Design, Simulation, and Testing of a CMOS Analog Decoder for the Block Length-40 UMTS Turbo Code

Alexandre Graell i Amat, Member, IEEE, Sergio Benedetto, Fellow, IEEE, Guido Montorsi, Senior Member, IEEE,
Daniele Vogrig, Member, IEEE, Andrea Neviani, Member, IEEE, and Andrea Gerosa, Member, IEEE

Abstract—In this paper, we present an all-analog implementation of the rate-1/3, block length 40, universal mobile telecommunications system (UMTS) turbo decoder. The prototype was designed and fabricated in 0.35 μm complementary metal–oxide–semiconductor technology and operates at 3.3 V. We also introduce a discrete-time first-order model for analog decoders which allows fast bit-error rate simulations, while taking into account circuit transient behavior and component mismatch. The model is applied to the rate-1/3 analog turbo decoder for UMTS defined in the Third Generation Partnership Project standard, and the discrete-time model predictions are compared with the decoder experimental performance and the transistor-level simulations. These results demonstrated that this model can be successfully used as a tool to both predict analog decoder performance and give design guidelines for complex decoders, for which circuit-level simulations are impractical.

Index Terms—Analog decoding, device mismatch, discrete-time model, iterative decoding, turbo codes.

I. INTRODUCTION

RECENTLY, two research groups led by Hagenauer and Loeliger noticed independently that soft-input soft-output (SISO) algorithms such as Bahl–Cocke–Jelinek–Raviv (BCJR), the soft-output Viteri algorithm (SOVA), and the sum-product algorithm in general, may lend themselves to an analog very-large-scale integration (VLSI) implementation [1], [2]. Envisaged advantages of analog decoders were higher decoding speed and lower power consumption, rendering such an approach attractive for applications that require very-high-speed and very-low-power decoders that cannot be built with digital technology. Since then, much effort has been spent towards turning these ideas into working chips. Several prototype analog VLSI chips of simple decoders are already available in the literature [3]–[6], showing an outstanding improvement in the power efficiency with respect to their digital counterparts, with only some error-correcting performance loss. However, all these implementations are limited to proof-of-concept decoders, with very short block lengths, and thus, they are not suitable for practical applications. The feasibility of the analog approach for larger, powerful code structures is still not well understood. Typical problems of large chips, such as maximum interconnect length, crosstalk, substrate noise, etc., must be taken into consideration in order to demonstrate the effectiveness of the analog approach to decoding.

A first step toward real-world analog decoders was made in [7] and [8], where a complementary metal–oxide–semiconductor (CMOS) all-analog turbo decoder for hard disk drives was considered. More recently, Gaudet and Gulak [6] implemented the first analog turbo decoder, which is a significant step ahead with respect to a decoder for a single convolutional code, but still far from a real application, due to the limited interleaver size (16 bits). In this paper, we discuss the design, implementation, and test results of an all-analog turbo decoder for a realistic application, the rate-1/3 universal mobile telecommunications system (UMTS) turbo code defined in the Third Generation Partnership Project (3GPP) standard [9], with interleaver length \( N = 40 \).

One of the main problems of the analog implementation of turbo decoders (and high-complexity codes, in general) is the impossibility of obtaining reliable estimations of circuit performance, due to the large size of the analog circuit. This, in turn, increases the difficulty of a proper circuit design optimization, since it must be done based on reasonings on much simpler circuits. Indeed, there is no simple way to relate system-level specifications [e.g., bit-error rate (BER)] and transistor-level parameters. Unfortunately, for such large circuits, running SPICE-like simulation of the whole decoder is impractical. Even higher abstraction-level models with a mixed behavioral/structural description (VHDL-AMS) result in an excessive computational load. Therefore, developing high-level tools which permit simulation of the circuit behavior, in order to facilitate design optimization and estimate chip performance, is a topic of great importance. As a salient feature of this paper, we present a discrete-time linear first-order model of the analog decoding networks which allow running very fast simulations of complex decoders, while taking into account circuit transient behavior and transistor mismatch. The model is applied to the rate-1/3 analog turbo decoder for UMTS defined in the 3GPP standard [9], and the discrete-time model predictions are compared with

Paper approved by S. G. Wilson, the Editor for Multicarrier Modulation of the IEEE Communications Society. Manuscript received August 1, 2004; revised August 29, 2005. This work was supported in part by PRIN under Project 2001091242 and in part by the MEC under the Juan de la Cierva contract. This paper was presented in part at the IEEE International Symposium on Information Theory, Chicago, IL, June 2004, and in part at the IEEE International Conference on Communications, Seoul, Korea, May 2005.

A. Graell i Amat was with the Department of Electronics, Politecnico di Torino, 10129 Torino, Italy, and with the Department of Technology, Universitat Pompeu Fabra, 30100 Barcelona, Spain. He is now with the Electronics Department, GET/ENST-Bretagne, 29238 Brest Cedex 3, France (e-mail: alexandre.graell@cnst-bretagne.fr).

S. Benedetto and G. Montorsi are with the Department of Electronics, Politecnico di Torino, 10129 Torino, Italy (e-mail: benedetto@polito.it; montorsi@polito.it).

D. Vogrig, A. Neviani, and A. Gerosa are with the Department of Information Engineering, Università di Padova, 35131 Padova, Italy (e-mail: daniele.vogrig@dei.unipd.it; neviani@dei.unipd.it; gerosa@dei.unipd.it).

Digital Object Identifier 10.1109/TCOMM.2006.884836
the decoder experimental performance and the transistor-level simulations. These results demonstrate that this model can be successfully used as a tool to both predict analog decoders’ performance and give design guidelines for complex decoders, for which circuit-level simulations are impractical. Several analyses of decoder performance using the proposed model are carried out, such as the evaluation of decoder performance degradation caused by device mismatch, and the performance loss with respect to the digital implementation.

The paper is organized as follows. Section II describes the encoder, the architecture of the analog turbo decoder, and the basics of the analog decoding algorithm. Details on the circuit-level design are discussed in Section III. Section IV introduces the discrete-time model of analog decoding networks, and reports several analyses of decoder performance based on this model. Section V addresses performance degradation due to mismatch, and Section VI presents test results of the chip. Finally, Section VI concludes this paper.

II. SYSTEM DESCRIPTION

In this paper, we consider the analog implementation of the rate-1/3 UMTS turbo code defined by the 3GPP standard [9]. The coding/decoding scheme is reported in Fig. 1. It consists of the parallel concatenation of two identical eight-state systematic recursive convolutional encoders. Each encoder is terminated according to the standard, i.e., 12 tail bits are generated. In order to limit the complexity of this first attempt to build an analog decoder for a realistic application, a block length of N = 40 bits is assumed, which corresponds to the shortest length defined in the standard. The corresponding decoder consists of two SISO decoders matched to both the constituent encoders.

In the analog domain, the SISO decoders depicted in Fig. 1 are realized as networks of interconnected cells that implement the fundamental computations of the decoding algorithm. The analog network of a single SISO decoder is depicted in Fig. 2. The network consists of 7 arrays of 40 blocks each, in rows. Each column corresponds to a bit line (or trellis section). There are only six different building blocks, each implementing one fundamental computation of the SISO algorithm. Cells F and A at the top perform the branch metrics computations, cells B and C perform the forward and backward recursions, respectively, and Cells D, E, and F at the bottom of the figure compute the extrinsic information and the a posteriori probabilities (APPs) on information bits, respectively. Notice that three extra Cells A and F are required to implement the encoder zero termination.

Based on the current-mode approach proposed by Loeliger et al. [2], each cell within the analog decoding network is realized with a circuit, derived from the well-known Gilbert multiplier [10], of metal–oxide–semiconductor field-effect transistors (MOSFETs) operating in the weak inversion region, with a one-to-one correspondence between the trellis diagram representing the function that implements the cell, and the circuit topology. Basically, each of these circuits, usually called the sum-product module, takes current signals as inputs, performs sums and products between these signals, and produces current output signals. The current signals, normalized to a reference current, represent bit probabilities. Therefore, the current-mode approach yields a natural implementation of digital decoders in the multiplicative domain, with currents representing probabilities.

An example of a sum-product cell at transistor level is shown in Figs. 3 and 4, reporting the trellis section of an E cell (Fig. 3), and the corresponding CMOS implementation (Fig. 4). Cell E
completes the computation of the extrinsic information of information bits \( \hat{p}(u = 0) \), \( \hat{p}(u = 1) \) according to the trellis of Fig. 3, where \( \phi(\cdot) \) is the product of state metrics computed by cell D (see Fig. 2), and \( p(y|f_k) \) are the channel-transition probabilities of the parity bits. In the analog implementation (Fig. 4), the probabilities \( p(y|f_k) \), \( \phi(\cdot) \), and \( \hat{p}(\cdot) \) are represented by the current vectors \( I_X \), \( I_Y \), and \( I_Z \), respectively. The nMOSFET at the bottom generates the normalization current \( I_B \). The 16 nMOSFETs in the middle are operated in the weak inversion region, and generate every pair of products of the two input current vectors \( I_X \) and \( I_Y \), i.e., perform the multiplication of the incoming probability vectors \( p(y|f_k) \) and \( \phi(\cdot) \). In the upper part of the circuit, the currents representing the products are summed, by simply shorting the corresponding wires, or discarded, connecting the wire to \( V_{DD} \), if they are not needed in the sum. The upper pMOSFETs mirror the output currents \( I_Z \) to make them available to the cascaded stages. Note that there is a topological correspondence between the transistor network in Fig. 4 and the trellis section in Fig. 3. The pMOSFET switch \( M_R \) (see the top of Fig. 4) was added to reset the extrinsic information computed by E cells to a uniform probability distribution before a decoding operation is started. This solution helps to recover from highly unbalanced initial states, thus improving the decoder speed [8], [11].

The complete transistor network needed to implement a SISO decoder is built directly by cascading the sum-product modules according to the diagram reported in Fig. 2. When two cells are cascaded, high-output-impedance, saturated pMOSFETs drive the output currents into diode-connected nMOSFETs. The overall analog decoding network of the turbo decoder is finally obtained by connecting, via interleaver/deinterleaver structures, two decoders like the one depicted in Fig. 2. The interleave is realized as a network of independent metal paths that permute the SISO1 and the channel output currents before they are fed to the SISO2 input terminals, according to the same permutation scheme adopted in the encoder. The deinterleaver is an identical network performing the opposite operation on the information flowing from SISO2 to SISO1.

Both SISO decoders are fed with 126 differential currents corresponding to the 86 channel outputs of code bits, \( p(y_k|x_k) \) (40 systematic bits, 40 parity bits, and 6 termination bits), and the 40 extrinsic values fed back by the other decoder, \( \hat{p}(u_k; I) \). At its output, the analog decoder provides the 40 differential currents corresponding to the a posteriori reliabilities, \( \hat{p}(u_k) \), and the extrinsic information, \( \hat{p}(u_k; O) \).

We emphasize the inherent differences between the decoding algorithm implemented in the analog domain and the iterative (turbo) decoding algorithm implemented by a classical digital decoder. In analog turbo decoders, iterations are replaced by an exchange information in continuous time; the analog network floods freely and stabilizes to a state hopefully corresponding to the transmitted data sequence.

### III. Circuit Design Issues

A prototype of the decoder, with block length 40, was designed and fabricated in a 0.35 \( \mu \)m CMOS technology with a single power supply of 3.3 V. A block diagram of the circuit, including the decoder core and the input/output (I/O) interface, is shown in Fig. 5. The I/O interface basic function is to store a frame of channel output symbols \( y_k \) on an analog memory, convert them to currents representing the channel transition probabilities \( p(y_k|x_k) \), perform decisions on the APPs computed by the decoder, and manage the output of the decoded bits \( u_k \).
The main challenge in the design of an analog decoder of realistic size is that transistor-level simulations are too computer-intensive to permit the prediction of the bit-error rate (BER) performance in an acceptable time. Thus, a research effort to develop a proper design procedure that yields results of predictable quality is highly beneficial. A possible solution is to follow a mixed, bottom-up and top-down approach, based on the use of high-abstraction-level models for the description of the system in conjunction with a library of basic building blocks (the sum-product modules) fully characterized at transistor level. This will be discussed in Section IV.

The analog decoder design was targeted at the minimization of power dissipation, while satisfying the data-rate requirements of the UMTS standard. The very basic design consideration was that transistor sizing has opposite effects on device matching (and thus, on the decoder static accuracy) and on parasitic capacitances (and thus, on the decoder speed). As a consequence, the minimum device size in the sum-product cells should be set based on matching considerations, then the cell bias current should be fixed so as to achieve the required speed.

The decoder model needed to implement the design procedure must then satisfy the following requirements:
1) it should take into account the circuit delay by modeling its behavior in the time domain;
2) it should take into account the effect of device mismatch on the analog decoder static response;
3) its computational efficiency should be significantly improved with respect to that of the transistor-level simulation, while maintaining a reasonable accuracy in the prediction of the decoder performance.

Based on these considerations and on our experience with previous circuits [7], [8], a MOSFET channel width $W = 5 \mu m$ and length $L = 0.5 \mu m$ is estimated. Once $W$ is fixed, the bias current $I_B$ is chosen in order to ensure a good exponential behavior and satisfy UMTS standard speed requirements. This approach produced an optimal value $I_B = 1 \mu A$.

The resulting chip size is 4.5 mm x 2 mm (excluding I/O pads), with approximately 40 000 transistors in the I/O interface and 26 000 transistors in the decoder core. The measured power consumption is 10.3 mW (6.8 mW for the decoder core alone), at a voltage supply of 3.3 V and at a channel data rate of 2 Mb/s.

IV. DISCRETE-TIME MODEL

Extended circuit-level simulations of large decoders, such as the one discussed in this paper, are impractical to carry out due to the high complexity of the analog circuit. Consider that estimating the nominal (i.e., without taking the effect of mismatch into account) BER of the decoder at $E_b/N_0 = 2$ dB would take at least two weeks of CPU time on a state-of-the-art, single-processor computer. Therefore, developing high-level simulation tools which permit simulation of the decoder behavior in an acceptable time are of extreme interest. In this section, we propose a discrete-time model of the analog decoding network based on characterizing each single cell as a linear filter [13]. The model of the single cell is then properly incorporated within a high-level C++ simulator, which allows discrete-time simulations in a very short time. According to circuit-level simulations, the model estimates very well the behavior of the analog decoder in the time domain. A method based on the successive over relaxation procedure, which is somehow related to the model presented here, has been independently proposed by Hemati and Banihashemi in [14].

As a first approach to estimate the analog decoder behavior, we model each single cell within the analog network (see Fig. 2) by a linear filter with a single pole. We therefore assume that each cell is described by an impulse response

$$h(t) = Ke^{-t/\tau}.$$  \hspace{1cm} (1)

We are particularly interested in a discrete-time model of the decoder to be embedded into a high-level simulator. Thus, we proceed to obtain the digital filter from the analog filter described by (1). Applying the Z-transform to the impulse response (1), we obtain the transfer function of the filter in the discrete-time domain

$$H(Z) = K \sum_{k=0}^{\infty} \frac{e^{-kT_\text{t}}}{z^k} = \frac{KZ}{z - e^{-T/\tau}} = \frac{K}{1 - e^{-T/\tau}z^{-1}}$$ \hspace{1cm} (2)

where we defined $\alpha = 1 - e^{-T/\tau}$, $K = \alpha$, and $T$ is the sampling period. Finally, from (2), we can write the differential equation that characterizes the cell in discrete time as

$$I'_z = H(Z)I_z = \frac{\alpha}{1 - (1 - \alpha)z^{-1}}I_z \Rightarrow I'_z = \alpha I_z + (1 - \alpha)I'_z z^{-1} \Rightarrow I'_z[n] = \alpha I_z[n] + (1 - \alpha)I'_z[n-1]$$ \hspace{1cm} (3)

where $n$ is the discrete-time index, and $(1 - \alpha)$ is the pole in the cell-transfer function in the z-transform domain. According to (3), each discrete-time cell is completely characterized by the single parameter $\alpha$, which defines the rapidity of its step response. For each cell type, we calculate $\alpha$ from the $\tau$ of the analog filter that models the corresponding analog circuit. To compute $\tau$, we model the step response of the cell by a single-pole filter: we assume that at time $t < 0$, the cell has inputs equally likely, corresponding to uniform input currents. The cell is then fed with an ideal step response which turns on a certain output, i.e., we force a certain output current to evolve to its maximum normalized value 1. $\tau$ is properly computed by approximating this response by the step response of a single-pole filter. Finally, $\alpha$ is obtained through $\alpha = 1 - e^{-T/\tau}$. The choice of a first-order model is a tradeoff between accuracy and computational efficiency. We can expect the best from the linear approximation when the input currents are sufficiently uniform, i.e., the sum-product cell is in a highly balanced state, which is the case when the decoding process of a new codeword starts, or when the APP of a given bit is close to 0.5: in this case, the accuracy is highly desirable.

The cells modeled according to (3) are incorporated into a discrete-time simulator that reproduces all the computations performed by the analog decoder. Therefore, the proposed discrete-time simulator does not perform iterations as a traditional digital turbo decoder, but performs a step-by-step
decoding, all cells performing their computations “in parallel.”

The computation of the output APPs and hard decisions is performed at each step, as well. This decoding procedure can be thought as similar to a sampling of the analog decoding process.

In order to validate the proposed model, we compared the evolution of the normalized output currents of the UMTS analog turbo decoder (obtained from circuit-level simulations) with the evolution of the APPs derived from the discrete-time model simulator. As an example, in Figs. 6 and 7, we plot the output currents trajectories and the APP trajectories for a particular frame, simulated at circuit level and by the high-level simulator, respectively. In Fig. 6 we plot the output normalized currents simulated at circuit level for all 40 information bits. Each normalized current can be interpreted as the APP of a certain bit being zero, $\hat{p}(\hat{u} = 0)$. For an additive white Gaussian noise (AWGN) channel with $E_b/N_0$ of 3 dB, the decoder corrects the 20 bits that would have been incorrect if hard decisions were taken. The plot is characterized by a transient after all currents evolve to the maximum and minimum values, corresponding to the highest reliability. The frame is completely corrected after a decoding time lower than 0.9 $\mu$s. The curves obtained from the high-level model discussed here are plotted in Fig. 7. In spite of the simplicity of the model, a significant agreement between the two simulations is observed. Indeed, the current trajectory of each particular bit in Fig. 6 translates into a very similar probability trajectory in Fig. 7. Only slight differences on both delays and probability values associated to a certain bit are observed between the circuit-level simulator and the discrete-time simulator. Simulations of other frames reported similar results. The agreement is even higher for “cleaner” frames. We also simulated several frames encoded with the eight-state rate-1/2 constituent encoder, with a very good agreement between discrete-time simulations and circuit-level simulations.

On the other hand, for very noisy codewords, corresponding to low $E_b/N_0$, that the decoder is not able to correct even after a long decoding time, output currents tend to fluctuate in a chaotic fashion. A similar behavior is observed in the evolution of the probabilities obtained with the discrete-time model, but in this case, the correspondence between the trajectories given by the two decoders is lost, in part. For frames in error after a long decoding time, we evaluate the correspondence between the two decoders by plotting the positions of the bits in error within the frame. We simulated a sample of 20 000 frames at $E_b/N_0 = 2$ dB with the discrete-time model, and focused upon 126 frames that the decoder is not able to correct. These frames show a chaotic behavior of the output probabilities. The same 126 frames have been then simulated at circuit level, also showing a chaotic evolution of output currents, but with low correspondence, as explained before. In Fig. 8, we plot the bit-error positions within the frame for 25 of these 126 frames in error. Although some differences can be appreciated between the two simulations, a significant correspondence is observed, encouraging the validity of our model. Overall, the correspondence between the circuit-level simulation and the discrete-time model for the error bits within the 20 000
simulated frames is 75%. However, our belief is that the discrepancies between the two simulations in these cases are not very significant, since they are due to a chaotic behavior at low $E_b/N_0$ values. Indeed, the outputs for some frames oscillate around the decision threshold.

Based on these results, the discrete-time model seems to approximate well the behavior of the analog decoder. According to this, it is possible to exploit the model to estimate performance of the analog decoder in an acceptable time. The advantages of the model presented here are twofold. First, it allows estimating BER performance of complex analog decoders, for which it is not possible to obtain an estimation with circuit-level simulations in a reasonable time. Second, and even more important, in the case of large, high-complexity decoders, it can be used on top of the circuit design as a tool to provide valuable guidelines for the design optimization of the circuit.

We used the discrete-time model to estimate decoder performance and to carry out several analyses on the performance degradation caused by device mismatch, and the performance loss with respect to the digital implementation.

A. Analog Decoder Performance

We predicted the performance of the rate-1/3 UMTS turbo analog decoder, in terms of BER and frame-error rate (FER), using the discrete-time model described above. In Fig. 9, we plot the estimation of the BER performance obtained from the discrete-time model. As a reference, we plot the BER curve of the digital decoder with 10 iterations. We plotted the BER curves for various decoding steps, corresponding to different decoding times. The figure shows that performance increases with the number of decoding steps, corresponding to a longer decoding time. One decoding step of the discrete-time model corresponds to a decoding time of 6.5 ns. Performance of the analog decoder tends to the ideal performance of the digital decoder as the decoding time increases. For instance, the curve corresponding to 1500 decoding steps (9.75 μs) matches the curve of the digital decoder with 20 iterations. A decoding time of 9.75 μs corresponds to a channel data rate about 13 Mb/s. Moreover, notice that Fig. 9 shows a behavior very similar to the one we would obtain if plotting the BER curves of the digital decoder for different numbers of iterations. With reference to Fig. 9, in a first region, decoder performance improves very rapidly with the decoding time. For instance, the curve corresponding to 100 decoding steps outperforms the first curve in the top (50 decoding steps) with more than 2 dB at $BER = 10^{-2}$. The improvement in performance tends to decrease, and becomes marginal after a certain decoding time (or number of decoding steps). A similar behavior is obtained for the digital decoder, i.e., a larger improvement in performance is obtained in the first iterations, and tends to saturate after a certain number of iterations. These results seem to establish some correspondence between the decoding time of the analog decoder and the number of iterations of the digital decoder. A deeper investigation on the relation between analog decoding time and digital iterations is now under way.

B. Cells Reset

Decoding of each frame requires decoding starting from a proper starting condition. Indeed, restoring the network to a uniform starting condition has been proved to improve and speed up decoding [7], [8]. Due to a design flaw in the control logic of type-E cells, the reset signal used to force the decoder network to a uniform initial condition does not work properly. As a result, decoding operation of each individual frame starts from an unbalanced initial state of the network due to the decoding of the precedent frame. We can, therefore, conjecture that our decoder will suffer from a performance degradation due to this flaw. The effect of the reset on circuit performance can be estimated using the discrete-time model described above. In Fig. 10, we plot the BER performance of the analog decoder without applying the reset to Type-E cells. Comparing Fig. 10 with Fig. 9, we note that the absence of the reset causes a degradation of the BER, which decreases with the decoding time. For instance, the curve corresponding to 200 decoding steps (1.3 μs) in Fig. 10 is 0.7 dB worse at $BER = 10^{-3}$ w.r.t. the same curve when reset is applied. However, as decoding time increases, the degradation of performance due to the absence of reset becomes marginal. In other words, as expected, the absence of the reset seems to only cause an increase of the time that the network needs to settle, but does not translate into a BER degradation at the nominal testing conditions (2 Mb/s) of the chip.

![Fig. 9. BER performance of the analog decoder. (△) Digital decoder (10 iterations).](image1)

![Fig. 10. BER performance of the analog decoder without reset.](image2)
The impact of the reset on decoding speed is also evaluated in Fig. 11, where the average number of decoding steps (decoding time) required by the analog decoder with and without reset are plotted. For low \( E_b/N_0 \) (1 dB), the absence of reset has no impact on the decoding time, since AWGN is the dominant factor in the decoder behavior. However, as \( E_b/N_0 \) increases, the effect of reset on the decoding time is more evident. For instance, at \( E_b/N_0 = 5 \) dB, the time needed to settle by the decoder without reset is twice that of the decoder if reset is applied. As a consequence, it can be argued that the absence of the reset will cause a performance degradation at high \( E_b/N_0 \). These results were expected, since for high \( E_b/N_0 \), decoder behavior in the first phase (transient) is highly dependent on an unbalanced initial state, increasing the settling time, whereas for low \( E_b/N_0 \), the unbalanced initial state is masked by the noise.

V. MISMATCH EFFECTS ON PERFORMANCE AND TEST RESULTS

The performance of analog decoders is affected by circuit nonidealties. Among them, the transistor mismatch appears to be a primary factor limiting decoder performance. Unfortunately, mismatch analysis for complex decoding networks, such as the one discussed in this paper, is intractable with Monte Carlo simulations at circuit level. This poses a significant challenge in the circuit optimization, since transistor sizing to satisfy device matching and speed requirements is difficult.

In [7], we proposed a simple model to characterize mismatch at single-cell level. The model was embedded into a digital (iterative) decoder to estimate the performance degradation due to transistor mismatch. Simulations at circuit level showed performance degradation within the limits predicted by the model. Nevertheless, the fact of using a pure digital decoder to estimate analog decoder performance may certainly be a cause of imprecision. In this section, we introduce the mismatch model proposed in [7] into the discrete-time simulator previously described. The model consists in lumping the effect of transistor mismatch into a deviation of the cell output currents from their ideal value \( I_z \), as

\[
I_z'[n] = (1 + \varepsilon)I_z
\]

where \( \varepsilon \) is a zero-mean random variable modeling the effect of mismatch. For each building block of the decoding network (Fig. 2), the statistical distribution of \( \varepsilon \) is derived through circuit-level Monte Carlo simulations by varying both MOSFET electrical parameters (threshold voltages, current factors, etc.) and circuit inputs. Therefore, \( \varepsilon \) turns out to be a random variable depending on both the data and the physical circuit parameters. In Table I, we report the variance of variable \( \varepsilon \) for all cells. Finally, introducing (4) in (3), the actual expression for the cell output current \( I_z' \) becomes

\[
I_z'[n] = \alpha I_z[n](1 + \varepsilon) + (1 - \alpha)I_z'[n - 1].
\]

The proposed model represents a good tradeoff between accuracy and computational complexity. The model is a further simplification of the model proposed in [12].

With the goal of obtaining an indication of the confidence of the proposed model, we resorted to circuit-level simulations of the component SISO decoder in the presence of mismatch, and compared the results with the discrete-time simulations including mismatch. In particular, we generated 10 different networks at circuit level, and simulated a total of 1146 frames for each one. Minimum and maximum values of \( 4.6 \times 10^{-4} \) and \( 7.4 \times 10^{-4} \) for the BER, and \( 9.8 \times 10^{-3} \) and \( 1.7 \times 10^{-2} \) for the FER, were obtained with a decoding time of 100 \( \mu \)s. Similar results were obtained with the discrete-time model including mismatch; we generated 500 different decoding networks with the discrete-time model, with minimum and maximum values \( 4.8 \times 10^{-4} \) and \( 8.94 \times 10^{-4} \) for the BER, and \( 9.64 \times 10^{-3} \) and \( 2.09 \times 10^{-2} \) for the FER. Both the circuit-level simulations and the discrete-time model showed identical performance with the absence of mismatch: BER = \( 4.81 \times 10^{-4} \) and BER = \( 8.72 \times 10^{-4} \). Although not conclusive, due to the limited Monte Carlo simulation at circuit level, these results give some optimism on the reliability of the proposed model. We also compared the circuit-level simulations with the discrete-time simulations assuming a decoding time of 10 \( \mu \)s, and observed a very good correspondence between the model and the discrete-time simulation.

The BER performance of the chip is reported in Fig. 12 (triangles), together with the BER curve of the digital decoder with 20 iterations. The measured curve corresponds to a data rate of 200 kb/s, i.e., one-tenth of the maximum data rate supported by the chip (2 Mb/s). The reason is that at this data rate, the influence of the reset flaw is minimized. The experimental results show a loss of about 0.5 dB with respect to the benchmark at BER = \( 10^{-5} \), which is kept almost constant for all \( E_b/N_0 \) (0.6 dB at BER = \( 10^{-6} \)). Therefore, no error floor is observed for the analog curve up to BER = \( 10^{-7} \). The chip was also tested

![Graph showing decoding time vs. Eb/N0](image-url)
at nominal conditions. In this case, while until $\text{BER} = 10^{-3}$, the curve is identical to the one reported in Fig. 12, the loss with respect to the benchmark slowly increases to 0.7 dB at $\text{BER} = 10^{-5}$ and to 1 dB at a $\text{BER} = 1.5 \times 10^{-6}$. The increase in performance loss at high $E_b/N_0$ can be explained through the absence of the E-cells reset, as previously discussed.

Several phenomena may contribute to cause deviations from the ideal behavior, like device mismatch, operation in moderate or strong inversion, incomplete transient settling, and electronic noise. We also plot in Fig. 12 performance of the analog decoder estimated using the discrete-time model previously described, including mismatch. In the simulation, the parameters $\varepsilon$ of all cells were considered as independent random variables with the variance of Table I, truncated to the range $[-1, 1]$. 500 different decoding networks were generated. The BER curves corresponding to the 5% and the 95% percentiles of these decoding networks are shown in Fig. 12. A decoding time of 65 $\mu$s is considered. The high-abstraction-level simulations, including mismatch, predict a loss between 0.3 and 0.6 dB with respect to the digital curve at $E_b/N_0 = 5$ dB. At least in this interval, mismatch seems to account for most of the deviation of the tested curve from the benchmark. Moreover, performance loss seems not to drastically degrade as $E_b/N_0$ increases, but remains almost constant, which suggests the absence of an error floor due to mismatch. Thus, mismatch seems to act as an additive noise term.

The actual scalability of analog decoders to block length of greater interest (at least on the order of a few hundred bits) is still an open question. Silicon area is probably the main issue, since a linear extrapolation of our decoder core to the maximum UMTS block length (5114 bits) would yield 524 mm$^2$, which is quite unviable. Technology scaling helps both through the reduction of transistor size and through the availability of a larger number of metal levels for the interconnections. As an example, consider scaling our decoder to a 0.18 $\mu$m CMOS technology. Transistor area can be scaled by a factor of $S^2$, with $S = 0.35/0.18$ 2, with basically no impact on transistor mismatch. In fact, the standard deviation of transistor parameters mismatch scales to a first order as the inverse of the square root of the transistor gate area, but, moving from the 0.35 $\mu$m to the 0.18 $\mu$m technology that have been considered in practice, the mismatch coefficients of transistor of the same area improve by at least a factor of two. We cannot expect the same trend in matching improvement moving to more scaled technologies, but, even with an 0.18 $\mu$m technology, the decoder core size would scale to about 1 mm$^2$; a decoder for a block length of 640 could be implemented in a 16 mm$^2$ area. It should be noted that the extrapolations given in [15] are even more optimistic. Different solutions to implement a fully programmable interleaver have been proposed in [16]. It is worth noting that these solutions can be significantly simplified, considering that, for most communication systems requiring codes with different block lengths, like UMTS, only a single interleaver per block length is actually required. The dependence of the analog decoder settling time, and thus of power consumption, on the block length is still under study, due to the huge computational effort required to predict the transient behavior of the circuit for decoders larger than the one presented here.

VI. CONCLUSIONS

We presented an all-analog implementation of the rate-1/3 UMTS turbo decoder defined in the 3GPP standard, with a block length of 40 bits. The turbo decoder chip was fabricated in a 0.35 $\mu$m CMOS technology with a single power supply of 3.3 V. The chip area is 9 mm$^2$ (4.5 $\times$ 2) excluding I/O pads (4.1 mm$^2$ for the decoder core), with approximately 66,000 transistors (26,000 for the decoder core). We tested the chip at the nominal conditions (2 Mb/s, $I_B = 1$ $\mu$A), with an overall power consumption of 10.3 mW at 3.3 V (6.8 mW for the decoder core alone), which corresponds to an energy per decoded bit of 11.2 nJ for the decoder core.

We also presented a discrete-time model of analog decoding networks that allows running very fast simulations and predict complex chips performance in a very short time, while taking into account circuit transient behavior and device mismatch. The model has been validated through circuit-level simulations of the UMTS analog turbo decoder. Although we are still improving the model, we believe it may result in a very useful tool to predict chip performance, and give circuit-optimization guidelines for those complex decoders for which it is not possible to run circuit-level simulations. Simulation results based on this model of the turbo decoder discussed here, show performance very close to the performance of the digital decoder, giving some optimism on the feasibility of the analog approach for powerful (complex) codes.

The decoder was successfully tested at the maximum channel data rate defined in the 3GPP standard. The performance loss with respect to the digital curve was 0.5 dB at $\text{BER} = 10^{-3}$, slightly increasing to 0.7 dB at $\text{BER} = 10^{-5}$, due to a flaw in the circuit reset. When the data rate is decreased to 200 kb/s (the influence of the reset flaw is minimized), the performance loss is almost constant for all $E_b/N_0$ (0.6 dB at $\text{BER} = 10^{-6}$).

The proposed discrete-time model estimated a performance close to the tested curve. According to the model, mismatch seems to account for most of the deviation from the ideal curve, whereas the increase in performance loss for high $E_b/N_0$ seems to be due to the absence of the E-cells reset.
REFERENCES


Dr. Graeff i Amat received the European Union Socrates Scholarship, the European Union Leonardo da Vinci Scholarship, and the Spanish Ministry of Education and Science Juan de la Cierva Fellowship. During 2001–2004, he was also granted a Doctoral Fellowship by the Italian Ministry of Education, Universities and Research, and by STMicroelectronics.

Sergio Benedetto (M’76–SM’90–F’97) has been a Full Professor of Digital Communications with the Politecnico di Torino, Torino, Italy, since 1981. He has been a Visiting Professor at the University of California, Los Angeles (UCLA), at the University of Canterbury, New Zealand, and is an Adjunct Professor with the Ecole Nationale Superieure de Telecommunications, Paris, France. He has coauthored two books on probability and signal theory (in Italian), the books Digital Transmission Theory (Englewood Cliffs, NJ: Prentice-Hall, 1987), Optical Fiber Communications (Norwood, MA: Artech House, 1996), and Principles of Digital Communications with Wireless Applications (New York: Plenum-Kluwer, 1999), and over 250 papers in leading journals and conferences. He has taught several continuing education courses on the subject of channel coding for the UCLA Extension Program and for the CEI organization.

Guido Montorsi (M’95–SM’03) was born in Torino, Italy, on January 1, 1965. He received the Laurea in Ingegneria Elettronica in 1990 from the Politecnico di Torino, Torino, Italy, with a master thesis concerning the study and design of coding schemes for HDTV, developed at the RAI Research Center, Torino. In 1994, he received the Ph.D. degree in telecommunications from the Dipartimento di Elettronica of the Politecnico di Torino.

In 1992, he was a Visiting Scholar in the Department of Electrical Engineering, Rensselaer Polytechnic Institute, Troy, NY. In December 1997, he became an Assistant Professor with the Politecnico di Torino. From July 2001 to July 2002, he spent one year with Sequoia Communications, developing a baseband algorithm for 3G wireless receivers. In 2003, he became an Associate Professor with the Politecnico di Torino. His research interests are in the area of channel coding, and particularly on the analysis and design of concatenated coding schemes and study of iterative receiver strategies. He is the author of more than 100 papers in international journal and conferences.

Daniele Vogrig (M’02) was born in Cividale del Friuli (Udine), Italy, in 1976. He received the Laurea degree in electronics engineering and the Ph.D. degree in electronic and telecommunication engineering in 2001 and 2005, respectively, both from the University of Padova, Padova, Italy.

He currently holds an Assistant Professor position with the University of Padova. He is mainly interested in the design of integrated circuits for analog signal processing. In particular, his work focused on the analog implementation of iterative error-control decoders. He contributed to the design and testing of two prototypes of analog turbo decoders in pure CMOS technology.
Andrea Neviani (M’06) received the “Laurea” degree (cum laude) in physics from the University of Modena, Modena, Italy in 1989, and the Ph.D. degree in electronics and telecommunication engineering from the University of Padova, Padova, Italy, in 1994.

From 1994 to 1998, he was a Research Associate at the University of Padova, where, from November 1998, he has been an Associate Professor. From November 1998 to November 1999, he was a Visiting Engineer at Rutherford Appleton Laboratory, Oxfordshire, U.K. Previously, he worked on numerical simulation, modeling, and characterization of compound semiconductor devices for high-frequency applications, and silicon devices from BiCMOS processes, with emphasis on effects induced by hot carriers. In the recent past, he has contributed to the study of methods for the statistical simulation of VLSI circuits. At present, his main interest is the design of analog and mixed-signal CMOS circuits for analog signal processing, for low-power implantable biomedical devices, and for RF ultra-wideband receivers. He has co-authored about 70 journal articles and conference papers.

Andrea Gerosa (M’96) received the “Laurea” (equivalent to M.S.) and Ph.D. degrees in electronic and telecommunication engineering in 1995 and 1999, respectively, both from the University of Padova, Padova, Italy.

He is currently an Assistant Professor with the University of Padova. From August 1997 to July 1998, he was an exchange Visiting Graduate Student at the University of California, Berkeley.

He is mainly interested in analog IC design for communication applications and recently he was involved in the design of analog decoders, A/D converters and RF front-ends for UWB receivers. He has published more than 35 papers in international journal or conference proceedings.