. H. Lee, J. I. Lee, K. S. Ha, Y. Choi, Y. C. Cho, Y. C. Bae, T. Jang, C. Park, K. Park, S. Jang, and J. S. Choi, "A 3.2 Gbps/pin 8 Gbit 1.0 V LPDDR4 SDRAM with integrated ECC engine for sub-1 V DRAM core operation," *IEEE Journal of Solid State Circuits*, vol. 50, no. 1, pp. 178–190, 2015.

[12] Y. H. Son, S. Lee, O. Seongil, S. Kwon, N. S. Kim, and J. H. Ahn, "CiDRA: A cache-inspired DRAM resilience architecture," in *Proceedings of HPCA*, 2015.

[13] P. J. Nair, V. Sridharan, and M. K. Qureshi, "XED: Exposing On-Die Error Detection Information for StrongMemoryReliability," in *Proceedings of ISCA*, 2016.

[14] Advanced Micro Devices (AMD), Inc., "BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 00h-0Fh Processors," Jan 2013.

[15] Joint Electron Device Engineering Council, *DDR4 SDRAM STANDARD, JESD79-4*, Sep. 2012.

[16] K. Kim and J. Lee, "A new investigation of data retention time in truly nanoscaled drams," *IEEE Electron Device Letters*, vol. 30, pp. 846–848, Aug 2009.

[17] M. K. Qureshi, D. H. Kim, S. Khan, P. J. Nair, and O. Mutlu, "AVATAR: A variable-retention-time (VRT) aware refresh for DRAM systems," in *Proceedings of DSN*, pp. 427–437, 2015.

[18] S. Khan, D. Lee, Y. Kim, A. R. Alameldeen, C. Wilkerson, and O. Mutlu, "The efficacy of error mitigation techniques for DRAM retention failures: A comparative experimental study," in *Proceedings of SIGMETRICS*, 2014.

[19] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Trans. on Dependable and Secure Computing*, vol. 1, pp. 11–33, Jan. 2004.

[20] V. Sridharan and D. Liberty, "A Study of DRAM Failures in the Field," in *Proceedings of SC*, 2012.

[21] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," in *Proceedings of ISCA*, 2014.

[22] A. Bacchini, M. Rovatti, G. Furano, and M. Ottavi, "Characterization of data retention faults in DRAM devices," in *Proceedings of DFT*, pp. 9–14, 2014.

[23] S.-L. Gong, J. Kim, and M. Erez, "DRAM Scaling Error Evaluation Model Using Various Retention Time," in *Workshop on SELSE*, March 2017.

[24] Y. C. Yu, C. S. Hou, L. J. Chang, J. F. Li, C. Y. Lo, D. M. Kwai, Y. F. Chou, and C. W. Wu, "A hybrid ECC and redundancy technique for reducing refresh power of DRAMs," in *Proceedings of the VLSI Test Symposium (VTS)*, pp. 1–6, 2013.

[25] B. Schroeder, E. Pinheiro, and W.-D. Weber, "DRAM Errors in the Wild: a Large-Scale Field Study," in *Proceedings of SIGMETRICS*, 2009.

[26] A. A. Hwang, I. A. Stefanovici, and B. Schroeder, "Cosmic Rays Don't Strike Twice: Understanding the Nature of DRAM Errors and the Implications for System Design," in *Proceedings of ASPLOS*, 2012.

[27] V. Sridharan, N. DeBardeleben, S. Blanchard, K. B. Ferreira, J. Stearley, J. Shalf, and S. Gurumurthi, "Memory Errors in Modern Systems: The Good, The Bad, and The Ugly," in *Proceedings of ASPLOS*, 2015.

[28] J. Meza, Q. Wu, S. Kumar, and O. Mutlu, "Revisiting memory errors in large-scale production data centers: Analysis and modeling of new trends from the field," in *DSN*, (Washington, DC, USA), pp. 415–426, 2015.

[29] J. Shalf, S. Dosanjh, and J. Morrison, "Exascale Computing Technology Challenges," in *Proceedings of VECPAR*, pp. 1–25, 2011.

[30] IBM, "Chipkill Memory," tech. rep., 2012.

[31] Hewlett-Packard, "HP Advanced Memory Error Detection Technology," 2011.

[32] Intel corp., "Intel Xeon Processor E7 Family: Reliability, Availability, and Serviceability," 2011.

[33] Oracle, Inc., "Oracle SPARC Server RAS Comparison,"

[34] Hewlett-Packard, "How memory RAS technologies can enhance the uptime of HP ProLiant servers," 2013.

[35] Sun Microsystems, Inc., "T2 core microarchitecture specification,"

[36] C. Di Martino, Z. Kalbarczyk, R. Iyer, F. Baccanico, J. Fullop, and W. Kramer, "Lessons learned from the analysis of system failures at petascale: The case of Blue Waters," in *Proceedings of DSN*, 2014.

[37] J. Kim, M. Sullivan, and M. Erez, "Bamboo ECC: Strong, Safe, and Flexible Codes for Reliable Computer Memory," in *Proceedings of HPCA*, 2015.

[38] A. M. Saleh, J. J. Serrano, and J. H. Patel, "Reliability of scrubbing recovery-techniques for memory systems," *IEEE Transactions on Reliability*, vol. 39, pp. 114–122, Apr 1990.

[39] Joint Electron Device Engineering Council, *DDR2 SDRAM Specification, JESD79-2F*, Nov. 2009.

[40] Joint Electron Device Engineering Council, *DDR3 SDRAM STANDARD, JESD79-3F*, July 2012.

[41] D. H. Yoon, M. K. Jeong, M. Sullivan, and M. Erez, "The Dynamic Granularity Memory System," in *Proceedings of ISCA*, 2012.

[42] M. Rhu, M. Sullivan, J. Leng, and M. Erez, "A locality-aware memory hierarchy for energy-efficient gpu architectures," in *Proceedings of MICRO*, 2013.

[43] S.-L. Gong, M. Rhu, J. Kim, J. Chung, and M. Erez, "Clean-ecc: High reliability ecc for adaptive granularity memory system," in *Proceedings of MICRO*, 2015.

[44] R. Bose and D. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, no. 1, pp. 68 – 79, 1960.

[45] Samsung Electronics Co., "8Gb B-die DDR4 SDRAM." http://www.samsung.com/semiconductor/global/file/product/2017/02/8G_B_DDR4_Samsung_Spec_Rev2_1_Feb_17-0.pdf.

[46] SK hynix Inc., "8Gb DDR4 SDRAM." https://www.skhynix.com/product/filedata/fileDownload.do?seq=7128.

[47] Micron Technology Co., "8Gb DDR4 SDRAM." https://www.micron.com/~/media/documents/products/data-sheet/dram/ddr4/8gb_auto_ddr4_dram.pdf.

[48] J.-H. Jeng and T.-K. Truong, "On decoding of both errors and erasures of a reed-solomon code using an inverse-free berlekamp-massey algorithm," *IEEE Transactions on Communications*, vol. 47, pp. 1488–1494, Oct 1999.

[49] X. Jian, H. Duwe, J. Sartori, V. Sridharan, and R. Kumar, "Low-power, low-storage-overhead chipkill correct via multi-line error correction.," in *Proceedings of SC*, 2013.

[50] J. Massey, "Shift-register synthesis and bch decoding," *IEEE Transactions on Information Theory*, vol. 15, pp. 122–127, Jan 1969.

[51] "The gem5 Simulator System: A Modular Platform for Computer System Architecture Research." http://www.gem5.org/.

[52] P. Rosenfeld, E. Cooper-Balis, and B. Jacob, "DRAMSim2: A Cycle Accurate Memory System Simulator," *IEEE Comp. Arch. Lett.*, vol. 10, pp. 16–19, Jan. 2011.
[53] D. Roberts and P. Nair, "FAULTSIM: A fast, configurable memory-resilience simulator," in *The Memory Forum*, 2014.
[54] V. Sridharan, J. Stearley, N. DeBardeleben, S. Blanchard, and S. Gurumurthi, "Feng Shui of Supercomputer Memory: Positional Effects in DRAM and SRAM Faults," in *Proceedings of SC*, 2013.
[55] Standard Performance Evaluation Corporation, "SPEC CPU 2006," 2006.
[56] Micron Technology Co., "8Gb DDR4 SDRAM datasheet." url=https:/www.micron.com/~/media/documents/products/data-sheet/dram/ddr4/8gb_ddr4_sdram.pdf, 2015.
[57] Micron Technology Co., "Calculating Memory System Power for DDR3." http://www.micron.com/-/media/Documents/Products/TechnicalNote/DRAM/TN41_01DDR3_Power.pdf, 2007.