

Design and implementation of a soft-decision decoder for Cortex codes

Patrick Adde, Christophe Jego, Raphaël Le Bidan, Ernesto Perez Chamorro Jorge

▶ **To cite this version:**

Patrick Adde, Christophe Jego, Raphaël Le Bidan, Ernesto Perez Chamorro Jorge. Design and implementation of a soft-decision decoder for Cortex codes. ICECS 2010 IEEE international conference on electronics, circuits, and systems, Dec 2010, Athens, Greece. pp.663 - 666. hal-00617881

HAL Id: hal-00617881
https://hal.science/hal-00617881

Submitted on 30 Aug 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Design and implementation of a soft-decision decoder for Cortex codes

Patrick ADDE*, Christophe JEGO*, Raphael LE BIDAN* and Jorge Ernesto PEREZ CHAMORRO**

* Institut TELECOM; TELECOM Bretagne; CNRS Lab-STICC UMR 3192
Technopôle Brest Iroise, CS 83818 29238 Brest Cedex 3
Université Européenne de Bretagne, France
email: firstname.surname@telecom-bretagne.eu

** INVIA 100 route des Houillères BP2, 13590 Meyreuil, France
email: jorge.perezchamorro@ieee.org

***Abstract*—Cortex codes are a family of rate-1/2 self-dual systematic linear block codes with good distance properties. This paper investigates the challenging issue of designing an efficient soft-decision decoder for Cortex codes. A dedicated algorithm is introduced that takes advantage of the particular structure of the code to simplify the decoding. Simulation results show that the proposed algorithm achieves an excellent trade-off between performance and complexity for short Cortex codes. A decoder architecture for the (32,16,8) Cortex code based on the (4,2,2) Hadamard code has been successfully designed and implemented on FPGA device. To our knowledge, this is the first efficient digital implementation of a soft-decision Cortex decoder.**

I. INTRODUCTION

Forward Error Correction (FEC) is a powerful technique to improve performance of digital communication systems. Nowadays, modern FEC techniques such as turbo codes or Low-Density Parity-Check (LDPC) codes allow to closely approach the ultimate limit of channel capacity on a variety of channel models, for sufficiently long code length. On the other hand, a long FEC code introduces latency which may not be tolerable for certain applications such as sensor networks. In the short block length regime (a hundred code bits or less), block codes with soft-decision decoding at the receiver side may offer a practical and efficient alternative to modern FEC schemes and traditional convolutional codes also. This requires in turn block codes with good minimum distance properties (to guarantee good error-rate performance), and, more importantly, enough structure to facilitate the design of a low-complexity soft-decision decoder. We recall here that, in contrast to classical hard-decision decoders which operate on binary values, a soft-decision decoder directly process the unquantized (or quantized in more than two levels in practice) samples at the output of the matched filter, thereby avoiding the loss of information. Soft-decision decoding may offer up to 3 dB coding gain over hard-decision decoding, but with more computational complexity.

Cortex codes are a family of rate-1/2 self-dual linear block codes first introduced in [1]. The construction of Cortex codes combines a very short mother code with a sequence of permutations to produce the parity bits. If the mother code is self-dual, the resulting Cortex code inherits from the self-dual property [2]. This implies in particular that the $(n\text{-}k)\times n$ parity-check matrix (in systematic form) of an (n, k) Cortex code can be written as $\mathbf{H}=[\mathbf{P}, \mathbf{I}_{n-k}]$, with $\mathbf{I}_{n-k}$ the $(n\text{-}k)\times(n\text{-}k)$ identity matrix and where the $(n\text{-}k)\times k$ sub-matrix $\mathbf{P}$ satisfies: $\mathbf{P}\times\mathbf{P}^t=\mathbf{I}_{n-k}$. The Cortex construction opens a broad spectrum of possibilities for the construction of either short or long structured rate-1/2 systematic linear block codes with good minimum distance properties [3].

Maximum-likelihood (ML) soft-decision decoding is known to offer the best decoding performance, but it is usually computationally intractable for most codes of practical interest. Hence sub-optimal decoding algorithms have to be considered. Belief Propagation (BP) is a soft-input soft-output decoding algorithm relying on the exchange of soft information along the edges of a graph defined by the parity-check matrix $\mathbf{H}$ of the code [4]. BP is known to closely approximate the performance of optimal Maximum A Posteriori (MAP) decoding at reduced complexity for codes with sparse parity-check matrices. On the other hand, BP performs poorly with codes like Cortex codes having high-density parity-check matrices, due to the presence of many short cycles in the graph. Another approach to soft-decision decoding, called reliability-based decoding, exploits the reliability of the received symbols to search for the most likely codeword in a reduced list of candidates. A first category of algorithms in this family applies a series of candidate error patterns in the least reliable positions in the received word followed by a hard-decision decoding step in order to correct the remaining errors (if any). This category includes the soft-decision decoding algorithms devised by Chase [5]. Another category performs successive re-encoding of a set of candidate information sequence to produce to generate the list of candidate codewords. The Ordered Statistics Decoding (OSD) approach falls into this category [6].

Recently, a simple and efficient algorithm was introduced in [7] for soft-decision decoding of rate-1/2 systematic linear block codes with a parity-check matrix $\mathbf{H}$ composed of an invertible sub-matrix $\mathbf{P}$. This algorithm can be applied to any self-dual codes. It is particularly attractive for short Cortex codes since it offers near-ML performance at low decoding complexity in this case. To our knowledge, no digital implementation of soft-decision Cortex decoders with acceptable performance has been reported up to now. In contrast, analogue decoding was successfully applied to Cortex codes by taking advantage of a particular bipartite

graph representation (Cortex graph) which can be mapped onto a network of analogue operators [8].

The remainder of the paper is organized as follows. Section II describes the construction of Cortex codes. Section III introduces the proposed decoding algorithm and its performance. The challenging issue of designing a decoder for (32, 16, 8) Cortex codes is discussed in Section IV. Finally, a FPGA prototype of the decoder is described in Section V.

II. CORTEX CODES CONSTRUCTION

The general encoding structure for Cortex codes is shown in Fig. 1 [1, 2]. The k information bits are first split into m sub-sequences of length-b bits each. Each sub-sequence is encoded by a rate-1/2 systematic base mother code such as the (8,4,4) extended Hamming code [2] or the (4,2,2) Hadamard code [9], to produce a sub-sequence of m intermediate parity bits. Then, a permutation is applied to the whole sequence of k parity bits and the encoding process is iterated several times. The final codeword is obtained by concatenating the k message bits with the k final parity bits.

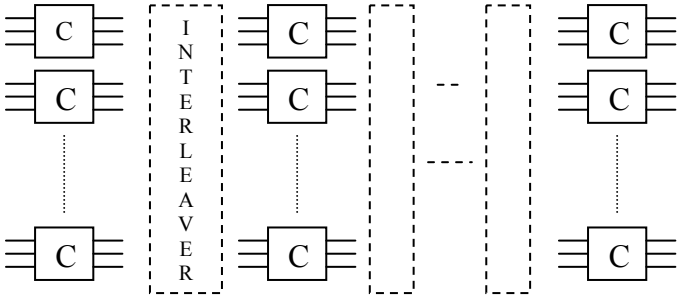
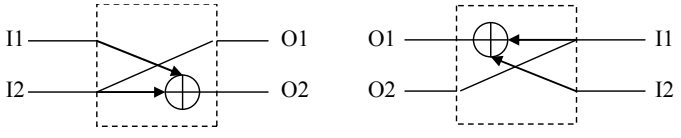


Fig.1: General Cortex encoding scheme

The minimum distance (and thus the decoding performance) of the Cortex code is influenced by the choice of the mother code, the number of encoding stages, as well as by the choice of the permutation functions. To our knowledge, no systematic design procedure has been yet proposed for the specification of the number of encoding stages and their structure. On the other hand, a method for the generation of the interleavers is described in [9]. This technique guarantees the construction of Cortex codes with the highest minimum distance for a given block length. A simple construction results by using the same mother code and interleaver at all encoder stages. The (4,2,2) Hadamard code has been chosen as the mother code in our study since it was shown in [9] to yield codes with high minimum distance. Further, the inverse of this code is easily obtained as shown in Fig. 2. Encoding equations and their reciprocals read: $O_1=I_2$, $O_2=I_1 \oplus I_2$ and $O_1= I_1 \oplus I_2$, $O_2=I_1$, respectively.



(4,2,2) Hadamard code (4,2,2) Hadamard inverse code

Fig.2: Encoding relations for the (4,2,2) Hadamard code

Fig. 3 shows how to construct a (16,8,5) Cortex code from the (4,2,2) Hadamard code, using 4 encoding stages separated by 3 identical interleavers. The minimum distance d_{min}=5 is known to be the highest possible minimum distance that can be achieved with a (16,8) binary linear block code.

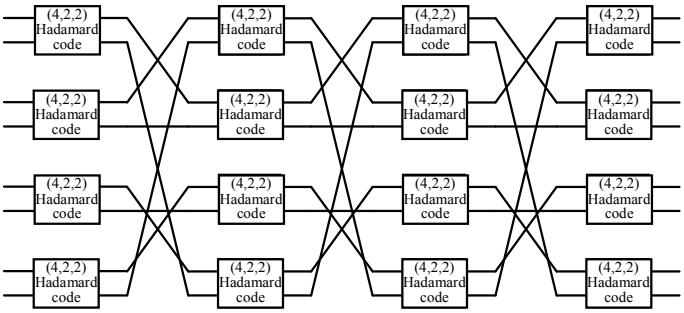


Fig. 3: (16,8,5) Cortex code built from (4,2,2) Hadamard code

III. SOFT-DECISION DECODING OF CORTEX CODES

The soft-decision decoding algorithm introduced in [7] first applies a series of *Neps* error patterns in order to try to eliminate the errors located in the least reliable positions inside the k information bits (first k bits in the received word). Each candidate information sequence is re-encoded to produce a candidate codeword. The same procedure is applied in parallel to the k parity bits, by inverting the encoding equations in order to re-compute the k message bits from the k parity bits. This produces a second list of *Nepr* candidate codewords. The decoder finally selects the candidate codeword at minimum Euclidean distance from the received word. Compared to the algorithms in [5] and [6], the proposed algorithm is computationally much simpler since it does not require a Gaussian elimination nor a hard-decision decoder. The error pattern generation strategy is similar to the one in [6].

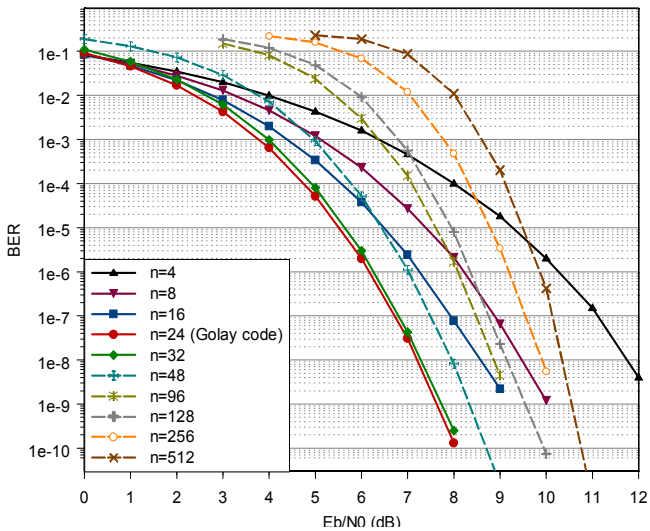


Fig.4: Soft decoding performance of different Cortex codes with Nep =32+32 and n=[4;512].

The Bit-Error Rate (BER) performance of several Cortex codes with increasing length constructed from the (4,2,2) Hadamard codes is shown in Fig. 4, assuming QPSK transmission over an additive white Gaussian noise (AWGN)

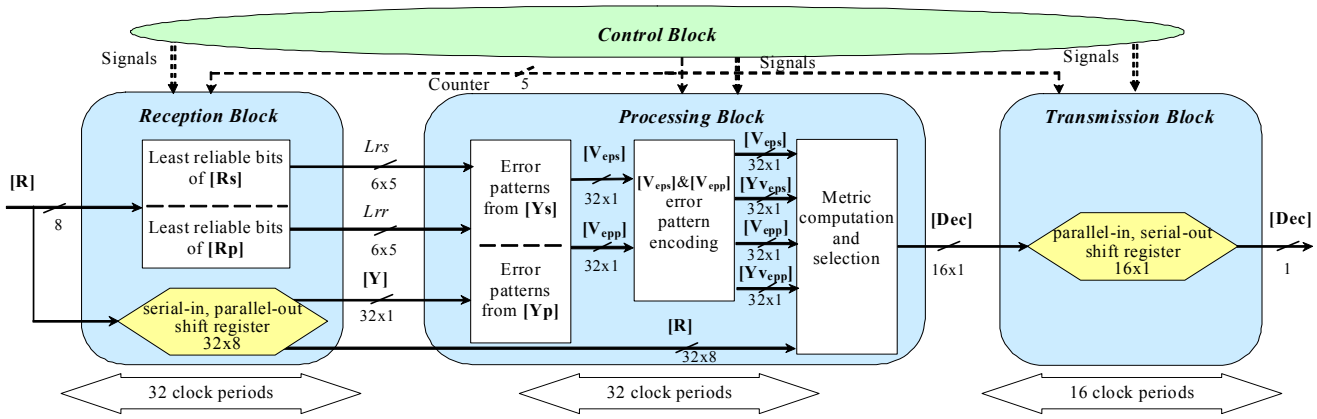


Fig.5: soft decoder architecture for a (32,16,8) Cortex code

channel. Soft-decision decoding was performed on 8-bit fixed-point received samples, using *Neps*+*Nepr*=64 error patterns for both the systematic and the parity part. Although not shown explicitly in Fig. 4, simulation results show that short Cortex codes ($n \leq 48$) essentially achieve ML decoding performance at low complexity. On the other hand, a higher number of error patterns is required to maintain performance close to ML soft-decision decoding with longer Cortex codes ($n > 48$).

IV. SOFT DECODER ARCHITECTURE FOR CORTEX CODES

Designing soft decoding architectures for Cortex codes is a challenging issue. In this section, a digital implementation of a decoder based on the proposed soft decoding algorithm for (32,16,8) Cortex code is detailed. The decoder uses information quantized on 8 bits with $L_{rs}=L_{rp}=6$ least reliable symbols and $Nep = Neps+Nepr = 32+32$ error patterns. The decoder architecture consists of four blocks: reception, processing, transmission and control as shown in Fig. 5. In the reception part, the n=32 binary symbols of the received word are processed sequentially. So, this block computes successively positions of the L_{rs} and L_{rp} least reliable symbols for the systematic and parity parts of the received word. In parallel, serial-in parallel-out shift register memorizes sequentially the 32 symbols of the received word. The processing part is composed of three main tasks. First, error patterns are generated from the sign bits of the received word by testing different combinations of 0 and 1 in the least reliable bit positions. Then, these error patterns are encoded. Finally, a selection function allows the maximum likelihood code word to be found. This selection is done from metric values computed for each error pattern. Note that a parallel processing is done for *Neps* and *Nepr* error patterns. Moreover, as the 32 binary symbols are available thanks to the serial-in parallel-out shift register, the metric of each encoded error pattern can be directly computed. The transmission unit is only composed of a parallel-in serial-out shift register. This operator enables to sequentially transmit the systematic part of the binary decoded word.

The three previous blocks are supervised by a control block. In our design, this task is achieved by a 5-bit counter that generates some control signals. The soft decoder architecture is structured in two pipelined stages: reception and processing-transmission as shown in Fig. 5. The first stage sequentially processes n=32 binary symbols in 32 clock periods. The second stage is composed of two main blocks. *Neps* and *Nepr* error patterns are generated, encoded and compared in 32 clock periods thanks to a parallel processing. Finally, the 16 decoded bits are sequentially transmitted in 16 clock periods. The decoder latency, L, can be defined as the symbol number processed by the decoder during the decoding of one symbol. It depends on the number of pipeline stages and the codeword length n. According to this definition, the resulting latency of the decoder architecture is $L=2n=64$ clock symbols.

Logic syntheses for the hardware complexity estimations were performed using the Synopsys tool with a STMicrolectronics 0.09 µm CMOS process ASIC target. Soft decoder is clocked at *f=200MHz*. Table 1 gives the complexity in terms of equivalent 2-input (NAND) gate count for each function of the soft decoder.

STMicrolectronics 90 nm CMOS	soft decoder functions	equivalent gate count
reception block	least reliable bits	1 644
	serial-in, parallel-out shift register 32x8	2 679
processing block	error pattern construction	2184
	error pattern encoding	
	metric computation and selection	4 043
transmission block	parallel-in, serial-out shift register 16x1	194
control block	5-bit counter	47
soft decoder complexity	10 791	

Table 1: hardware complexity of the soft decoder

As shown, the more complex function is the metric computation and selection. Its complexity is almost the half of the total soft decoder complexity. By analyzing the architecture of this critical part, two parameters appear to directly affect its complexity: *Nep* (error patterns) and L_r (least reliable binary symbols). The number of quantization bits q has also a direct impact on the decoder complexity and, in particular, on the shift register sizes. Decreasing these

parameter values improves area-efficiency while the performance could decrease. A complexity analysis of the soft decoder has to be done to lead to low a complexity architecture with negligible performance degradation.

V. FPGA PROTOTYPE OF DESIGNED SOFT DECODER

In order to validate the designed soft decoder, BER performance measures have to be carried out. For this reason, we have integrated the proposed decoder into an experimental setup. The experimental setup is a development board from Dinigroup that contains 6 Xilinx Virtex5 LX330 devices. All the components of the experimental setup were implemented onto only one of the FPGAs. A Pseudo Random Generator (PRG) sends out pseudo random data streams at each clock period f_0. This module is composed of flip-flops and XOR gates. An (32,16,8) Cortex code encoder processes the data streams. The last task of the transmitter is a BPSK mapping. We have chosen to integrate an AWGN channel from a white Gaussian noise generator adapted to hardware implementation. The Wallace method generates a variable that is normally distributed from a pool of variables that follows a normal distribution as well. High accuracy, speed and low-cost hardware are reached using this method as explained in [10]. Each output sample is a 5 bit vector to be sent in one clock period (f_0). The Signal to Noise Ratio (SNR) is controllable via a computer that calculates the equivalent variance square root σ and sends it to the experimental setup thanks to a PCI bus. Computation resources of the complete system take up 2,116 slice Flip-Flops and 3,974 slice LUTs. Moreover, 7 DSP resources and 1 BlockRAM of 18kbits are also necessary. The soft decoder occupies 1,114 slice LUTs, 2,905 slice Flip-Flops and 0 BlockRAM of 18Kbs. The maximum frequency estimated after place and route is 72 MHz for the soft decoder, resulting in an input data rate of T_{in}=72 Mb/s. However, taking into account the code rate R=0.5, the output data rate becomes T_{out}=36 Mb/s.

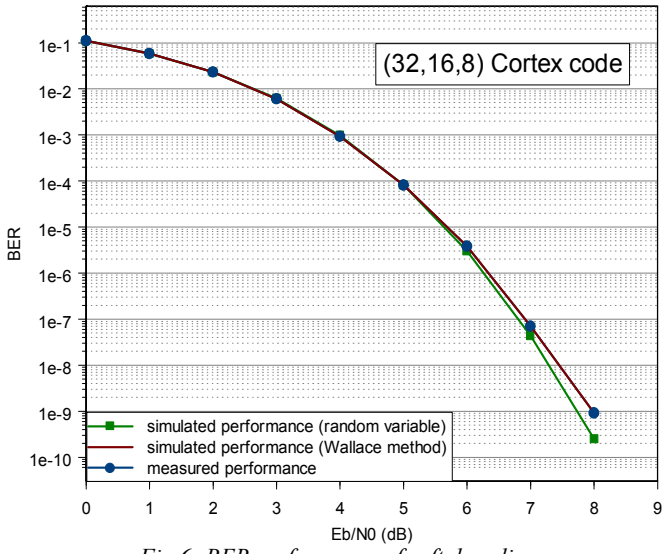


Fig.6: BER performance of soft decoding of a (32,16,8) Cortex code with Nep =32+32

Simulated performance of ML soft decoding algorithm and measured performance of designed soft decoder for the (32,16,8) Cortex code are plotted in Fig. 6. Performance is evaluated by Monte-Carlo simulation and hardware emulation using our experimental setup, respectively. Monte-Carlo simulations are done using a random variable and Wallace method for the AWGN channel. The prototype shows identical performance when compared to fixed-point simulation based on Wallace method, for *Nep*=32+32 error patterns.

VI. CONCLUSION

Soft-decision decoding of short Cortex codes has been investigated. A soft-decision decoding algorithm has been introduced which exploits the code structure to achieve ML performance using a small number of error patterns. The digital implementation of this algorithm has been described for a specific Cortex code. The simulation results and the hardware complexity of the prototype demonstrate the practicality and the benefits of the proposed decoding algorithm in this context.

ACKNOWLEDGMENT

The authors wish to thank Pierre-Louis Bullot, Romain Ducaffy, Rémy Lopez, Angela Navarro and Wiklund Xavier (Telecom Bretagne students) for their significant help during the architecture design of the soft decoder.

REFERENCES

[1] J. C. Carlach and C. Vervoux, "A new family of block turbo-codes," in *13th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, Honolulu, USA, Nov. 1999.

[2] J.-C. Carlach and A Otmani, "A systematic construction of self-dual codes, " *IEEE Transactions on Information Theory*, Vol. 49, Issue 11, Nov. 2003, pp. 3005 – 3009.

[3] G. Olocco and J. Tillich, "A family of self-dual codes which behave in many respects like random linear codes of rate 1/2," *in Proceedings of the IEEE ISIT*, 24-29 June 2001, p. 15.

[4] F. R. Kschischang, B. J. Frey, H.-A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, Feb. 2001, pp. 498-519.

[5] D. Chase, "A class of algorithms for decoding block codes with channel measurement information", *IEEE Trans. Inform. Theory*, vol IT-18, Jan. 1972, pp. 170-182.

[6] M. P. C. Fossorier & S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics", *IEEE Trans. Inform. Theory*, vol. 41, n°5, pp. 1379—1396, Sept. 1995

[7] P. Adde, C. Jégo, J. E. Perez Chamorro, R. Le Bidan, "Maximum likelihood soft decoding of systematic block codes: from algorithm to architecture", submitted at electronics letter.

[8] J. E. Perez Chamorro, F. Seguin, C. Lahuec, M. Jezequel, "A subthreshold PMOS analog cortex decoder for the (8,4,4) Hamming code," *journal of the electronics and telecommunication research institute*, Oct. 2009, vol. 31, n° 5, pp. 585-592.

[9] J. E. Perez Chamorro, F. Seguin, C. Lahuec, M. Jezequel, G. Le Mestre, "Decoding a family of dense codes using the sum-product algorithm and subthreshold PMOS, " *IEEE ISCAS*, Taiwan, 24-27 may 2009, Taipei, Taiwan, 2009, pp. 2685-2688.

[10] O. Sanchez Gonzalez, M. Arzel, C. Jégo, A. Garcia, M. Guerrero, "Design and implementation of a MIMO channel emulator onto FPGA device", *IWS'09: XV proyecto Iberchip*, 25-27 March, 2009.