On the Performance of LDPC and Turbo Decoder Architectures with Unreliable Memories

Joao Andrade*, Aida Vosoughi†, Guohui Wang†, Georgios Karakonstantis‡,
Andreas Burg‡, Gabriel Falcao*, Vitor Silva*, Joseph R. Cavallaro†
* Instituto de Telecomunicações, Univ. of Coimbra, Portugal; † Department of ECE, Rice Univ., Houston, TX, USA
‡ Telecommunications Circuits Lab, EPFL, Lausanne, Switzerland; Email: jandrade@co.it.pt, vosoughi@rice.edu

***Abstract*—In this paper, we investigate the impact of faulty memory bit-cells on the performance of LDPC and Turbo channel decoders based on realistic memory failure models. Our study investigates the inherent error resilience of such codes to potential memory faults affecting the decoding process. We develop two mitigation mechanisms that reduce the impact of memory faults rather than correcting every single error. We show how protection of only few bit-cells is sufficient to deal with high defect rates. In addition, we show how the use of repair-iterations specifically helps mitigating the impact of faults that occur inside the decoder itself.**

I. Introduction

Aggressive technology scaling has allowed the realization of decoders for powerful error–correcting codes (ECCs) for wireless communication systems, but unfortunately it has also led to extensive process variations in integrated circuits which may result in a variety of failures [1]. Specifically, embedded memories in sub-45nm are particularly sensitive to variations and may often fail to correctly retain the stored data. Such failures are further increasing when voltages are scaled and when classical static memories (SRAMs) are substituted by embedded memory (eDRAM) for improving the energy efficiency of communication systems [2], [3]. Traditional techniques such as using larger memory bit-cells or adding redundancy to stored data through circuit-level ECCs may help to ensure 100% reliable storage, however such measures incur large area and power overheads. It is becoming apparent that as we move beyond the 22nm process technology where variations are expected to be more severe such conventional schemes will not be sustainable due to the large amount of incurred overhead required to address every single fault.

Such a reality has led to the exploration of alternative paradigms – such as approximate computing – in which the strict requirements for 100% reliable operation are relaxed in an attempt to limit the design overhead of classical fault mitigation methods [4]. Recent works under this paradigm have shown that it is indeed possible to limit the overheads and allow low cost unreliable memories by utilizing the system level redundancy of communication systems [5]. Studies of the impact of arithmetic errors in Low–Density Parity–Check (LDPC) decoders [6]–[8], and memory faults in Turbo decoders [9], [10] have indicated the inherent error resilience of such essential blocks and resulted even in the design of new Viterbi decoders under such errors [11]. However, there is still a need to further study and compare popular iterative decoders and explore alternative fault mitigation schemes.

In this paper, we analyze the behavior of two iterative codes and their respective decoders under unreliable memories and propose two mitigation strategies based on the selective protection of certain bits and execution of few redundant iterations for limiting the impact of errors. Our contributions can be summarized as:

- We inject stuck-at-faults into the various memory modules at the input of and within LDPC and Turbo decoders used in Digital Video Broadcasting – Satellite 2 (DVB–S2) [12] and 3GPP LTE [13] systems and analyze the BER performance;
- This analysis provides an estimate of the number of memory cell failures that can be tolerated in each buffer which can then be used for developing suitable memory macros that provide only the required amount of reliability for the benefit of better energy- or area efficiency and for improving yield by avoiding to discard chips with only few failing bit cells;
- We explore unequal error protection, which prevents stuck-at failures in the most significant bit (MSB)-cells;
- We introduce the concept of repair iterations starting after a given number of decoding iterations which can eliminate all the errors in the decoder inner memories.

II. Unreliable Memory Model

In order to determine how many bit-cell failures can be tolerated by the decoder memories without a noticeable loss in BER performance, we consider a system in which information bits are coded, modulated, and sent through an additive white Gaussian noise (AWGN) channel. At the receiver, symbols are demodulated and the corresponding log–likelihood ratios (LLRs) are passed on to the channel decoder. To model the memory reliability issues, we consider a stuck-at fault model which matches the standard failure model for embedded memories [16]. This model assumes that during production

This work was partially supported by:
* The Fundação para a Ciência e Tecnologia (FCT) under grants SFRH/BD/78238/2011 and PEst-OE/EEI/LA0008/2013 through POPH/FSE funding, and Fundação Luso-Americana para o Desenvolvimento (FLAD) under grant 2014/CON15/CAN8.
† The US National Science Foundation under grants ECCS-1408370, CNS-1265332, and ECCS-1232274.
‡ The Swiss NSF grant no. 200021_153640 and the EU Marie-Curie DARE grant no. 304186.

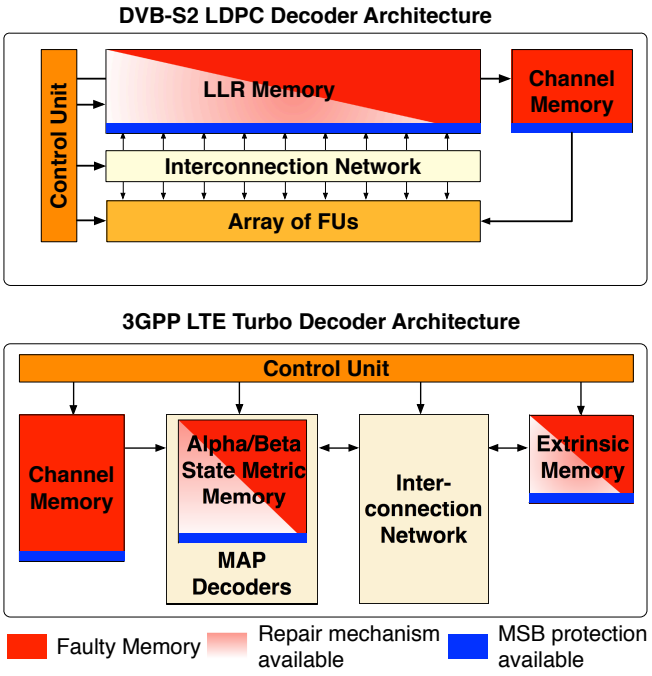


Figure 1: LDPC and Turbo decoder architectures [14], [15].

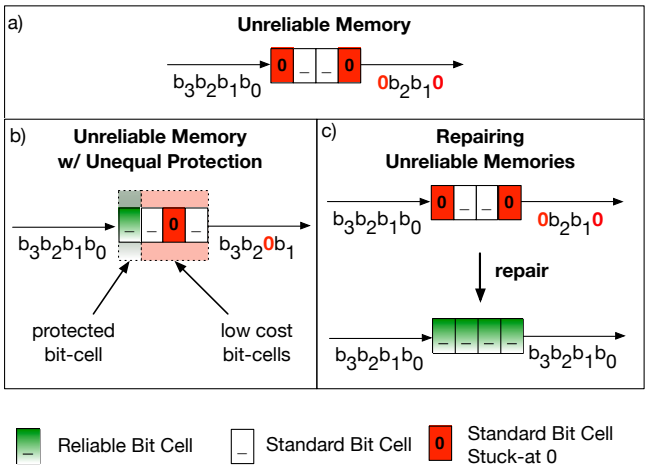


Figure 2: Unreliable memory model and mitigation strategies on 4-bit word: a) stuck-at 0 faults might occur more than once per word; b) the MSB is protected and always retains its correct value (unequal protection); c) repairing the bit cells prevents stuck-at faults from occurring thereon.

each bit-cell independently comes out as a failing or not with a bit-cell failure probability P_s. In post-production testing, all circuits with memories exceeding a given percentage $P_s = P$ of failing bit-cell are discarded (note that today, $P = 0$ which leads to a yield penalty).

The size of a memory block m with depth of W_m words and width of B_m bits is defined as $S_m = W_m \times B_m$. The number of stuck-at bit-cells in a worst-case memory block is given by $E_m = \lfloor S_m \times P_s \rfloor$. In our simulations, we consider only the performance of such worst-case chips. To this end, we first fix P_s and calculate E_m (number of cell failures) for each of the memory blocks in the channel decoder. Then, in each iteration of the Monte Carlo simulation a random block of information bits is generated and for each memory block m, E_m cells are marked as stuck-at based on i.i.d. uniform distribution. The simulation is then repeated for a different bit-cell failure probability P_s. We modeled the bit-cell failures to be of type stuck-at 0 with a probability denoted by P_{s0}.

III. BER Degradation Mitigation Strategies

LDPC and Turbo codes are widely used in wireless communications standards [12], [13], [17]. Therefore, it is desirable to characterize the sensitivity of each type of ECC to errors introduced by bit-cell failures across different memories in the decoding architectures. To this end, we replace the memory blocks shown in Fig. 1 with memories which are unreliable and inject stuck-at 0 faults with probability P_{s0} in a set of different experiments, summarized in Tbl. I.

First, we evaluate the behavior of the decoders under stuck-at 0 faults injected only in the channel memory only, as shown in Fig. 2-a). Secondly, we extend the error injection across all the memories in the decoder (LLR memory in the LDPC case, and extrinsic and state metric memories in the Turbo case). It should be noted that all memories are treated as separate blocks. This means that we compute a fixed number of bit-cell faults uniformly distributed across each memory type and not across the collective sum of all the bit-cells in the memories. Faults injected in the channel memory can be portrayed as distortions in the wireless channel due to noise or interference [10], [16], while this analogy no longer holds for the memories within the decoders.

As BER will degrade due to the errors injected by the bit-cell failures, mitigation strategies can be employed based on different assumptions. The concrete implementation of the proposed mitigation strategies on hardware remains an open issue and lies outside the scope of this paper.

A. Unequal Bit Protection – MSB Protection

Both LDPC and Turbo decoding algorithms are LLR-based [14], [15], whereupon the LLR of bit b, of received baseband sample y, $L(b)$ is obtained as follows:

$$L(b) = \log\left(\frac{P(b=0 \mid y)}{P(b=1 \mid y)}\right). \quad (1)$$

Thus, whenever bit b has a higher probability of being 0, its LLR is positive and negative otherwise. This makes the sign the most important information retained in the LLR representation of the likelihood. Naturally, the remaining bits are also important for determining the LLR magnitude, but not as critical as the former.

Hence, we can devise an unequal error protection strategy whereupon the MSB, which directly or indirectly carries the sign information for both sign-magnitude and 2's-complement data representation, is protected, as shown in Fig. 2-b). This

Table I: Experiments for the LDPC and Turbo decoders with unreliable memory and BER mitigation strategies.

Exp.	Unreliable Memories	MSB protection?	Repair Iterations?	Fig.
I	None	N/A	N/A	3, 4
II	Channel	No	No	3a), 3c)
III	Channel	Yes	No	3a), 3c)
IV	All	No	No	3b), 3d)
V	All	Yes	No	3b), 3d)
VI	All	No	Yes*	4

All memories except channel repaired from X up to MAX_{iter}
* w/ $X \in \{0, 9, 14, 19, 24, 29\}$ and $X \in \{0, 1, 2, \cdots, 8\}$ respectively for LDPC and Turbo.

means that the bit-cells storing signs are no longer subject to bit-cell failures. Thus, we rename P_{s0} to $P_{s0,MSB}$to specify the percentage of failing bit-cells among only those cells that are actually subject to errors.

B. Follow-up Repair Iterations

An alternative protection strategy dedicated specifically for limiting the impact of BER loss in case of errors is to introduce fault-free repair iterations toward the end of the decoding process. The rationale behind this strategy is based on the insight that errors in the decoder are harder to mitigate when the decoding process itself is faulty and that few fault free iterations could remove those "residual" errors. Such repair-iterations can be realized for example by raising the supply voltage of the decoder memories for the last iterations. Since errors typically only manifest at low voltages, those errors can temporarily disappear (at the expense of power overhead). Overall, the decoders execute a maximum number of iterations denoted by MAX_{iter} which can be distinguished into two types:

a) the first X iterations ($0 \leq X \leq MAX_{iter}$) where errors are present in all the decoder memories;
b) the error-free repair iterations that follow the initial ones, starting at the Xth iteration

A representation of this procedure is shown in Fig. 2-c). Note that when $X=0$, this is equivalent to injecting faults only into the channel memory (same as experiment II) because all the memories, except for the channel memory, are repaired starting from the first iteration. When $X=MAX_{iter}$ it means that faults are introduced in all memories and there is no repair iterations (same as experiment IV).

For this strategy, the lower and upper bounds of the BER performance of the decoders with unreliable memories are easy to see. If the repair iterations are performed for $X=0$ only the channel memory is subject to faults. If no repair iterations are run at all ($X=MAX_{iter}$), this corresponds to the case where bit-cells fail in both channel and LLR memories. The BER loss lies between these two limits.

Tbl. I summarizes the different categories of experiments performed in this work. We investigate both mitigation strategies explained above.

Table II: LDPC and Turbo decoder and BER simulation parameters, memory block designation, type and dimensions.

	N Rate	Mod.	Decod. Algor.	Memory	W_m	B_m	S_m
LDPC	64800 1/3	QPSK	MSA	Channel	64K	8	518K
				LLR	216K	8	1.7M
Turbo	6144 1/3	BPSK	MAP	Channel	18K	6	110K
				Extrinsic	8K	7	42K
				State Metric	25K	10	250K

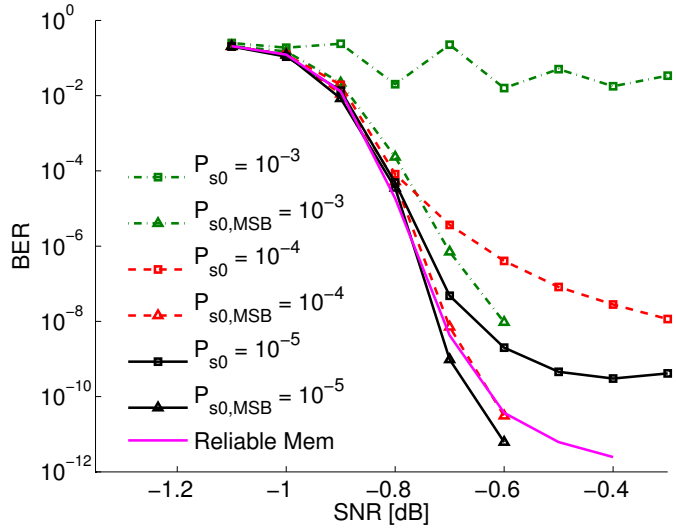
IV. Experimental Results

To evaluate the performance of the LDPC and Turbo when unreliable memories are introduced in the decoders we perform Monte Carlo simulations, summarized in Tbl. I, for the DVB–S2 LDPC and 3GPP LTE/LTE-Advanced Turbo rate $1/3$ codes, with block length N as defined in Tbl. II. The bit-cell failures follow a uniform distribution for a fixed number of failing cells, computed as $E_m = \lfloor S_m \times P_{s0} \rfloor$ for each memory block shown in Tbl. II, with $P_{s0} \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. Our simulations rely on a genie-aided early termination method, thus decoding only stops early whenever the decoded word matches the transmitted codeword or when reaching the maximum number of iterations. The BER simulation performance of the decoders is shown in Fig. 3 for no-protection and for MSB protection and in Fig. 4 for decoding with repair iterations.

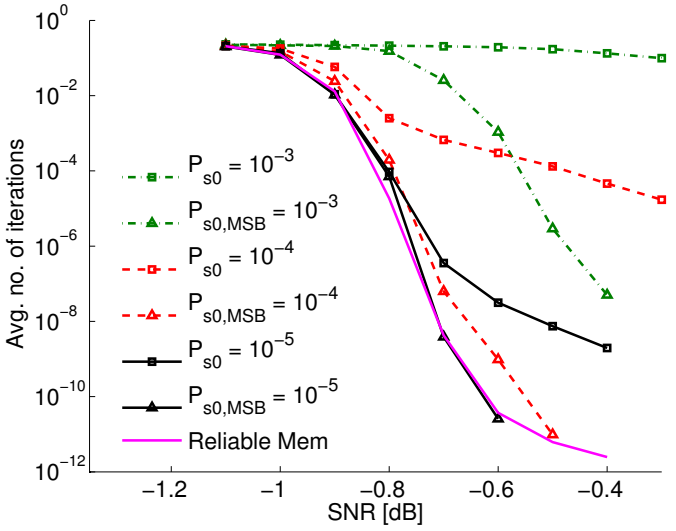
The maximum number of decoding iterations executed is 50 and 8, and the follow-up repair iterations experiment is run for $X \in \{0, 9, 14, 19, 24, 29\}$ and $X \in \{0, 1, \cdots, 8\}$, for LDPC and Turbo decoders respectively. Note that, in repair iterations, all the memories are read without errors except for the channel memory, which is not re-written in the decoding process and thus retains any faults introduced when loading a new codeword. Furthermore, in Tbl. III we analyse the number of decoding iterations executed in all scenarios for a data point in the error-floor region. We analyze this in terms of the absolute value and additional number of iterations required for unreliable memories (experiments II, III, IV, V, and VI) compared to the fully reliable case (experiment I).

A. LDPC Decoder

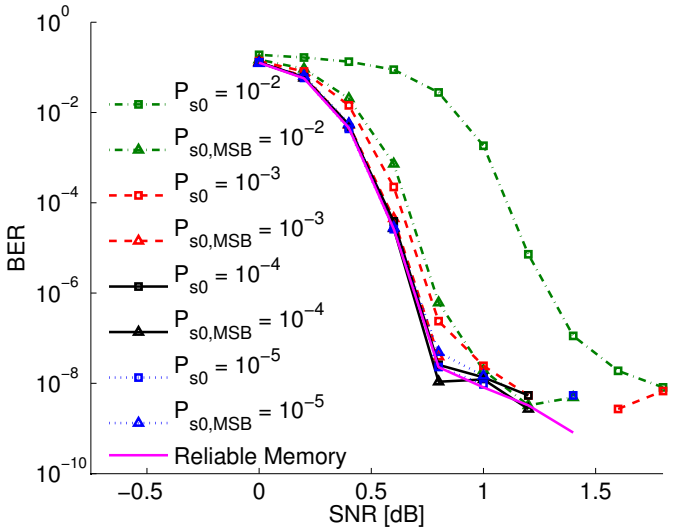
Fig. 3a) shows the simulation result in the case that bit faults are injected only into the channel memory while the internal memories of the decoder operate reliably. It is clearly observed that memory faults in the channel memory raise the error-floor considerably. For $P_{s0}=10^{-3}$ decoding fails across the entire SNR range with a high error floor. However, introduction of MSB protection allows the decoder to recover from this floor. Fig. 3b) depicts the BER obtained when stuck-at faults are present in both channel and inner memories of the LDPC decoder. In this case, we also observe a rapid increase in the error floor, even for a very low percentage of failing memory cells (10^{-5}). Similar to the channel memory, the introduction of MSB protection helps mitigate this effect, with rates of up to 10^{-4} not showing any degradation in the high SNR regime.



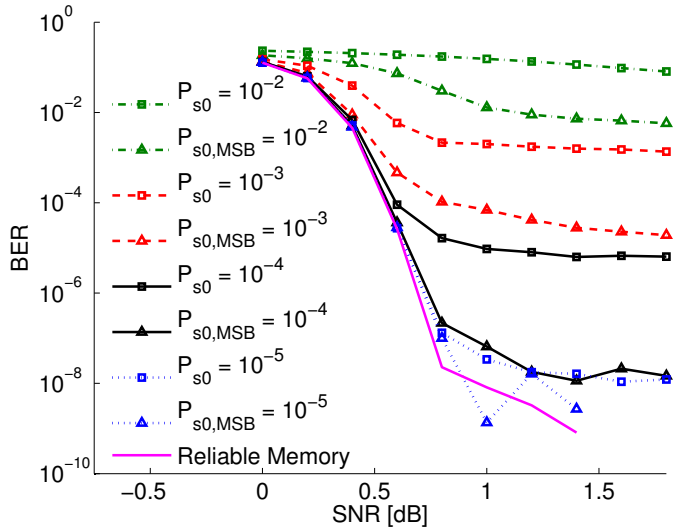
(a) LDPC experiments I, II and III.



(b) LDPC experiments I, IV and V.



(c) Turbo experiments I, II and III.



(d) Turbo experiments I, IV and V.

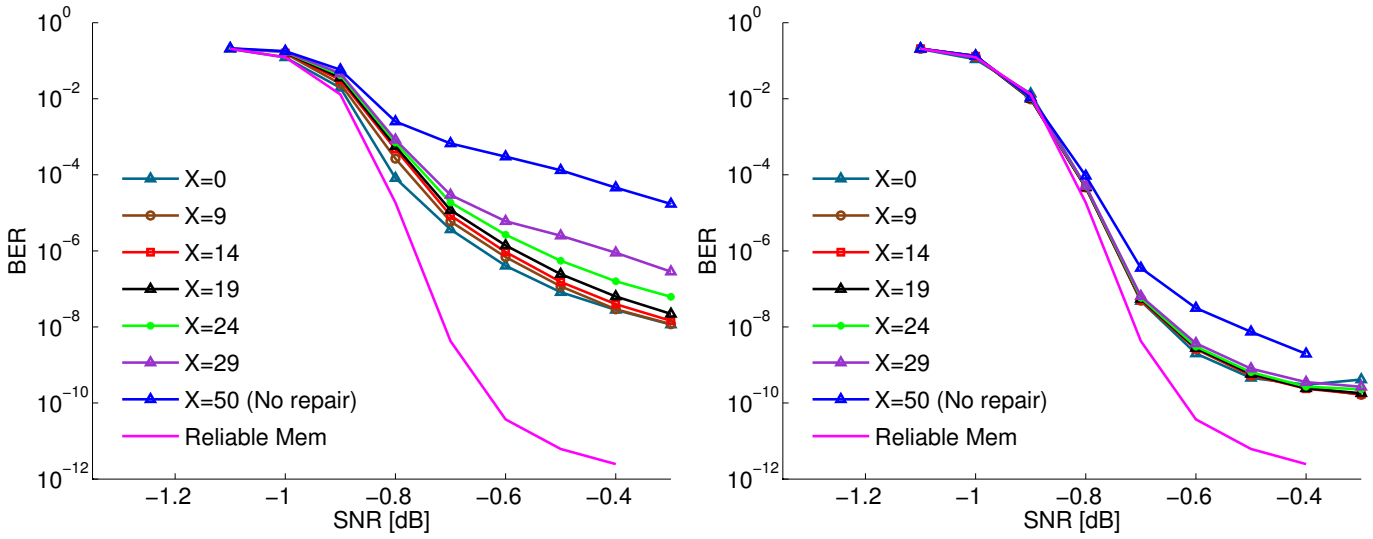
Figure 3: LDPC (a,b) and Turbo (c,d) decoders BER behavior with stuck-at-0 probabilities $P_{s0} \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ for experiments I, II, III, IV and V.

Moreover, we investigated the effect of introducing repair iterations to the decoding procedure. The results are shown in Fig. 4a) and 4b) for $P_{s0} = 10^{-4}$ and $P_{s0} = 10^{-5}$, respectively. We can observe how the BER improves as we reduce the number of iterations in which the inner decoder memories are affected by faults. For a rate of failure of 10^{-4}, a difference in obtained error-floor of 10^{-2} can be attributed to how soon we start repairing. This difference becomes negligible in the 10^{-5} case between how soon we start repairing. As previously said, since we cannot obtain lower BERs than those obtained for X=0, the proposed repair iterations strategy alone does not suffice to obtain a BER similar to the reliable memory case.

The average number of decoding iterations is provided in Tbl. III. We observe that the average number of iterations is closely related to the BER performance degradation due to reliability issues in the memories. For example, we observe that when MSB protection is activated, the extra number of decoding iterations is fairly low. For P_{s0}=10^{-4} and below, at most $\sim$5% extra iterations are required. In the opposite case, when BER degradation is still high, for the repair iterations case, the extra number of iterations required is higher. In this particular case, we can also observe that the sooner the repairing procedure begins – the lower the X – the lower iterations will be the average iteration count.

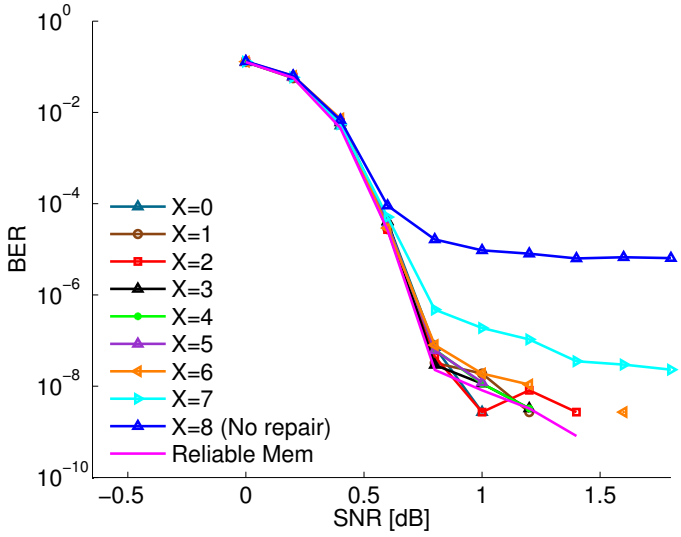
B. Turbo Decoder

Fig. 3c) shows the simulation results in case that bit faults are only injected into the channel input memory while the other memory blocks are assumed to be fault-free. This figure clearly shows a remarkable tolerance against reliability issues

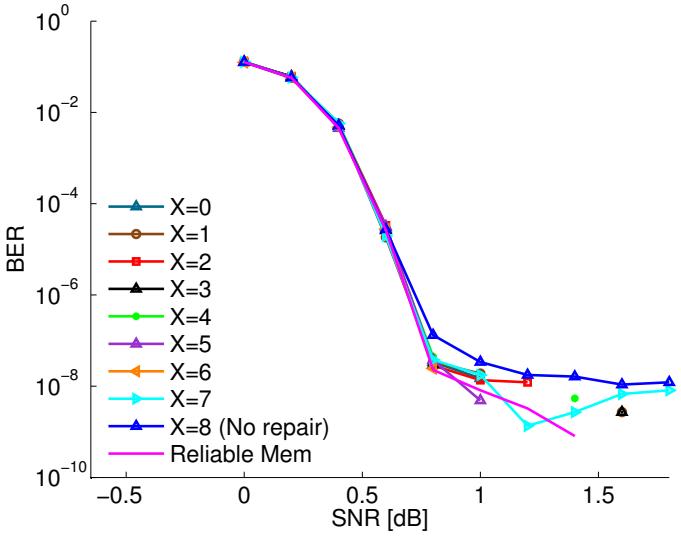


(a) LDPC experiments I and VI with $P_{s0} = 10^{-4}$

(b) LDPC experiments I and VI with $P_{s0} = 10^{-5}$



(c) Turbo experiments I and VI with $P_{s0} = 10^{-4}$



(d) Turbo experiments I and VI with $P_{s0} = 10^{-5}$

Figure 4: LDPC (a,b) and Turbo (c,d) decoders BER behavior for experiments I and VI, starting the repair at iteration X. In a,b) $X \in \{0, 9, 14, 19, 24, 29\}$, and in c,d) $X \in \{0, 1, 2, \cdots, 8\}$.

in the channel memory which only lead to a very small performance loss for a stuck-at fault rate as high as 10^{-3}, even without MSB protection. Fig. 3d) depicts the BER performance of the Turbo decoder in the case where we insert faults in all memory blocks. Stuck-at-0 faults greatly degrade the BER performance. However, it is also evident from the figure that protecting the MSB improves the raised error floor by an order of magnitude. A comparison between Fig. 3c) and Fig. 3d) suggests that most of the performance degradation can be attributed to the two memories within the Turbo decoder, namely state metric and extrinsic. In order to inspect which of these two memories is causing most of the performance loss due to the stuck-at faults, we performed experiments where we inserted faults only in the state metric memory or in extrinsic memory. We observed that faults in state metric memory imposes the largest performance loss. As can be seen in Tbl. II, this memory is the largest memory in Turbo decoder.

Furthermore, as explained in the previous section, we investigated the effect of having one or more repair iterations at the end of turbo decoding. Fig. 4c) and Fig. 4d) show the results for $P_{s0} = 10^{-4}$ and $P_{s0} = 10^{-5}$, respectively. Intuitively, the sooner we start the repair iteration, the better is the BER performance we achieve. The results confirm this intuition in both cases. However, we can also see that even when the repair is started as late as iteration 6 (out of 8 total iterations), we achieve a sufficiently good error floor.

As mentioned before, our simulated decoder applies a genie-aided early terminator. Tbl. III summarizes the average number of iterations in each of our experiments with Turbo decoder at SNR$=$ 1.4 dB. As seen from the table, in the case of

Table III: Number of decoding iterations executed for each scenario measured relative to the reliable memory scenario.

Exp.	P_{s0}	LDPC -0.5 dB X	LDPC -0.5 dB Avg. Iter.	LDPC -0.5 dB Δ iter. (%)	Turbo 1.4 dB X	Turbo 1.4 dB Avg. Iter.	Turbo 1.4 dB Δ iter. (%)
I	0	N/A	33.04	0	N/A	2.74	0
II	10^{-3}	N/A	50.00	51	N/A	2.85	4
	10^{-4}		50.00	51		2.75	0
	10^{-5}		38.16	16		2.74	0
III	10^{-3}	N/A	38.09	15	N/A	2.75	0
	10^{-4}		33.62	2		2.74	0
	10^{-5}		33.18	0.4		2.74	0
IV	10^{-3}	N/A	50.00	51	N/A	7.97	191
	10^{-4}		50.00	51		3.63	32
	10^{-5}		38.16	16		2.79	2
V	10^{-3}	N/A	49.50	50	N/A	4.76	74
	10^{-4}		34.72	5		2.90	6
	10^{-5}		33.10	0.2		2.75	0
VI	10^{-4}				1	2.76	1
		9	43.04	30	2	2.81	3
		14	43.93	33	3	3.18	16
		19	45.05	36	4	3.42	25
		24	46.86	42	5	3.52	28
		29	49.05	48	6	3.57	30
					7	3.62	32
	10^{-5}				1	2.74	0
		9	35.18	6	2	2.74	0
		14	35.28	7	3	2.78	1
		19	35.42	7	4	2.79	2
		24	35.94	8	5	2.79	2
		29	37.17	13	6	2.79	2
					7	2.79	2

$P_{s0} = 10^{-5}$, the average number of iterations in all of the experiments is almost the same as the reliable case (experiment I). In addition, for experiments II and III, where faults are only injected into the channel memory, we do not see a notable increase in the number of iterations. On the contrary, when the fault rate is higher than 10^{-5} and faults are injected into all the memories in the decoder, the increase in the average number of iterations is notable. In these cases, as seen from the table, enabling MSB protection and repair iterations decrease the number of iterations significantly. In case of the repair iterations, the sooner the repair is started, the more significant is the improvement in the average number of iterations.

V. Conclusions

In this paper, we quantify the impact of unreliable memories on the performance of two widely used forward error correction decoders, LDPC and Turbo decoders. We consider stuck-at-0 faults that are uniformly distributed throughout the cells of the memory modules of the decoder. Our experiments reveal that although error correction decoders are inherently error-resilient, their performance can degrade in the presence of high memory fault rates (10^{-4} and higher). Therefore, there is a need to design cost-efficient mechanisms to mitigate the effect of memory faults in these decoders.

We showed that a significant BER performance improvement is achieved by only protecting the most significant bit in memory modules of the decoders. Furthermore, we analyzed the benefit of having repair decoding iterations for both LDPC and Turbo decoders. We show that both decoders can well tolerate stuck-at-0 faults in internal memory cells if a large enough number of repair iterations follow the iterations affected by faults. Therefore, the path is paved for studying the design of decoder architectures that take advantage of operating memories in low power and unreliable modes. Coupled with the proposed BER degradation mitigation strategies this may lead to lower energy consumption with negligible BER loss.

References

[1] S. Bhunia and S. Mukhopadhyay, *Low-power variation-tolerant design in nanometer silicon*. Springer, 2011.

[2] V. Gutnik and A. Chandrakasan, "Embedded power supply for low-power DSP," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 425–435, Dec. 1997.

[3] Y. S. Park, D. Blaauw, D. Sylvester, and Z. Zhang, "Low-power high-throughput LDPC decoder using non-refresh embedded DRAM," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 3, pp. 783–794, March 2014.

[4] S. Venkataramani, V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Quality Programmable Vector Processors for Approximate Computing," in *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-46. New York, NY, USA: ACM, 2013, pp. 1–12.

[5] C. Novak, C. Studer, A. Burg, and G. Matz, "The effect of unreliable LLR storage on the performance of MIMO-BICM," in *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, Nov. 2010, pp. 736–740.

[6] C. Ngassa, V. Savin, and D. Declercq, "Analysis of min-sum based decoders implemented on noisy hardware," in *Signals, Systems and Computers, 2013 Asilomar Conference on*, Nov 2013, pp. 866–870.

[7] M. May, M. Alles, and N. Wehn, "A case study in reliability-aware design: A resilient LDPC code decoder," in *Design, Automation and Test in Europe, 2008. DATE '08*, March 2008, pp. 456–461.

[8] C.-H. Huang, Y. Li, and L. Dolecek, "Gallager B LDPC decoder with transient and permanent errors," *Communications, IEEE Transactions on*, vol. 62, no. 1, pp. 15–28, Jan. 2014.

[9] J. Geldmacher, K. Hueske, and J. Gotze, "Turbo equalization for receivers with unreliable buffer memory," in *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, Sept. 2011, pp. 1–5.

[10] J. Geldmacher and J. Gotze, "On fault tolerant decoding of Turbo codes," in *Turbo Codes and Iterative Information Processing (ISTC), 2012 7th International Symposium on*, Aug. 2012, pp. 245–249.

[11] A. Hussien, M. Khairy, A. Khajeh, K. Amiri, A. Eltawil, and F. Kurdahi, "A combined channel and hardware noise resilient Viterbi decoder," in *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, Nov. 2010, pp. 395–399.

[12] A. Morello and V. Mignone, "DVB-S2: The second generation standard for satellite broad-band services," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 210–227, Jan. 2006.

[13] *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 9).* 3GPP Organizational Partners TS 36.212 Rev. 8.3.0, May 2008.

[14] G. Falcao, M. Gomes, V. Silva, L. Sousa, and J. Cacheira, "Configurable M-factor VLSI DVB-S2 LDPC Decoder Architecture with Optimized Memory Tiling Design," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–16, 2012.

[15] Y. Sun and J. R. Cavallaro, "Efficient hardware implementation of a highly-parallel 3GPP LTE/LTE-advance turbo decoder," *INTEGRATION, the VLSI journal*, vol. 44, no. 4, pp. 305–315, Sept. 2011.

[16] C. Roth, C. Benkeser, C. Studer, G. Karakonstantis, and A. Burg, "Data mapping for unreliable memories," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, Oct. 2012, pp. 679–685.

[17] The 3rd Generation Partnership Project (3GPP), "Technical specification group radio access network; multiplexing and channel coding (FDD), tech. spec. 25.212 release-11," 2012.