

Low-Power Techniques for Flexible Channel Decoders

G. Gentile¹, M. Rovini², L. Fanucci²

¹European Space Agency,

Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands,

²Dept. of Information Engineering, University of Pisa,

Via G. Caruso, I-56122 Pisa, Italy

e-mail : giuseppe.gentile@esa.int, m.rovini@iet.unipi.it, l.fanucci@iet.unipi.it

Abstract:

This paper proposes a framework for a low-power design of flexible multi-standard channel decoders which are the most computational demanding blocks of modern communication systems. A power-efficient design envisages hardware level techniques to reduce static power consumption and algorithmic level technique to early stop the iterative decoding when the received information is estimated to be correct. Particularly, the paper focuses on two different stopping rules for Turbo codes which are well-suited for a multi-standard scenario. Simulation results indeed show an achievable power saving ranging from 50% to 80%.

INTRODUCTION

The ever increasing demand of multi-standard applications and contents is considerably affecting the computational load of new generation mobile devices for communication, resulting in higher power consumption. Low-power techniques represent then a pivotal issue towards mobility, and particular care must be paid in the design of advanced forward error correction (FEC) decoders which represent the most computationally intensive part of the whole transceiver. low-density parity-check (LDPC) [1] and Turbo [2] codes are the preferred choice of modern communication standards due to their remarkable performance approaching the Shannon limit of channel capacity [3].

Power efficient designs can be achieved at two different levels: at hardware level, by switching-off those circuitries not involved in a specific task, and at algorithmic level, by exploiting the iterative nature of the decoding algorithm to prevent useless iterations when the received codeword has been already declared correct. The last technique, known as stopping rule, can be inherently implemented with the even parity check of the estimated codeword in LDPC decoders, while different criteria have been proposed for Turbo decoders [4], [5], [6].

After an overview of the most common hardware and algorithmic techniques to reduce power consumption in Sec. II and Sec. III, Sec. IV recalls two stopping criteria for Turbo codes, the Input-Output-Consistency (IOC) [5] and the Hard-Decision Aided (HDA) [6], and proposes some algorithmic improvements for enhanced performance. Finally, Sec. V shows error correction, average speed and power saving performance of the considered solutions, by using a very likely benchmark scenario for a near-future mobile device which includes the support of UMTS-HSDPA, DVB-SH and IEEE 802.16e (WiMAX).

HARDWARE LEVEL TECHNIQUES

Addressing very-low power applications, the static power consumption of those sub-units which are not in use or in stand-by mode, is a key factor to take into account. Depending on the particular resource to “switch-off”, different techniques can be applied.

Memory blocks can be typically disabled by means of the chip select (CS) input pin when are not used. In this condition the minimum overhead of static power is guaranteed.

Focusing on registers, the clock gating technique can be applied [7]. This strategy is implemented on registers with the “enable” pin, since when a register is disabled, a particular circuitry prevents any transition on the clock input port. Furthermore, registers could be mapped on the so-called multi-threshold CMOS (MT-CMOS) retention flip-flops [8], which rely on low threshold-voltage transistors during normal operations and high threshold-voltage, low leakage transistors to hold the register state during stand-by mode.

Finally, in data-path intensive designs where the complex combinational circuits may count for the majority of the power consumption, the operand isolation technique can be used [9]. Particularly, this approach envisages the use of isolation logic (AND or OR gates) along with few control signals to hold the inputs of data-path operators to a constant value when a certain part of logic is not used. Consequently, no switching activity at the inputs propagates through the circuit and power is not unnecessarily wasted.

ALGORITHMIC LEVEL TECHNIQUES

At the algorithmic level, the iterative nature of Turbo/LDPC decoding algorithms can be exploited in order to early stop the decoding process whenever the decoder has reached a reliable estimation of the

transmitted codeword. In this case, running the remaining iterations would only result in wasting additional power, with negligible improvement on error correction performance. Figure 1 graphically depicts how the early stop is used in an iterative decoder to save power.

As an opposed approach, early stopping rules can be also exploited to increase the throughput by means of an ad-hoc system of buffers at the input and the output of the decoder [10], [11]. However, this technique does not provide any power-reduction, so it has not been considered in this paper.

Several stopping rules have been proposed in the past for Turbo codes, offering different trade-offs between implementation complexity, error correction performance and power saving. These can be classified as soft or hard-decision stopping rules, which use as an input, hard (1-bit quantized) or soft-reliabilities respectively. Among the soft rules it is worth mentioning the cross-entropy (CE) criterion [4] and the Input-Output Consistency (IOC) check [5] which have shown performance very close to the Genie rule¹.

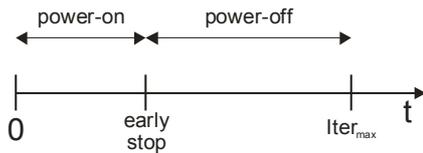


Fig. 1. Stopping rules and power consumption reduction.

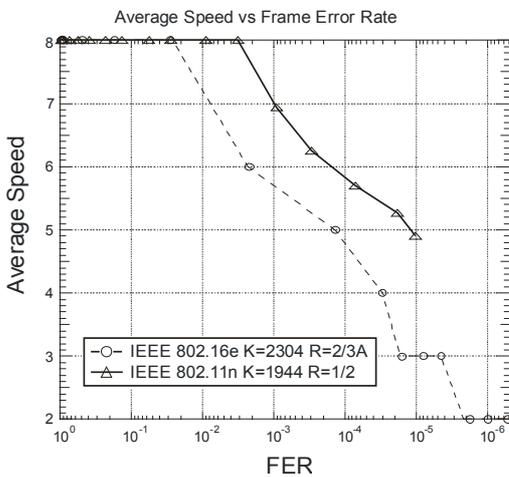


Fig. 2. Average speed vs. FER correction performance.

On the other hand, two simple hard rules have been described in [6] called hard-decision-aided (HDA) and sign-change-ratio (SCR). As far as LDPC codes are concerned, the design of an efficient stopping rule is more straightforward, since it is possible to check the even parity of each single parity-check constraint of the estimated codeword to assess the correctness of the current estimation. As an example, Fig. 2 shows

¹ the Genie rule uses the information available at the transmitter to stop the decoding in case of correct estimation.

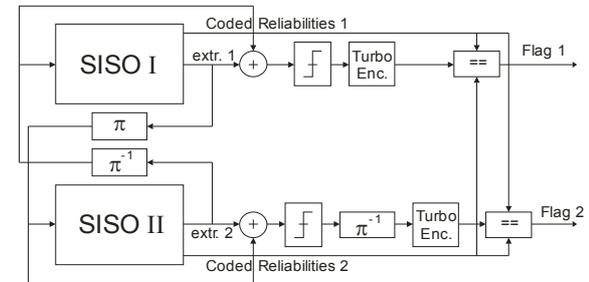
the average convergence speed of two relevant LDPC codes as a function of the frame error-rate (FER). This curve, obtained with the parity-check early stop, demonstrates that a high signal-to-noise ratio (SNR) (corresponding to low FER), the average convergence speed is significantly smaller than the maximum number of iterations. Then a significant amount of power can be saved by early stopping the decoding process.

STOPPING RULES FOR FLEXIBLE TURBO DECODERS

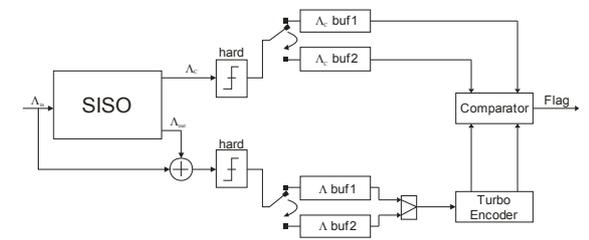
In this section, the IOC and HDA stopping rules are reviewed in terms of implementation complexity, and some modifications are proposed in order to solve possible deadlock issues at high SNR.

Input-Output Consistency Check

The reference scheme of the IOC rule is shown in Fig. 3(a). The basic idea is to check on-the-fly whether the SISO decoders are estimating the proper trellis path. In this case, each SISO decoder provides as an extra-output the a-posteriori reliability of the coded bits, which are then compared with the data stream produced by a local turbo encoder. Actually, the hard decision of the estimated information word feeds a local replica of the turbo encoder used in transmission which produces an estimated version of the coded bits. These last are compared with the hard-decisions of the coded bits coming from the SISO decoders and, once equal, convergence is declared.



(a) IOC reference scheme



(b) IOC architecture

Fig. 3. IOC rule: reference scheme and architecture.

The consistency check is performed every half an iteration, then two flags are used to signal convergence of SISO I and SISO II, respectively. Consequently two different stopping strategies can be

conceived depending on whether only one (single-consistency) or both (double-consistency) the flags declare consistency. The former strategy attains faster convergence speed while the latter, being more conservative, reduces the number of false convergences, i.e. the case when convergence is declared while the estimated word is still not correct. The detailed architecture of the IOC is shown in Fig. 3(b). Typically, only one SISO decoder is used to implement in time-division the SISO I and SISO II operations, and a system of buffers is inserted to hold the data for comparison. Particularly, during the SISO I (SISO II) phase, the hard estimation of the information word and the coded bits are stored in the upper (lower) buffers, while the encoder is fed with data produced at the previous half iteration and stored into the Λ -buf2 (Λ -buf1). Finally, the consistency is performed between the estimated coded bits output by the Turbo encoder and the contents of the Λ_c buffers.

As far as the complexity overhead is concerned, an increase of the logic resources ranging from 40% to 50% is expected due to the modification to the SISO block to produce the coded-reliabilities, while memory is estimated to increase by 15-25% due to the buffers. As shown later, these overheads could reduce the gap in power saving between the IOC and modified HDA rule, since a certain amount of power is consumed by the circuitry to implement the rule itself.

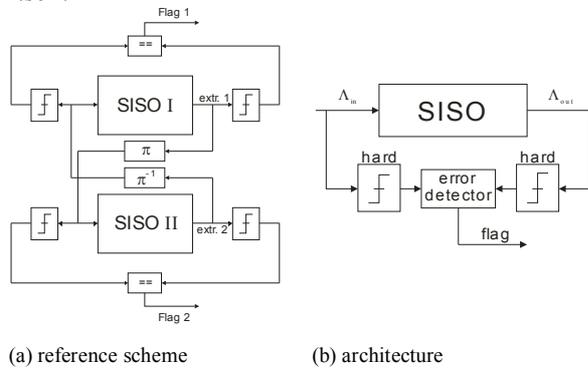


Fig. 4. Modified HDA rule: reference scheme and architecture.

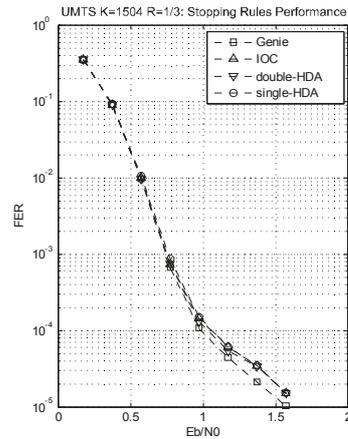
Modified Hard-Decision Aided

The reference scheme of the modified hard-decision aided (mod-HDA) stopping rule is shown in Fig. 4(a). Differently from [6], mod-HDA compares the hard decisions of the extrinsic reliabilities at the input/output of each SISO instead of the hard decisions of the total a-posteriori reliabilities at the end of each iteration. This modification is shown to improve the performance of the rule at high SNR. Similarly to IOC, convergence can be declared either when one (single-consistency) or both (double-consistency) flags are asserted. As shown later, duobinary codes require double consistency in order to

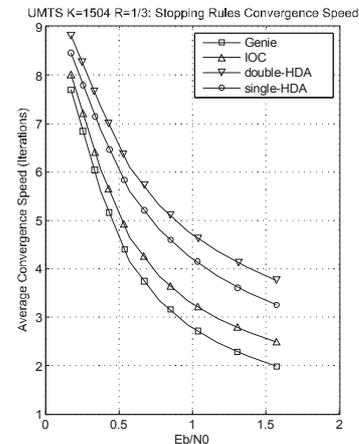
avoid floor issues, while single-consistency is well-suited for binary codes.

As far as the binary codes are concerned, error correction performance at high SNR can be spoiled by deadlock issues. Actually, after a certain number of iterations, the extrinsic reliabilities may remain stuck, and do not change with the following iterations. In this case it could happen that the extrinsic metrics produced by one SISO are very reliable towards the correct value of the transmitted bits while the ones produced by the other SISO are less reliable towards the incorrect value. The estimated information word is then correct but convergence is not declared.

To solve the problem, a threshold of five iterations has been set ($N_{th} = 5$) so that when $N_{it} \leq N_{th}$, decoding is stopped only upon consistency while, when $N_{it} > N_{th}$, the decoding is stopped whenever the number of errors, i.e. hard decisions which do not match, remains constant between two half iterations. The architecture of the mod-HDA rule, shown in Fig. 4(b) does not require memory resources or additional complexity into the SISO block. Indeed, the complexity overhead is only due to two hard decision blocks and simple logic performing the consistency check (*error detector*).



(a) FER



(b) convergence speed

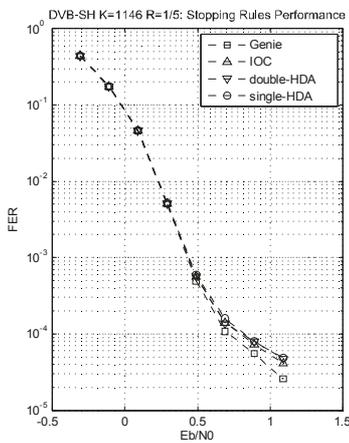
Fig. 5. UMTS: stopping rule performance.

PERFORMANCE ANALYSIS

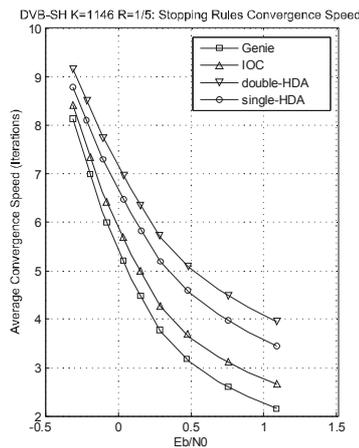
The performance of IOC and mod-HDA have been analyzed in terms of error correction capability, average convergence speed and expected power saving for three reference codes belonging to the UMTS-HSDPA, DVB-SH and WiMAX standards. Results have been carried out by considering $N_{it,max} = 10$ and the following early stop strategies have been compared with the reference Genie stopping rule:

- IOC single-consistency (IOC);
- mod-HDA single-consistency (single-HDA);
- mod-HDA double-consistency (double-HDA).

As a baseline, single-consistency strategies are preferred due to the faster convergence speed while double-consistency is applied only when the number of false convergences with the single-consistency strategies are intolerable.



(a) FER



(b) convergence speed

Fig. 6. DVB-SH: stopping rule performance.

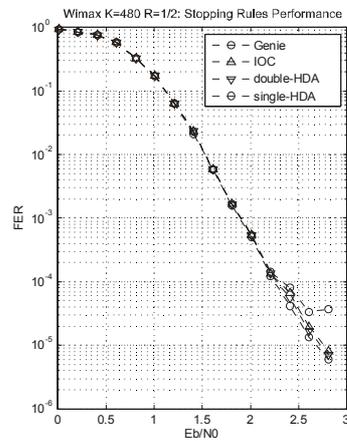
Forward Error-Correction Capabilities

The error correction and convergence speed of the UMTS reference code are shown in Fig. 5. Particularly, Fig. 5(a) shows that the three rules basically have the same performance, that is a

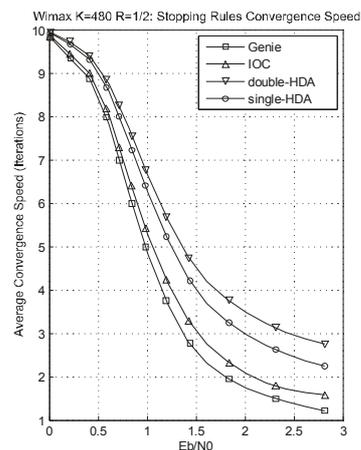
maximum loss of 0.1 dB at high SNR. IOC outperforms the other rules in terms of convergence speed (Fig. 5(b)) being only 0.5 iterations lower than the Genie rule, while the single-HDA and double-HDA feature an average impairment of 1.4 and 1.8 iterations, respectively.

As far as the DVB-SH code is concerned, the curves shown in Fig. 6 are very similar to the case of the UMTS code with a maximum error correction loss of 0.15 dB (Fig. 6(a)) with the best convergence speed reached by the IOC (Fig. 6(b)).

Finally, the performance of the WiMAX code are shown in Fig. 7. Particularly, Fig. 7(a) shows that a floor issue arises with single-HDA. Only IOC or double-HDA can be then effectively used in this case with an impairment of 0.3 and 1.4 iterations, respectively (Fig. 7(b)) and a negligible loss in error correction performance.



(a) FER



(b) convergence speed

Fig. 7. WiMAX: stopping rule performance.

Savings in Power Consumption

Since the turbo decoders have memory-dominated architecture, the power saving can be assessed by considering, at a first approximation, the power consumed by logic negligible w.r.t. that of memory. Accordingly, only the overhead due to the buffers of

the IOC rule has been taken into account. Figures 8(a), 8(b) and 8(c) show the achievable power saving as a function of E_b/N_0 for the UMTS, DVB-SH and WiMAX reference codes, respectively. According to the baseline, single-HDA rules have been considered where possible. As far as the UMTS and DVB-SH are concerned (binary codes), single-HDA and IOC are very similar at very high SNR, the power saving ranging from 50% to 70%, while IOC becomes less efficient at low SNR, due to the power consumed within buffers. On the other hand, for the WiMAX code (duo-binary), IOC outperforms double-HDA at high SNR with savings up to 80%. As a drawback, IOC increases the power consumption of the overall decoder at very low SNR, where the power added by the buffers is dominant.

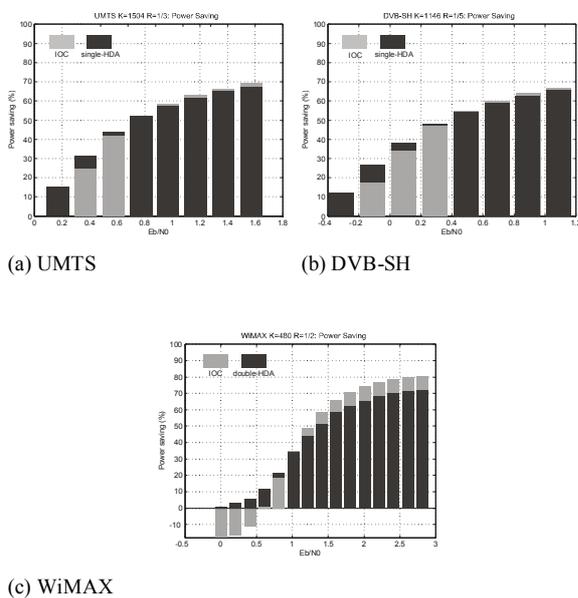


Fig. 8. Power saving performance.

CONCLUSIONS

This paper proposed a framework for power efficient designs of multi-standard flexible channel decoders. Power reduction can be achieved at hardware level by switching-off parts of the circuit when not used and at algorithmic level by early stopping the iterative decoding process upon reliable estimation of the transmitted word.

LDPC codes feature an inherent stopping rule which is the on-the-fly check of the parity constraints according to the constituent parity check matrix, while different strategies can be applied for Turbo codes. Particularly this paper revised two different rules (IOC and HDA), especially focusing on the complexity overhead and proposed a modified version of the HDA rule which solves deadlock issues for binary codes. Simulation results were carried out considering a flexible scenario including the UMTS-HSDPA, DVBSH and WiMAX standard and showed power savings up to 80%. More in detail, IOC was

shown to be more efficient at high SNR due to the low convergence speed while HDA was preferable at low SNR since almost no power is consumed for the implementation of the rule itself.

REFERENCES

- [1] R. Gallager, "Low-Density Parity-Check Codes," Ph.D. dissertation, Massachusetts Institutes of Technology, 1960.
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in IEEE International Conference on Communications, vol. 2, May 1993, pp. 1064–1070.
- [3] C. E. Shannon, "A mathematical theory of communication," Bell System Technical Journal, vol. 27, pp. 379–423, 623–656, 1948.
- [4] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," IEEE Trans. Inform. Theory, vol. 42, pp. 429–445, Mar 1996.
- [5] M. Rovini and A. Martinez, "Efficient stopping rule for turbo decoders," IEE Electr. Lett., vol. 42, no. 2, pp. 235–236, Feb 2006.
- [6] R. Shao, S. Lin, and M. Fossorier, "Two simple stopping criteria for turbo decoding," IEEE Trans. Commun., vol. 47, no. 8, pp. 1117–1120, Aug 1999.
- [7] L. Benini, P. Siegel, and G. De Micheli, "Saving power by synthesizing gated clocks for sequential circuits," IEEE Des. Test. Comput., vol. 11, pp. 32–41, 1994.
- [8] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," IEEE J. Solid-State Circuits, vol. 30, pp. 847–854, 1995.
- [9] M. Munch, B. Wurth, R. Mehra, J. Sproch, and N. Wehn, "Automating RT-level operand isolation to minimize power consumption in datapaths," in Design, Automation and Test in Europe Conference and Exhibition 2000., 2000, pp. 624–631.
- [10] E. Boutillon, C. Douillard, and G. Montorsi, "Iterative decoding of concatenated convolutional codes: Implementation issues," Proceedings of the IEEE, vol. 95, no. 6, pp. 1201–1227, Jun 2007.
- [11] M. Rovini and A. Martinez, "Iterative decoders based on statistical multiplexing," in Proc. 3rd Intern. Symposium on Turbo Codes, 1 Sep 2003.