Abstract— This paper presents the design of an analog turbo decoder for DVB-RCS-like applications using a slice architecture. The constituent decoders for different frame lengths are made up of duplicated elements chosen from a small set of reduced-size MAP decoders. This slicing technique enhances the design of the decoder in terms of simplicity, testability, re-usability and robustness. It is illustrated with the example of a turbo decoder for frames of 48 symbols sliced into two sub-frames. The error correction performance of this analog slice decoder is equal to that of the digital counterpart without slices. Transistor-level simulations show a potential throughput up to 1.5Gb/s for a 2.9mW core power consumption per information bit and per state in a 0.25µm BiCMOS process.

I. INTRODUCTION

Research interest in analog decoding has grown over the last decade due to its potential advantages over digital decoding techniques in terms of speed, power consumption and on-chip area reduction [1]–[4]. Few circuits target industrial standards: [5] proposed a decoder for UMTS and in [6] the first steps toward a DVB-RCS decoder were presented. The DVB-RCS channel coding scheme is not adapted to analog decoding since it was designed for digital circuits whose implementation has fewer constraints and is more automated. The authors propose a joint code-decoder more suitable to analog decoding for DVB-RCS-like applications.

Recently, a particular type of turbo code was proposed in [7]: slice turbo codes. In slice turbo codes, frames are split into shorter ones. Each one is independently encoded in each dimension, but they are all connected through the interleaver. [7] showed that this does not introduce performance degradation. Implementing the decoder for this type of turbo code makes it possible to concentrate on the design of a small set of reduced-size elementary decoders much smaller than those required to decode the full frame lengths. Thus, the design is simpler, easier to characterize, more robust and reusable for different frame lengths.

In this paper, a first analog implementation of a slice turbo decoder is proposed as a solution for industrial applications similar to the DVB-RCS standard, which the authors use as a reference. The paper is organized as follows. Part II presents the slice turbo architecture. In part III the performances of the Maximum A Posteriori (MAP) decoder designed in [6] are recalled and the interleaving is described. Part IV presents some simulation results obtained from a behavioral model and transistor-level simulation of the slice turbo decoder.

II. SLICE TURBO ARCHITECTURE

The principle of slice turbo codes is illustrated in Fig. 1. A frame of length \( k \) is sliced into \( m \) parts (two parts in our example). Each sub-frame, or slice, is encoded independently using the usual turbo scheme of two circular recursive systematic convolutive (CRSC) encoders and an interleaver [7]. It is worth noting that the interleaving is done over the full-length \( k \) of the initial frame. The work presented in this paper deals with the shortest frame length of the DVB-RCS standard: 48 duo-binary symbols or 96 bits [8]. This frame can be divided into two slices of equal length (24 symbols). This is the simplest case of slicing: using only one slice length is the first step to proving the validity of the slicing technique. Therefore, each CRSC encoder in Fig. 1 encodes slices of 24 duo-binary symbols. The corresponding turbo decoder block diagram is illustrated in Fig. 2.

![Fig. 1. A two-slice turbo encoder.](image-url)
A critical part in the slice architecture is the design of the interleaver [7]. A hierarchical interleaver with two levels of permutations has been designed for a frame split into $P$ slices.
Likewise, in the interleaved order, the symbol with index $l$ time symbol with index $l$ on independent consecutive blocks of $M$ symbols. A first level called the spatial permutation exchanges the symbols between the slices, and a second level called the temporal permutation shuffles the symbols within one slice as presented in Fig. 5. Hence, the $M$ symbols in one slice in the natural order are distributed over the $P$ slices in the interleaved order.

**P slices of M symbols in the natural order**

<table>
<thead>
<tr>
<th>$SN_0$</th>
<th>$SN_1$</th>
<th>$SN_r$</th>
<th>$SN_{P-1}$</th>
</tr>
</thead>
</table>

- $\Pi_T$ - $\Pi_T$ - $\Pi_T$ - $\Pi_T$

**P slices of M symbols in the interleaved order**

<table>
<thead>
<tr>
<th>$SI_0$</th>
<th>$SI_1$</th>
<th>$SI_r$</th>
<th>$SI_{P-1}$</th>
</tr>
</thead>
</table>

- $\Pi_S$ - $\Pi_S$ - $\Pi_S$ - $\Pi_S$

1) **Design of the Interleaver**: The interleaver $\Pi$ is defined by its function $\Pi(k)$ given by (1), which associates each symbol with index $k$ in the interleaved order to a symbol with index $\Pi(k)$ in the natural order. Let $l$ and $k$ denote the indices of the symbols in the natural and interleaved order, respectively. The coding process is performed in the natural order on independent consecutive blocks of $M$ symbols. The symbol with index $l$ is used in slice ${l/M}$ at temporal index $t = k \mod M$. Note that $t = M \cdot r + t$, where $r \in \{0, ..., P - 1\}$ and $t \in \{0, ..., M - 1\}$. The interleaving function is then given by:

$$\Pi(k) = \Pi(t, r) = \Pi_S(t, r) \cdot M + \Pi_T(t)$$  \hspace{1cm} (1)

2) **Design of the Temporal and Spatial Permutations**: The temporal and spatial permutations are designed to optimize the performance of the turbo code. The temporal permutation is then given by:

$$\Pi_T(t) = \alpha_{\Pi} \cdot t + \beta_{\Pi}(t \mod 4) \mod M$$  \hspace{1cm} (2)

where $\alpha_{\Pi}$ is relatively prime with $M$, and $(\beta_{\Pi}(t))_{0 \leq r < 4}$ are four coefficients smaller or equal to $M$, which verify that their values modulo 4 are all different. The spatial permutation is defined as a circular rotation:

$$\Pi_S(t, r) = (A(t \mod P) + r) \mod P$$  \hspace{1cm} (3)

where $A$ is a bijection of variable $t$ and $(\beta_{\Pi}(t))_{0 \leq r < 4}$ are four coefficients smaller or equal to $M$. The optimization process is recalled here. The choice of the $\alpha_{\Pi}$ parameter of the temporal permutation maximizes the spread of the interleaver. Then, the bijection $A$ is chosen to be irregular in order to introduce a high dispersion in the interleaver. Finally, the $\beta_{\Pi}$ parameters are chosen in order to maximize the minimum distance of the code. It has been shown in [9] that this optimized interleaver leads to very good performance and therefore slice turbo codes introduce no performance degradation compared to conventional turbo codes.

**IV. SIMULATION RESULTS**

**A. Transistor-Level Simulations**

The full two-slice turbo decoder was simulated at transistor level using the same frames used to test the MAP decoder. Fig. 6 presents the correction of the same error treated by the MAP in Fig. 4. The presented output of the MAP 11 (Fig. 2) decoder benefits from the work of MAP 12, MAP 21 and MAP 22 thanks to the exchange of extrinsic information through the interleaving network. Fig. 6 shows that the reliability of the decision in the turbo case is better than in the MAP case. Indeed, the difference between the most reliable values and the others is greater in the turbo case than in the MAP one. Fig. 6 also shows a correction within 20ns. In fact, this is the best case observed; in the other cases, the correction is done after up to 35ns. The interleaving network does not introduce latency but reduces decoding time. As it takes a further 30ns to convert the soft decision into a hard decision, the data throughput — normalized to information bits — is estimated at 1.5Gb/s, which is greater than the speed of a stand-alone MAP decoder. Moreover it is a sound performance compared to digital circuits (a few hundreds of Mbit/s). Therefore, the turbo scheme not only enhances error correction but speeds up the decoding in an analog implementation. Simulations enable the power consumption of the decoding core to be estimated at 2.9mW per information bit and per state. The
power consumption doubles as expected: each information bit is decoded by two MAPs (one in the natural order and one in the interleaved order).

B. Behavioral Modeling and Simulations

Behavioral modeling is necessary to evaluate the performance of the decoder in terms of Bit Error Rate (BER) and Frame Error Rate (FER) since, to obtain these curves, several thousand frames need to be fed to the decoder. At transistor-level this is impractical due to long simulation time (over a week per frame!). Therefore, a behavioral model for the CRSC MAP decoder was developed in Simulink. Fig. 7 shows a behavioral simulation comparing the BER/FER performance of the analog slice turbo decoder and its digital counterparts: the DVB-RCS standard decoder and the proposed slice decoder. The digital decoders run for 15 iterations and use fixed point number representation to guarantee the best MAP decoding performances. This figure actually shows that the analog turbo decoder performs better than its digital slice counterpart by 0.1dB and as good as the DVB-RCS standard digital decoder. The authors think that there are two possible explanations. First, there is a real benefit in continuously exchanging information between the CRSC decoders instead of doing it iteratively: the analog way is better for the slice implementation. Second, the convergence in each dimension of the turbo decoding scheme is better for a single long frame than for the same frame split into slices. Therefore, the standard decoder without slices benefits from a better convergence and achieves the performance of the analog slice decoder. To conclude, the slice analog decoder offers the same error correction performance as a decoder designed for the DVB-RCS standard — without slices — but with the advantages of analog implementations: greater speed, lower consumption and reduced latency.

V. Conclusion

The authors have proposed a joint code-decoder architecture for the analog implementation of DVB-RCS-like decoders. The proposed turbo code uses a slicing technique which introduces no performance degradation compared to conventional turbo codes. This slice turbo code enhances the design of the analog decoder in terms of simplicity, robustness, testability and re-usability. Indeed, one only has to design a small set of reduced-size elementary analog decoders much smaller than those required to decode the full frame lengths. Thus, an analog turbo decoder for the shortest frame of the DVB-RCS standard — 48 duo-binary symbols or 96 bits — has been designed using slices. Transistor-level simulations show a potential throughput up to 1.5Gbps for a 2.9mW core power consumption per information bit and per state in a 0.25μm BiCMOS process. Tape out for this circuit is expected in April 2005.

REFERENCES