



**HAL**  
open science

# Analog decoder performance degradation due to BJT's parasitic elements

Nicolas Duchaux, Cyril Lahuec, Matthieu Arzel, Fabrice Seguin

► **To cite this version:**

Nicolas Duchaux, Cyril Lahuec, Matthieu Arzel, Fabrice Seguin. Analog decoder performance degradation due to BJT's parasitic elements. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2009, 56 (11), pp.2402 - 2410. hal-01801572

**HAL Id: hal-01801572**

**<https://hal.science/hal-01801572>**

Submitted on 28 May 2018

**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.

# Analog Decoder Performance Degradation Due to BJT's Parasitic Elements

Nicolas Duchaux, Cyril Lahuec, Matthieu Arzel, Fabrice Seguin  
 Département Electronique  
 TELECOM Bretagne  
 Brest, Brittany, France  
 nicolas.duchaux@telecom-bretagne.eu

**Abstract**—This paper presents the effect of BJT's parasitic elements on the decoding performance of a BiCMOS analog decoder. The transistor's parasitic effects are taken into account to develop a more accurate behavioral model of the computing nodes. The model is applied to double-binary 0.25- $\mu\text{m}$  BiCMOS analog decoders. Behavioral simulations show that the BJT's parasitic elements deteriorate the error correcting performance of a stand-alone APP decoder by 0.5dB compared with the ideal BER. In a turbo scheme, the loss is reduced to 0.2dB for BER smaller than  $10^{-2}$ . A simple solution based on an NMOS amplifier is proposed to counterbalance the dominant parasitic element. The amplifier reduces the degradation by 0.2dB for the APP decoder. Similar improvements are observed in the turbo scheme for BER greater than  $10^{-2}$ .

**Index Terms**—Analog decoding, BiCMOS, parasitic elements, behavioral modeling

## I. INTRODUCTION

The usual architecture of a digital receiver employs a forward-error correcting device to overcome transmission errors due to a noisy communication channel. In the late nineties two research teams, led by Professor Hagenauer and Professor Loeliger showed that the digital decoder could be advantageously replaced by an analog counterpart [1] - [3]. Probability-propagation algorithms such as the Belief Propagation or Turbo ones are directly mapped onto silicon thanks to the non-linear characteristics of transistors. The resulting analog network of computing nodes converges to a steady state to finally correct errors as well as digital counterparts, but with the major advantages of lower power consumption, higher speed and/or smaller on-chip area as shown by the first chips [4] [5]. Since then, several other research teams have developed proof-of-concept circuits to validate new architectures, design methodology, automatic synthesis and even built-in self test [6] - [8]. Nevertheless, many issues are still to be solved in order to challenge digital decoders. As for any other system, before designing the decoder at transistor level, developing a behavioral model is mandatory first to validate the architecture, as in [2] [9] [10] and second to estimate the decoder's Bit and Frame Error Rates (BER/FER). For the latter purpose, transistor level simulations are too time consuming to obtain BER and FER when dealing with relatively large decoders. The behavioral models presented in previous works, such as [4] and [11], consider ideal computing nodes and only implement the decoding algorithm. For very

small decoders there is, in general, a good agreement between behavioral and actual measurements from fabricated circuits, such as [4]. When considering complex decoders, there is a relatively large discrepancy, from a few hundredths [12] up to a couple of dBs [13], between the two. As most of the published work to date concerns the design of CMOS analog decoders [14] - [17], the performance degradations in terms of BER brought in by MOS transistors imperfections such as mismatches have been studied, see for instance [17]. Fewer works have been published on BJT-based analog decoders [4] [12] [18]. Performance degradation of such decoders due to transistor mismatch was studied in [18]. The authors concluded that assuming random mismatch, the errors introduced do not impair the decoder's performance but merely imply a slower convergence [19]. However, no comprehensive study on the effects of non random BJT's parasitic elements over decoding performance exists to date. These effects are not random as they affect every single BJT in the decoder and hence could impair its performance. Some insights on this matter were given in [20] [21] but the studies were rather incomplete. Moreover, although the cause of degradation was identified in the previous works, no attempt has been made to address these additional errors to improve the overall decoding performance of the decoders.

This paper aims thus at providing an in-depth study of performance degradation of BJT-based analog decoders due to transistors' non-idealities. A simple solution counteracting the BJT parasitic elements is proposed and simulations are run for a stand-alone A Posteriori Probability (APP) decoder and a turbo decoder. The results are shown taking the case study of a 0.25- $\mu\text{m}$  BiCMOS analog tail-biting double-binary APP decoder [10].

The paper is organized as follows. Section II presents the targeted code and the decoder's architecture. Section III is about the different BJT's parasitic effects to be taken into account and their corresponding behavioral models. Section IV presents the computing node behavioral model. Then, the decoding performance of a stand-alone APP decoder and a turbo-decoder are assessed. In Section V, a circuit is proposed to counterbalance the deterioration of the decoder's performance. Finally, Section VI concludes the paper.

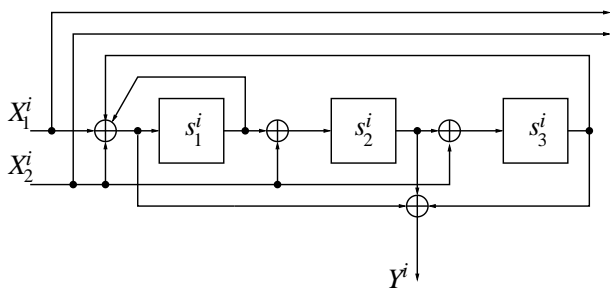


Fig. 1. Double-binary encoder, code rate 2/3.

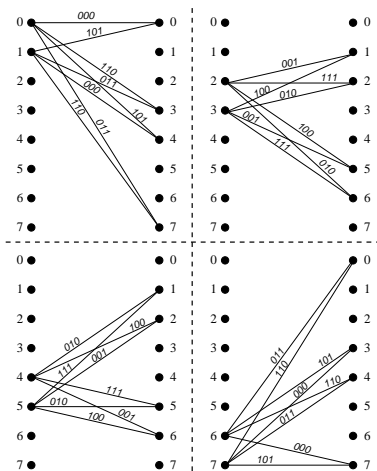


Fig. 2. Trellis associated with the encoder of Fig. 1 (split in four sub-parts for clarity). Branches are labeled with the encoded symbol  $X_1X_2Y$ .

## II. APP DECODER ARCHITECTURE

### A. Target code

The background of the study concerns a double-binary 8-state tail-biting Recursive Systematic Convolutional code. Double-binary simply means that the decoder processes 2-bit symbols. The convolutional encoder has a code rate of 2/3 and produces a single parity bit  $Y$  per double-binary input symbol  $X_1X_2$  as shown in Fig.1. A trellis section of this code is illustrated in Fig. 2. Compared to a trellis section of a code using one-bit symbols, the number of branches is doubled but the length is halved.

The motivation behind choosing such a code is that it has major advantages compared with its simple binary counterparts. Berrou et al. [22] showed that the convergence of m-ary turbo codes is better but the gain is less noticeable for  $m > 2$ . Their minimal distances and their asymptotic gains are larger. They are also less sensitive to puncturing. Already used as the code for the DVB-RCS standard [23], these advantages also make double-binary turbo codes key candidates for many other telecommunication standards.

### B. Decoder structure

The APP decoder of the tail-biting code is implemented as a ring, see Fig. 3. The decoder is designed to deal with a frame length of 24 double-binary symbols meaning that the decoder

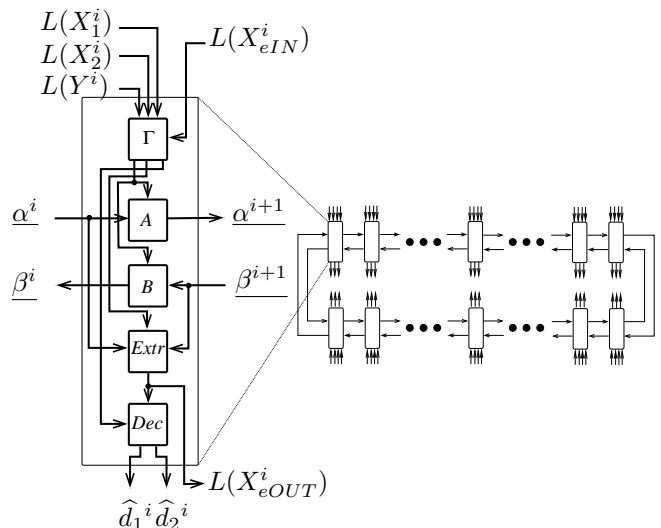


Fig. 3. Tail-biting APP decoder structure and decoding section.

processes 72-bit long input blocks, 48 information bits and 24 redundancy bits. There are as many decoding sections as symbols to decode, i.e. 24 in the present case. As in [5], each section is built from four modules: a  $\Gamma$  module to compute the branch metrics, an  $A$  module for the forward state metrics, a  $B$  module for the backward state metrics and a  $Dec$  module to decide on the value of the double-binary symbol. There are two sets of inputs to the section. The first set is the data generated by the channel,  $L(X_1^i) L(X_2^i) L(Y_i)$ , which are associated with the  $i^{th}$  symbol  $X_1^i X_2^i$  and its parity bit  $Y_i$ . The second set of inputs is composed of the forward and backward state metrics  $i$  and  $i + 1$  produced by the adjacent trellis sections. The outputs are the metrics  $i + 1$  and  $i$ , fed to the adjacent sections, and the decisions  $\hat{d}_1^i \hat{d}_2^i$  for the transmitted symbol  $X_1^i X_2^i$ . A fifth module is required if the tail-biting APP decoder is part of a turbo decoder: the  $Extr$  module. This module computes the extrinsic information  $L(X_{eOUT}^i)$  which is then used by the module of the second tail-biting APP decoder as  $L(X_{eIN}^i)$ . All the above modules — except the  $Dec$  one — are basically sum-product modules implementing the decoding algorithm. The design of the required computing nodes is described next.

### C. Sum-product nodes

BJT-based Gilbert cells [24] are used as probability multipliers. Considering that the input voltages are proportional to Log-Likelihood Ratios (LLR), the output currents are proportional to the products of probabilities. Summing probabilities represented by currents is then simply done by connecting wires. This simple sum-product circuit was demonstrated in [25] for any field of probability and has been used in a number of analog decoders [19] [26]. In the present case for instance, the  $A$  module of the decoding section requires the design of an extended Gilbert structure with 8-ary inputs, as illustrated in Fig. 4. It uses 72 transistors. The lower stage is an emitter-coupled set of bipolar transistors whose bases are connected to  $n_X$  voltages  $V_i^X$  that are proportional to the log-likelihoods of the data  $X$ . The upper

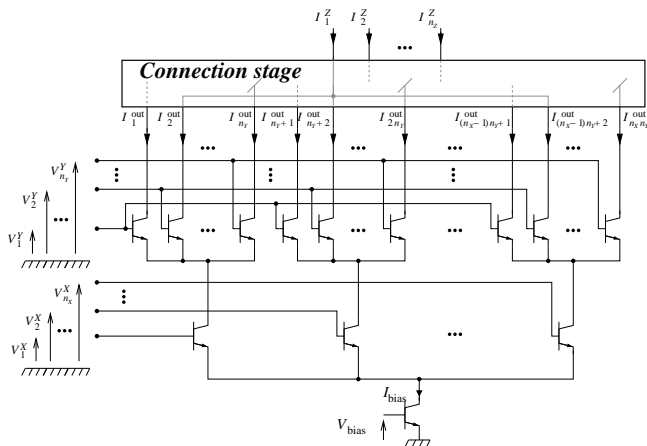


Fig. 4. Extended Gilbert cell.

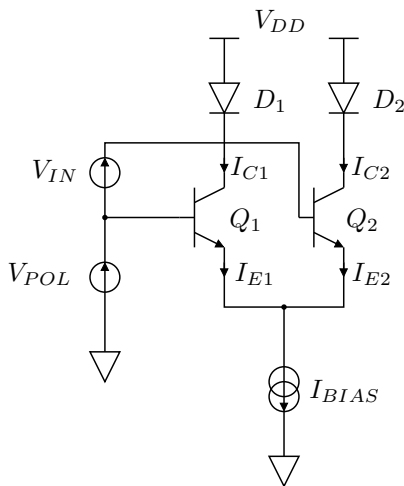


Fig. 5. Emitter-coupled pair.

stage is made up of  $n_X$  identical emitter-coupled bipolar sets, each of them being connected to a different collector on the lower level. The bases of these bipolar sets are connected to  $n_Y$  voltages  $V_j^Y$  proportional to the log-likelihoods of the data  $Y$ . Thus, the outputs are  $n_X \times n_Y$  currents  $I_k^{out}$  given by:

$$I_{(i-1) \times n_Y + j}^{out} = p_X(x_i) p_Y(y_j) I_{BIAS} \quad (1)$$

All the pair-wise products  $p_X(x_i) p_Y(y_j)$  are thus available for further computations. For instance, in Fig. 4, the currents are summed using simple nodes to produce the output currents  $I_k^Z$ .

Equation (1) is correct considering the collector current to be solely an exponential function of the base-emitter voltage. The next section shows how the BJT's parasitic elements impair the conversion of LLRs into probabilities.

#### D. Error converting LLRs into probabilities

The basic circuit in Fig. 5 converts the input LLR, represented by  $V_{IN}$ , into probabilities, represented by the collector

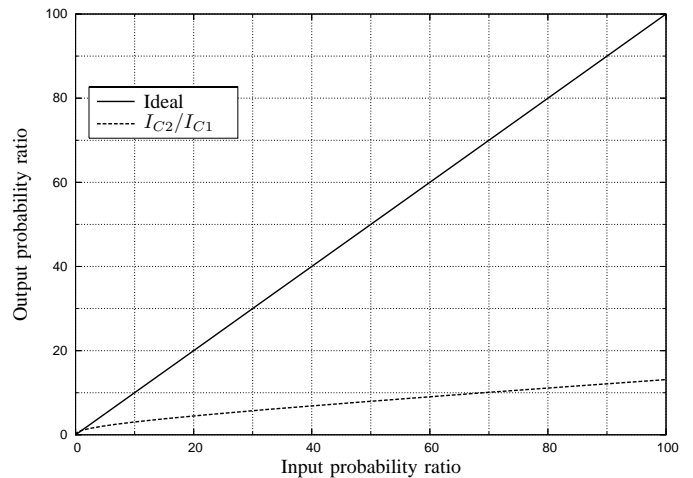


Fig. 6. Probability ratio transfer function, i.e.  $I_{C2}/I_{C1}$  vs  $\exp(\frac{V_{IN}}{V_T})$  of a minimal size emitter-coupled bipolar pair: ideal and simulated (transistor level),  $I_{BIAS} = 250\mu A$ ,  $V_{DD} = 2.8V$ .

currents  $I_{C1}$  and  $I_{C2}$ . Designed with Cadence® design tools for NXP's QUBIC4 0.25- $\mu m$  BiCMOS process, the circuit uses minimal size transistors, a 2.8V supply and a 250 $\mu A$  biasing current. A transistor level simulation is run with Spectre® simulator to assess the accuracy of the LLR into probability conversion. This simulation takes into account all the parasitic effects present in the transistor electrical compact model (Mextram 504) [27] provided by NXP. Fig. 6 represents the output probability ratio versus the input probability ratio, i.e. the ratio of the two collector currents  $I_{C2}$  and  $I_{C1}$  versus  $\exp(\frac{V_{IN}}{V_T})$ . Ideally, the curve obtained should be a straight line with a slope of 1. It is far from being the case as the conversion error is as high as 85 percent for an input LLR of 100. This simple simulation shows that the assumption of having an ideal exponential relationship between the collector current and the base-emitter voltage is absolutely incorrect. Two questions arise from this, where does this come from and is this really a problem for the decoder?

Answering these requires a careful study of the transistor electrical model provided by NXP. From this analysis the relevant parasitic phenomena are extracted to build an accurate behavioral model of the transistor.

### III. EFFECTS TO BE TAKEN INTO ACCOUNT

The transistor behavioral model has to be accurate and yet simple enough to simulate the decoder in a reasonable amount of time. Finding which parasitic effects have to be taken into account is somehow lengthy but not complicated. It is assumed that the transistor is in the forward active region, which reduces the number of physical effects to test for relevance. Each effect described by one or several equations in the Mextram model [27] has been simulated separately to see its impact on the collector current. Among all of them, only three have to be added to the Ebers-Moll model to correctly describe the transistor's collector current. These effects are the parasitic emitter resistor, the reverse Early and the Webster effects. First, a brief description of these

effects is given. Second, a comparison is made between the collector currents obtained from spice-level simulations using the Mextram model and from three behavioral models, each described with an equation, taking the different effects into account.

#### A. Ebers-Moll model

This model is the one usually assumed to simulate the computing nodes of analog decoders designed using bipolar transistors as in [4]. This is also the model used to obtain Equation (1). The Ebers-Moll model takes into account only the exponential I-V characteristic of the transistor. For the region of operation of the transistor and considering the Ebers-Moll model, the collector current  $I_C^{*em}$  is:

$$I_C^{*em} = I_S \exp\left(\frac{V_{BE}}{V_T}\right) \quad (2)$$

where  $I_S$  is the saturated current of the of the base-emitter junction,  $V_T$  the thermal voltage and  $V_{BE}$  the base-emitter voltage. Despite the fact that this model is very convenient, it departs from reality too much as Fig. 6 shows. The effects to be added to obtain a collector current equation closer to the transistor's Mextram model are described next.

#### B. Parasitic emitter resistor

The most obvious effect to be added to the Ebers-Moll model is the emitter resistor. This resistor may not be negligible when small size transistors are used and large biasing currents are necessary. Considering that  $I_C^{*R} \approx I_E$ , the collector current  $I_C^{*R}$  is given by:

$$I_C^{*R} = I_S \exp\left(\frac{V_{BE}}{V_T}\right) \exp\left(-\frac{R_E I_C^{*R}}{V_T}\right) \quad (3)$$

where  $R_E$  is the parasitic emitter resistor. Its value is about  $220\Omega$  for the process used and the minimal size transistors.

#### C. Forward and reverse Early effects

The Early effects take into account the finite output resistor of the BJT and the base-width modulation due to the biasing of the junctions. Base-width modulation reduces the collector current and affects more small size devices than large ones. The first effect is the well-known forward Early effect, characterized by the Early voltage  $V_{AF}$ . The second effect is known as the reverse Early effect. It is also characterized by a voltage, denoted  $V_{AR}$ . The Early effects term  $q_1$  is described in [28] as:

$$q_1 = 1 + \underbrace{\frac{V_{BE} - R_E I_C}{V_{AR}}}_{\leq 0.5} + \underbrace{\frac{V_{BC} + R_C I_C}{|V_{AF}|}}_{\leq 0.05} \quad (4)$$

The values of  $V_{AF}$  and  $V_{AR}$  are, respectively,  $-33V$  and  $2V$  in the process used and for minimal size BJT. The forward Early effect is ten times smaller than the reverse effect. Hence  $q_1$  can be approximated to:

$$q_1 \approx 1 + \frac{V_{BE} - R_E I_C}{V_{AR}} \quad (5)$$

This approximation yields a modeling error of four percent at most. Let  $I_C^{*RE}$  be the collector current taking the emitter resistor and the reverse Early effect into account:

$$I_C^{*RE} = I_S \frac{\exp\left(\frac{V_{BE}}{V_T}\right) \exp\left(-\frac{R_E I_C^{*RE}}{V_T}\right)}{\left(1 + \frac{V_{BE} - R_E I_C^{*RE}}{V_{AR}}\right)} \quad (6)$$

#### D. Webster effect

The last effect which has to be taken into account is the Webster effect [29]. This effect is not well-known and hence is often discarded when analyzing BJT circuits. Basically, it accounts for the increase of conductivity in the base. When higher current is injected into the base, the recombination rate increases hence lowering the emitter efficiency. The Webster effect term is used as defined in [28]. Let  $I_C^{*REW}$  be the collector current where the emitter resistor, the reverse Early effect and the Webster effect are taken into account:

$$I_C^{*REW} = \frac{I_F}{\left(1 + \frac{V_{BE} - R_E I_C^{*REW}}{V_{AR}}\right) \left(\frac{1 + \sqrt{1 + 4 \frac{I_F}{I_K}}}{2}\right)} \quad (7)$$

where  $I_K$  is the knee current which is a model parameter due to the Webster effect.  $I_K$  is equal to  $1.97\text{mA}$  for the process used.  $I_F$  is the ideal forward current of the transistor described in [27]:

$$I_F = I_S \exp\left(\frac{V_{BE}}{V_T}\right) \exp\left(-\frac{R_E I_C^{*REW}}{V_T}\right) \quad (8)$$

#### E. Comparison of the different models

In this section, the accuracy of the different behavioral models defined in the above subsections is assessed. Let  $I_C$  be the collector current simulated by Spectre® using the Mextram model provided by NXP. Let  $I_C^*$  be the collector current obtained from the behavioral model sought. Then, plotting the ratios  $I_C/I_C^*$  versus  $V_{BE}$ , where  $I_C^*$  is successively defined by Equation (2) to Equation (7), and comparing them to a ratio of 1 shows how good the models are with respect to the Mextram model. The closer to 1 the ratio is, the better the behavioral model is. This is shown in Fig. 7 for  $V_{BE}$  ranging from  $0.5V$  to  $1V$ , which are typical values found in the actual decoder. It is important to have a behavioral model accurate over this full range of  $V_{BE}$  as the lower range corresponds to a low transistor biasing, i.e. low probability, and the higher range corresponds to a high biasing, i.e. a high probability.

From Fig. 7 it can be seen that only the model described by Equation (7) is close enough (less than four percent in error) to the Mextram model over the full range of  $V_{BE}$ . Hence, this model is chosen and used in the next section to build the complete decoder's behavioral model.

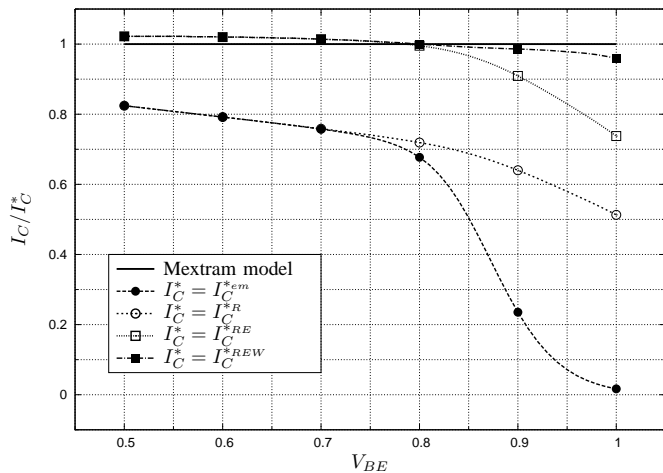


Fig. 7. Ratio of  $I_C$ , the collector current obtained from spice-level simulation, over  $I_C^*$ , the collector current yielded by the various behavioral models, versus  $V_{BE}$  ( $V_{BC} = 0V$ ).

#### IV. DECODERS BEHAVIORAL MODELING

##### A. Behavioral modeling of an emitter-coupled pair

The bipolar pair represented in Fig. 5 is the basic block of the sum-product nodes. It can be considered as a system composed of one input voltage  $V_{IN}$  and two coupled outputs currents  $I_{C1}$  and  $I_{C2}$ . These two currents are modeled using Equation (7) and their ratio is:

$$\frac{I_{C2}^*}{I_{C1}^*} = \frac{(1 + \frac{V_{BE1} - R_E I_{C1}^*}{V_{AR}})(1 + \sqrt{1 + 4 \frac{I_{F1}}{I_K}})}{(1 + \frac{V_{BE2} - R_E I_{C2}^*}{V_{AR}})(1 + \sqrt{1 + 4 \frac{I_{F2}}{I_K}}} e^{(\frac{V_{IN}}{V_T} - \frac{R_E}{V_T} I_\delta)} \quad (9)$$

where

$$I_\delta = I_{C2}^* - I_{C1}^* \quad (10)$$

$V_{BE1}$  is the base-emitter voltage of transistor  $Q_1$ ,  $I_{F1}$  is the forward current of transistor  $Q_1$  (respectively  $V_{BE2}$  and  $I_{F2}$  for transistor  $Q_2$ ).

Equation (9) is not easy to use as  $V_{BE1}$ ,  $V_{BE2}$ ,  $I_{F1}$  and  $I_{F2}$  depend on  $V_{IN}$ . It would be easier to implement the behavioral model if the collector current ratio depended directly on  $V_{IN}$ ,  $I_{C1}$  and  $I_{C2}$ . This can be easily done if some approximations are made. Using the following Taylor expansions when  $x$  is small:

$$\exp(x) \approx 1 + x \quad (11)$$

$$\sqrt{1+x} \approx 1 + 0.5x \quad (12)$$

and doing the first order approximation  $I_F \approx I_C^*$  in the Webster effect terms, then Equation (9) can be simplified to:

$$\frac{I_{C2}^*}{I_{C1}^*} = \exp\left(\frac{V_{IN}}{V_T} - \frac{R_E}{V_T} I_\delta + \frac{R_E}{V_{AR}} I_\delta - \frac{V_{IN}}{V_{AR}} - \frac{I_\delta}{I_K}\right) \quad (13)$$

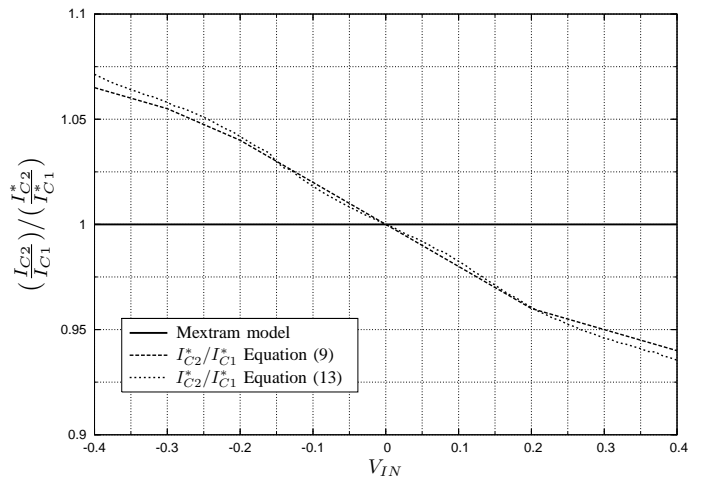


Fig. 8. Ratio  $(\frac{I_{C2}}{I_{C1}})/(\frac{I_{C2}^*}{I_{C1}^*})$  versus  $V_{IN}$  for an emitter-coupled pair of minimal size transistors.  $I_{C2}/I_{C1}$  and  $I_{C2}^*/I_{C1}^*$  are obtained from spice-level simulation and behavioral models respectively.  $I_{BIAS} = 250\mu A$ ,  $V_{DD} = 2.8V$

To show that the simplifications done are acceptable, the ratio  $(\frac{I_{C2}}{I_{C1}})/(\frac{I_{C2}^*}{I_{C1}^*})$  is plotted versus  $V_{IN}$  when  $\frac{I_{C2}^*}{I_{C1}^*}$  is successively defined by Equations (9) and (13).  $I_{C1}$  and  $I_{C2}$  still represent the collector currents obtained from the NXP transistor model. The result is shown in Fig. 8. As in section III.E, the closer to 1 this ratio is the better the modeling. The error of modeling is at most six percent over the full range of  $V_{IN}$  when Equation (9) is used and is at most seven percent when using Equation (13). This validates the final behavioral model represented by Equation (13). It can be noted this behavioral model is based on the NXP QUBIC4 0.25- $\mu m$  process which is a typical process. The values of  $R_E$ ,  $V_{AR}$  and  $I_K$  should be adapted if another process is used but the results should be similar.

##### B. Behavioral simulation of the APP decoder

In this section, the bit error rate curve obtained using two behavioral models of the analog APP decoder defined in section II are presented. The first model considers that the decoder is made of ideal multipliers. The second model takes into account the non-idealities of the BJTs as described in Equation (13). Both models are first-order models implemented using Simulink®. Each module of the decoder in Fig. 3 is described by blocks programmed with C language and the analog exchange of information is made through RC lines. Runge-Kutta ODE solver with variable time-steps is used to compute the solution. The bias currents of the computing nodes are all equal to  $250\mu A$  and the transistors are minimal size ones. The degradations brought in by the parasitic BJT's elements impair the error rate by 0.5dB when compared with the ideal case as shown in Fig. 9.

This degradation is significant enough to be taken into account for a stand-alone APP decoder. It is interesting to study if the parasitic elements impact an analog turbo decoder in the same manner.

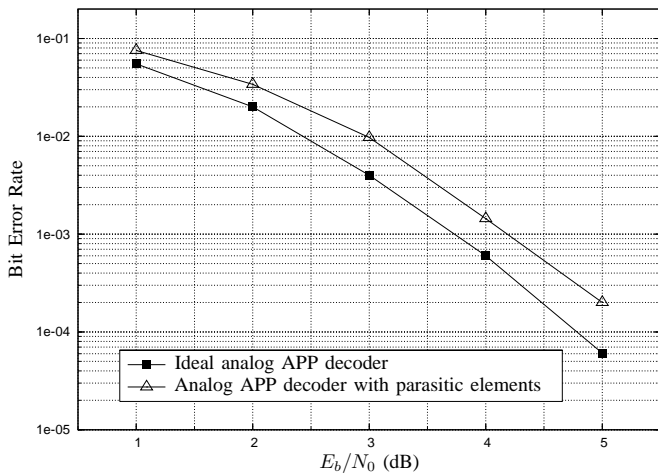


Fig. 9. Behavioral simulation of the analog APP decoder considering the computing cells ideal and affected by BJTs' parasitic elements, frame length 48 information bits,  $I_{BIAS} = 250\mu A$ ,  $V_{DD} = 2.8V$ .

TABLE I  
INTERLEAVING

$i$	0	1	2	3	4	5	6	7	8	9	10	11
$j$	1	8	15	22	5	6	7	14	21	4	11	12
$i$	12	13	14	15	16	17	18	19	20	21	22	23
$j$	13	20	3	10	17	18	19	2	9	16	23	0

### C. Behavioral simulation of a turbo decoder

The behavioral models of the APP decoder presented in the above sections are now used to implement a turbo scheme. A turbo code with the following parameters is assumed: rate of 1/2 with no puncturing. The interleaver used is presented in Table I. The  $i^{th}$  symbol in the natural order corresponds to the  $j^{th}$  symbol in the interleaved order. Moreover a local disorder is introduced, the two bits of one double-binary symbol over two are inverted. The interleaver is modeled by RC lines in Simulink®. The analog turbo decoder is simulated considering the multipliers ideal and then taking into account the BJT's parasitic elements. The BER curves obtained are presented in Fig. 10. For a BER greater than  $10^{-2}$ , the parasitic BJT's elements deteriorate the performance by more than 0.5dB, while for a BER smaller than  $10^{-2}$  the loss is 0.2dB compared with the ideal case. As expected, the turbo structure compensates the errors brought by the two APP decoders at high SNR. Thanks to the uncorrelated data in each decoder, the turbo structure can almost overcome the errors added by the BJTs' parasitic elements when the SNR is larger than 2dB.

### D. Computational cost

The computational cost of the proposed model is compared with the ideal and the Mextram models. A single 24-symbol frame is randomly generated at an SNR of 4dB. This frame is fed to two Simulink® behavioral models of the stand-alone APP decoder, one is ideal (using Equation (2)) and the other takes the parasitic elements into account (using Equation (13)). The time it takes to simulate the decoding of the frame on one

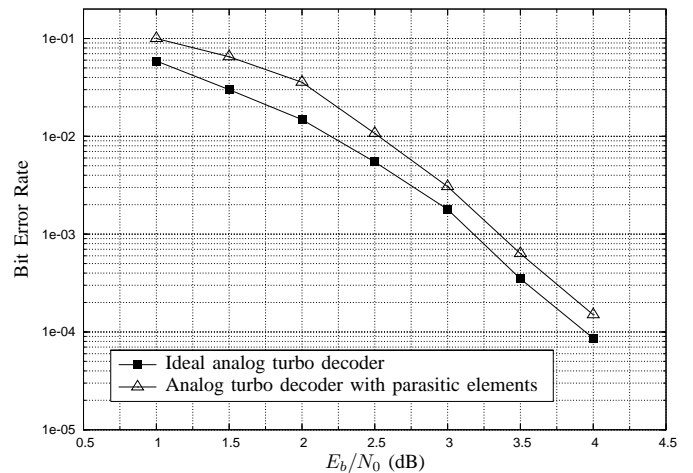


Fig. 10. Behavioral simulation of the analog turbo decoder considering the computing cells ideal and affected by BJTs' parasitic elements, frame length 48 information bits,  $I_{BIAS} = 250\mu A$ ,  $V_{DD} = 2.8V$ .

TABLE II  
NORMALIZED SIMULATION TIME OF THE DECODING OF ONE RANDOM FRAME USING THE DIFFERENT MODELS FOR AN SNR OF 4dB

	Ideal model Equation (2)	Proposed model Equation (13)	Mextram model
Normalized simulation time	1	24	155 000

core of an Intel Xeon 2.66GHz processor is recorded. For the ideal decoder model, the simulation runs for 1.3s while 31s are required when the proposed model is used. Decoding the same frame on the same computer using spice-level simulation (hence with the Mextram model) requires 2.3 days. These results are shown in Table II, normalized to the simulation time of the ideal decoder. Based on that, it is possible to estimate the simulation time required to reach a given BER. For instance, to reach  $10^{-4}$  for 200 bits in error, it takes two weeks with the proposed model while more than 300 years are necessary using spice-level simulations. Although the proposed model is relatively slow, it should provide a better idea on how the actual decoder will perform. The simulation time could be reduced using the importance sampling approach [30] [31]. However, this work concentrates on the model accuracy and not on the simulation time, so importance sampling was not considered.

## V. COUNTERBALANCING BJTs' PARASITIC ELEMENTS

### A. Most relevant effect

The effects described with Equation (13) seem to be quite hard to counterbalance. The parasitic emitter resistor could be counterbalanced by modifying either the size of the BJTs or the bias current. The value of  $R_E$  is directly proportional to the area of the emitter. Considering that the bias current remains the same, choosing a large BJT will reduce the size of  $R_E$ . Whereas this is valid technically, it is not economically as the resulting circuit would be too expensive to produce due to its large size. Then, one can consider keeping minimal size BJT, for the aforementioned reason, and lowering the biasing

current. This is not an option either, at least to a certain extent, as it would also reduce the decoding speed. Speed is one of the main reasons for using BJTs rather than subthreshold MOS to design the decoder. Hence, the design of the multiplier cannot be changed and an additional correcting circuit must be designed.

Before undertaking such a task, it is interesting to know if one of the three parasitic effects described is more relevant than the others. By using a simulation trick, the contribution of the parasitic  $R_E$  can be accurately estimated from transistor level simulations. Adding an ideal negative resistor  $R_{EC}$  such as  $R_{EC} = -R_E$  at each emitter of the transistors in the differential pair shown in Fig. 5 cancels out the effect of the actual  $R_E$ . In other words the effective parasitic  $R_E$  is zero. The result is shown in Fig. 11 and is compared with what is obtained when the effective  $R_E = 220\Omega$ . From this figure, the parasitic emitter resistors account for almost 80 percent of the probability transfer function error. The hardest effects to counterbalance are those which are the less relevant, that is to say the reverse Early and Webster effects. Therefore, this work concentrates on counterbalancing the effects of  $R_E$ .

### B. Counterbalancing circuit

Noting that the parasitic resistor lowers the effective transconductance  $g_m$  of the transistor implies a less efficient conversion of voltages into collector currents and hence of the LLRs into probabilities. From a circuit point of view, as the effective  $g_m$  is lowered, it implies a gain loss of the Gilbert multiplier thus modifying its transfer function. This suggests that a gain stage would suffice to counterbalance the inaccurate conversion due to the parasitic  $R_E$ . The gain  $K$  is chosen so that the overall gain of the computation node is approximately one. This gain stage is implemented as a simple differential MOS stage as presented in Fig. 12. Actually, this circuit already exists and was proposed in [4] where it is only used as DC level shifter. The DC level shift is necessary to adjust the input biasing of the next stage. One simply needs to design the differential NMOS stage to provide the necessary DC level shift and also some gain. Thus, there is no added complexity to the circuit. The circuit presented is for two outputs but it can be extended to a larger number of outputs. To assess the efficiency of this simple correcting circuit, a behavioral model of the gain stage is done in Simulink®. The MOS differential stage is modeled as a piecewise linear model with a linear region of gain  $K$  and two saturation regions. This is added to the overall behavioral model of the decoder. The results of the simulations are presented next.

### C. Simulation of the corrected APP decoder

Considering that the biasing current is evenly divided into each branch of the Gilbert multiplier, the larger the multiplier the smaller biasing current of each transistor. It implies that the effect of the parasitic  $R_E$  in each branch is reduced and hence, a smaller correction is required. Thus, the correction gain  $K$  is chosen depending on the size of the multipliers. The larger the multiplier is, the smaller  $K$  is. Fig. 13 shows that the corrected decoder performs better than the uncorrected

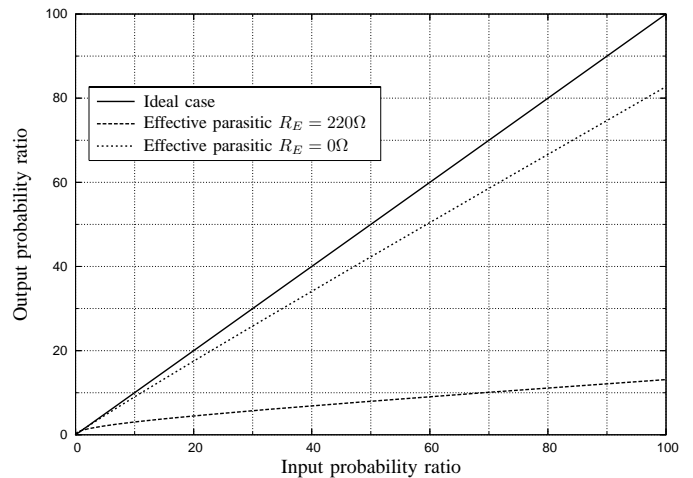


Fig. 11. Probability ratio transfer functions of a minimal size emitter-coupled pair. Solid line: ideal case; dashed line Effective parasitic  $R_E = 0\Omega$ ; dotted line: Effective parasitic  $R_E = 220\Omega$ .  $I_{BIAS} = 250\mu A$ ,  $V_{DD} = 2.8V$  (transistor level simulation).

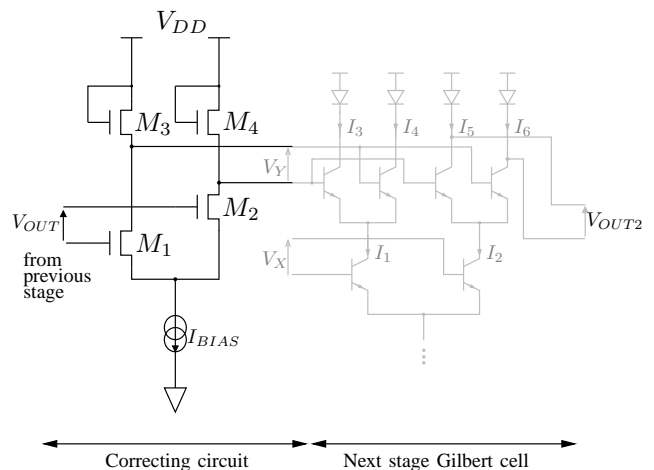


Fig. 12. Correcting and interconnecting circuit.

one. The simple differential gain stage improves the error rate by 0.2dB. However, to match the ideal case, this is not enough but the effect of the parasitic resistors is only corrected to a first order. Again, considering the simplicity of the correction circuit, the improvement is nevertheless significant. In the next section are presented the simulation results of the turbo decoder.

### D. Simulation of the corrected turbo decoder

The APP decoder implementing the correction circuit is now put in a turbo scheme. As shown in Fig. 14, the correction is very efficient for small SNR values, i.e. between 1dB and 2dB, as the corrected BER curve gets very close to the ideal one. However, above 2dB the correction circuit does not improve the decoding performance which remains 0.2dB away from the ideal case. Thus, the correction circuit shows its interest in the APP decoder and for the turbo decoder in the low SNR range.

It is worth noting that if the parasitic emitter resistor were fully compensated (effective  $R_E = 0\Omega$ ), the analog turbo



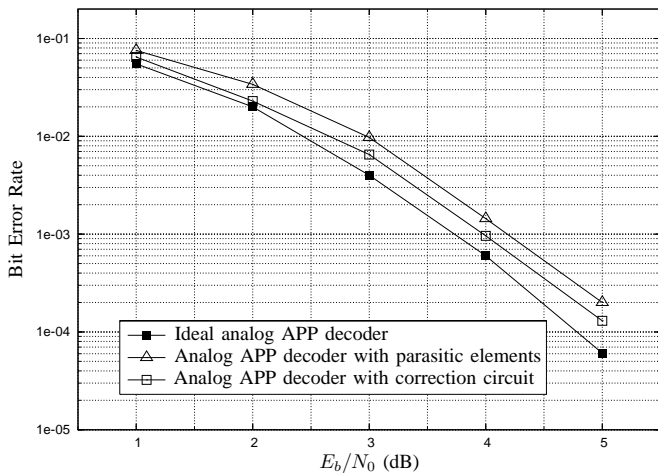


Fig. 13. Behavioral simulation of the analog APP decoder considering the computing cells ideal, affected by BJTs' parasitic elements and counterbalanced, frame length 48 information bits,  $I_{BIAS} = 250\mu A$ ,  $V_{DD} = 2.8V$ .

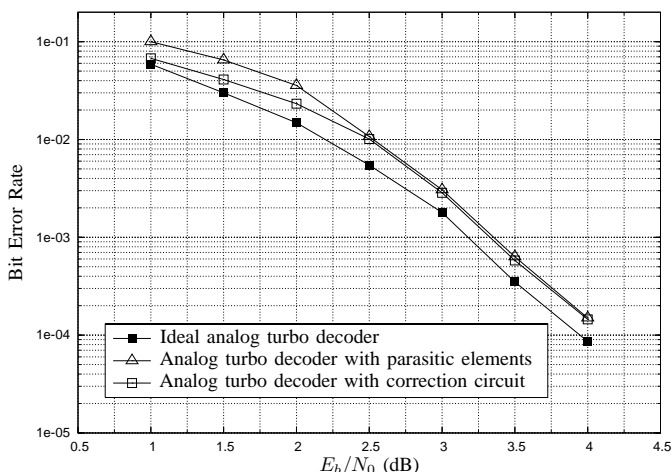


Fig. 14. Behavioral simulation of the analog turbo decoder considering the computing cells ideal, affected by BJTs' parasitic elements and counterbalanced, frame length 48 information bits,  $I_{BIAS} = 250\mu A$ ,  $V_{DD} = 2.8V$ .

decoder would perform as good as the ideal decoder and its digital counterpart (8 iterations, floating point), see Fig. 15. This means that the Early and Webster effects alone do not impact the performance of the analog turbo decoder even though they account for about 20 percent of the LLR to probability conversion as shown in Section V.A. Hence, these secondary effects need not to be taken into account if the decoder is used in a turbo scheme.

## VI. CONCLUSION

Relevant BJTs' parasitic elements have been identified and characterized. A high level model has been developed and it shows that BJTs' parasitic elements deteriorate the decoding performance of the APP decoder by 0.5dB for a bias current of  $250\mu A$  and minimal size transistors. When the APP decoder is used in a turbo scheme, the loss is down to 0.2dB for a BER smaller than  $10^{-2}$ . A simple circuit necessary for interconnecting decoding stages can also be used

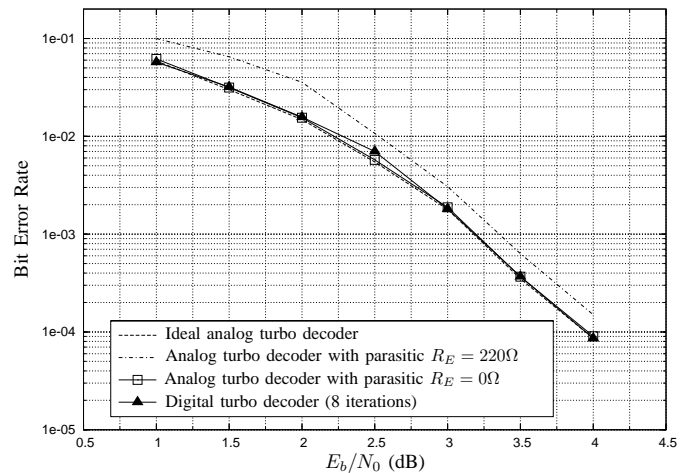


Fig. 15. Behavioral simulation of the analog turbo decoder considering the computing cells perfect, affected by BJTs' parasitic elements, with BJTs emitter resistor fully compensated compared to a digital turbo decoder with 8 iterations, frame length 48 information bits,  $I_{BIAS} = 250\mu A$ ,  $V_{DD} = 2.8V$ .

to counterbalance the BJT parasitic  $R_E$ . Previously used as a simple DC level shifter, it can implement a compensating gain too. Thus, there is no added complexity to the decoder while improving the performance by 0.2dB compared with the uncorrected APP decoder. In a turbo scheme applying the correction helps reducing the loss only in the lower range of the SNR. However, to match the BER of a digital counterpart, a better compensating circuit needs to be designed. This certainly requires further investigations.

## REFERENCES

- [1] J. Hagenauer, "Der analoge Decoder," German Pat. Appl. No 197 25 275.3, Jun. 1997.
- [2] J. Hagenauer and M. Winklhofer, "The analog decoder," in *Proc. 1998 IEEE International Symposium on Information Theory*, Aug. 1998, p. 145.
- [3] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarköy, "Probability propagation and decoding in analog VLSI," *IEEE Transactions on Information Theory*, vol. 47, pp. 837–843, Feb. 2001.
- [4] M. Moerz, T. Gabara, R. Yan, and J. Hagenauer, "An analog 0.25 $\mu m$  BiCMOS tailbiting MAP decoder," in *Proc. IEEE International Solid-State Circuits Conference*, Feb. 2000, pp. 356–357.
- [5] F. Lustenberger, M. Helfenstein, G. S. Moschytz, H. A. Loeliger, and F. Tarkoy, "All analog decoder for a binary (18,9,5) tail-biting trellis code," in *Proc. 25<sup>th</sup> European Solid-State Circuits Conference*, Sept. 1999, pp. 362–365.
- [6] J. Dai, "Design methodology for analog VLSI implementations of error control decoders," Ph.D. dissertation, University of Utah, Dec. 2002.
- [7] M. Yiu, V. C. Gaudet, C. Schlegel, and C. Winstead, "Digital built-in self test of CMOS analog iterative decoders," in *Proc. IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005, pp. 2204–2207.
- [8] A. G. i Amat, D. Vogrig, S. Benedetto, G. Montorsi, A. Neviani, and A. Gerosa, "Reconfigurable analog decoder for a serially concatenated convolutional code," in *Proc. IEEE Global Telecommunications Conference*, Nov. 2006, pp. 1–6.
- [9] S. Hemati and A. Banihashemi, "Dynamics and performance analysis of analog iterative decoding for low-density parity-check (LDPC) codes," *IEEE Transactions on Communications*, vol. 54, no. 1, pp. 61–70, Jan. 2006.
- [10] M. Arzel, C. Lahuec, F. Seguin, D. Gnaedig, and M. Jezequel, "Semi-iterative analog turbo decoding," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 6, pp. 1305–1316, Jun. 2007.

- [11] M. Arzel, C. Lahuec, F. Seguin, D. Gnaedig, and M. Jézéquel, "Analog slice turbo decoding," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 1, Kobe, Japan, May 2005, pp. 332–335.
- [12] C. Lahuec, G. L. Mestre, M. Arzel, F. Seguin, and M. Jézéquel, "Design and test of a 0.25- $\mu\text{m}$  BiCMOS double-binary analogue APP decoder," in *Proc. 5<sup>th</sup> Analog Decoding Workshop*, June 2006, pp. 35–38.
- [13] F. Gioulekas, M. Birbas, A. Birbas, G. Bilonis, K. Efstathiou, and M. Papamichail, "A high-speed sige bicmos analog map decoder," in *Proc. 5<sup>th</sup> Analog Decoding Workshop, Torino, Italy, June 5-6, Jun. 2006*, pp. 19–22.
- [14] V. Gaudet and G. Gulak, "A 13.3mbps 0.35 $\mu\text{m}$  CMOS analog turbo decoder ic with a configurable interleaver," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 2010–2015, November 2003.
- [15] C. Winstead, J. Dai, S. Yu, C. Meyers, R. Harrison, and C. Schlegel, "CMOS analog MAP decoder for (8,4) hamming code," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 122–131, jan 2004.
- [16] C. Winstead, N. Nguyen, V. Gaudet, and C. Schlegel, "Low-voltage cmos circuits for analog iterative decoders," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 4, pp. 829–841, Apr. 2006.
- [17] D. Vogrig, A. Gerosa, A. Neviani, A. G. i Amat, G. Montorsi, and S. Benedetto, "A 0.35- $\mu\text{m}$  cmos analog turbo decoder for the 40-bit rate 1/3 UMTS channel code," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 753–762, March 2005.
- [18] F. Lustenberger and H.-A. Loeliger, "On mismatch errors in analog-VLSI error correcting decoders," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 4, May 2001, pp. 198–201.
- [19] F. Lustenberger, "On the design of analog iterative VLSI decoders," Ph.D. dissertation, ETH Zrich, No 13879, Hartung-Gorre, Konstanz, Series in Signal and Information Processing, Vol. 2, ISBN 3-89649-622-0, ISSN 1616-671X, Nov. 2000.
- [20] M. Arzel, "Semi-iterative analogue turbo decoding - An application to DVB-RVS-like codes," Thèse de Doctorat, École Nationale Supérieur des Télécommunications de Bretagne, 2006.
- [21] N. Duchaux, C. Lahuec, F. Seguin, M. Arzel, and M. Jezequel, "Effect of BJT's parasitics on computing cells for analog decoders," in *Proc. Joint 6<sup>th</sup> International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference*, Jun. 2008, pp. 301–304.
- [22] C. Berrou, M. Jézéquel, C. Douillard, and S. Kérouédan, "The advantages of non-binary turbo codes," in *Proc. IEEE Information Theory Workshop*, Sep. 2001, pp. 61–63.
- [23] *Digital Video Broadcasting; Interaction channel for satellite distribution systems*, ETSI, 2002, EN 301 790, available at <http://www.etsi.org>.
- [24] B. Gilbert, "A precise four-quadrant multiplier with subnanosecond response," *IEEE Journal Solid-State Circuits*, vol. SC-3, pp. 353–372, Dec. 1968.
- [25] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarköy, "Probability propagation and decoding in analog VLSI," in *Proc. 1998 IEEE International Symposium on Information Theory*, 16-21 Aug 1998, p. 146.
- [26] M. Arzel, C. Lahuec, M. Jézéquel, and F. Seguin, "Analogue decoding of duo-binary codes," in *Proc. International Symposium on Information Theory and its Applications*, Parma, Italy, Oct. 2004, pp. 10–13.
- [27] *Bipolar Transistor Level 504*, 2006, [http://www.nxp.com/acrobat\\_download/other/philipsmodels/tm\\_504.pdf](http://www.nxp.com/acrobat_download/other/philipsmodels/tm_504.pdf).
- [28] J. Paasschens and T. Toom, *Introduction to and Usage of the Bipolar Transistor Model Mextram*, Aug. 2002, Nat.Lab. Unclassified Report NL-UR 2002/823.
- [29] W. Webster, "On the variation of junction-transistor current-amplification factor with emitter current," in *Proc. of the Institute of Radio Engineers*, vol. 42, 1954, pp. 914–920.
- [30] J. Sadowsky, "A new method for viterbi decoder simulation using importance sampling," *IEEE Transactions on Communications*, vol. 38, no. 9, pp. 1341–1351, Sep. 1990.
- [31] C. Winstead and C. Schlegel, "Importance sampling for spice-level verification of analog decoders," in *Proc. IEEE International Symposium on Information Theory*, Jun.-Jul. 2003, p. 103.

Manuscript received February 14, 2008.

Revised manuscript received September 10, 2008.

Affiliation of authors: N. Duchaux, C. Lahuec, M. Arzel and F. Seguin are with Télécom Bretagne, 29238 Brest, Brittany, France.

LIST OF FIGURES

1	Double-binary encoder, code rate 2/3. . . . .	2
2	Trellis associated with the encoder of Fig. 1 (split in four sub-parts for clarity). Branches are labeled with the encoded symbol $X_1X_2Y$ . . . . .	2
3	Tail-biting APP decoder structure and decoding section. . . . .	2
4	Extended Gilbert cell. . . . .	3
5	Emitter-coupled pair. . . . .	3
6	Probability ratio transfer function, i.e. $I_{C2}/I_{C1}$ vs $\exp(\frac{V_{IN}}{V_T})$ of a minimal size emitter-coupled bipolar pair: ideal and simulated (transistor level), $I_{BIAS} = 250\mu A$ , $V_{DD} = 2.8V$ . . . . .	3
7	Ratio of $I_C$ , the collector current obtained from spice-level simulation, over $I_C^*$ , the collector current yielded by the various behavioral models, versus $V_{BE}$ ( $V_{BC} = 0V$ ). . . . .	5
8	Ratio $(\frac{I_{C2}}{I_{C1}})/(\frac{I_{C2}^*}{I_{C1}^*})$ versus $V_{IN}$ for an emitter-coupled pair of minimal size transistors. $I_{C2}/I_{C1}$ and $I_{C2}^*/I_{C1}^*$ are obtained from spice-level simulation and behavioral models respectively. $I_{BIAS} = 250\mu A$ , $V_{DD} = 2.8V$ . . . . .	5
9	Behavioral simulation of the analog APP decoder considering the computing cells ideal and affected by BJTs' parasitic elements, frame length 48 information bits, $I_{BIAS} = 250\mu A$ , $V_{DD} = 2.8V$ . . . . .	6
10	Behavioral simulation of the analog turbo decoder considering the computing cells ideal and affected by BJTs' parasitic elements, frame length 48 information bits, $I_{BIAS} = 250\mu A$ , $V_{DD} = 2.8V$ . . . . .	6
11	Probability ratio transfer functions of a minimal size emitter-coupled pair. Solid line: ideal case; dashed line Effective parasitic $R_E = 0\Omega$ ; dotted line: Effective parasitic $R_E = 220\Omega$ . $I_{BIAS} = 250\mu A$ , $V_{DD} = 2.8V$ (transistor level simulation). . . . .	7
12	Correcting and interconnecting circuit. . . . .	7
13	Behavioral simulation of the analog APP decoder considering the computing cells ideal, affected by BJTs' parasitic elements and counterbalanced, frame length 48 information bits, $I_{BIAS} = 250\mu A$ , $V_{DD} = 2.8V$ . . . . .	8
14	Behavioral simulation of the analog turbo decoder considering the computing cells ideal, affected by BJTs' parasitic elements and counterbalanced, frame length 48 information bits, $I_{BIAS} = 250\mu A$ , $V_{DD} = 2.8V$ . . . . .	8
15	Behavioral simulation of the analog turbo decoder considering the computing cells perfect, affected by BJTs' parasitic elements, with BJTs emitter resistor fully compensated compared to a digital turbo decoder with 8 iterations, frame length 48 information bits, $I_{BIAS} = 250\mu A$ , $V_{DD} = 2.8V$ . . . . .	8

LIST OF TABLES

I	Interleaving . . . . .	6
II	Normalized simulation time of the decoding of one random frame using the different models for an SNR of 4dB . . . . .	6

Reviewer 1:

*The explanation of how the correcting circuit is used is a bit unclear. Perhaps, a figure showing how it is connected with the rest of the decoder would help.*

We added to the Fig. 12 the connection of the correcting circuit with a Gilbert cell.

*The main advantage of your model is computational efficiency. Therefore, I would like to have a comparison of the simulation time for your model versus the spice-level model. It would also be interesting to compare your simulation time with important sampling approach.*

In Section IV, we added paragraph D: Computational cost. This subsection compares the simulation times required to decode a frame using an ideal behavioral model, our proposed model and a spice level model (Mextram model) of a stand-alone APP decoder. A table (Table II) summarizing this was added too. We did not consider importance sampling approach to speed up simulation time. This was not our main goal, as paragraph D states. However, for further works we will most likely use it.

*There are also several small typos:*

Typos have been corrected, hopefully non are left. A space was missing in Extrmodule : Extr module. It refers to Extrinsic module not to Extra module.

*There was some work that used important sampling to improve simulation time to produce BER curves by Winstead/Schlegel. The references [30] and [31] have been added in the paragraph IV.D.*

*Shuhuan Yu did work looking at affect of mismatch in analog decoders which may also be interesting for you.*

We did not find the work of Shuhuan Yu concerning the mismatch in analog decoders. We do not have access to her Ph.D Thesis which might evoke this particular aspect. We would be pleased if you could indicate us available work from her on this matter.

Reviewer 2:

*Page 1 - Column 1 - Line 1(P1-C1-L1): you are writing about a "channel". I think it's better is you refer to a "communication channel".*

Most of the introduction has been rewritten and what was named "channel" is now termed "communication channel" as suggested.

*P1-C1-L15: You use for the second time the English structure "wheter ...or..". It's better if you change this one.*

*P1-C1-L21: "BER and FER ...simulations.". This paragraph is not too clear*

These paragraphs have been rephrased.

*P1-C2-L23: "These effects...performances.". The novelty in the analysis of this non-random effects is clear. Are there other random effects that can impact on decoder performance? Have you considered them? Are they minor or they need a further analysis?*

Transistor mismatch has been studied in [18] and concludes it does not impair the decoder's performance. This was already cited in the introduction. Others random effects such as temperature (on chip gradient for instance), chip to chip process variations have not been considered in this work. This needs a further analysis for higher order model.

*SectionII - Par.D*

*I haven't completely understood what is error analyzed in this section. When I have an Input probability ratio of 100, is  $V_{in} = 0$ ? And if this is true, how is possible that  $I_{c2}/I_{c1}$  in the real simulation results about 15?*

Paragraph II.D has been rewritten. As stated by the last sentence of paragraph C, paragraph D analyzed the conversion error of LLR into probabilities. It also now clearly states that it takes into account all the parasitic effects present in the transistor electrical model (Mextram 504). A ratio of 100 ( $\exp(V_{IN}/V_T) = 100$ ) corresponds to  $V_{IN} \approx 120mV$ ,  $V_{IN} = 0$  is for a ratio of 1.

*Section III*

*P3-C2-L8: "among...current". It's not clear if you have considered the three effects like the most important directly from a literature analysis or if you have selected after a experimental work.*

The Mextram model provided by NXP is described by many equations [27] and [28]. Using those equations, simulations have been run on each different effect to see its impact on the collector current. From this analysis, the three most important

effects have been found. This explanation has been added to Section III.

Reviewer 3:

*The introduction is complete gibberish. Begin by telling the reader about the general area and then guide him to the particular topic.*

*Make sure that concepts and terms are properly introduced.*

Most of the introduction was rewritten to take your remarks into account.

*About the title: what about something like "On the influence of nonideal transistors on analog decoders"?*

We think our title is more precise so we keep it.

*Last two statements in the abstract: the mentioned operating points ("2dB" and "3.25dB") should be stated in terms of BER rather than dB.*

This has been corrected.

*"BER performance" -> simply "BER"*

This has been corrected.

*Beginning of the Introduction: "Therefore, two solutions exist..." This is a repetition of an ancient misunderstanding. These "two solutions" are really the same circuits described in different words.*

This is not in the introduction anymore as it has been rewritten.

*Section II.A: "The target code is..." -> e.g. "For our numerical results, we will assume a turbo code with the following parameters..."*

We changed the first sentence of this section.

*Section II.A: "circular" is not a standard term in coding theory.*

*Use "tail-biting".*

"Circular" has been replaced with "tail-biting".

*Section II.A and II.B: Only the component code (a tailbiting convolutional code) is described. What about the whole turbo code?*

The description of the whole turbo code has been added in Section IV.C, in particular the interleaver (Table I) was added.

*Section II.A: The standard term "binary code" is seriously misused here.*

*Fig. 1 is an ordinary (3,2) binary convolutional encoder; there is nothing nonbinary about it. In other words, a doubly binary convolutional code (as in this paper) is a special case of a binary code, not a nonbinary code.*

Section II.A was partially rewritten to take that into account.

*Fig. 2: Tell the reader that this is only ONE trellis (not four).*

We added in the legend of Fig. 2 that the trellis is split in four sub-parts for clarity.

*Section II.B. It should perhaps be mentioned that the basic modular architecture is that of [3] and [14].*

Added the reference [5](previously [3]): "As in [5], each section is built from four modules..."

*Section III: The term "electrical model" is not defined and requires some explanation. (Another name would be preferable, too.) I assume that this is also based on some equations?*

It is exact that "electrical model" is not appropriate, the exact term is "electrical compact model" as used by foundries such as NXP, textbooks on IC design. However, we understand this can be confusing and we changed the term "electrical model" into "Mextram model". This model is described by some equations [27] as now stated in the first paragraph of Section III.

*How reliable is this "electrical model" in reality?*

This model is provided by the foundry (NXP) and is based on test and characterization of components. This is done before releasing the design kit. Hence, it is as close to reality as possible otherwise it would not be possible to design a working circuit.

*Equation (3): what is the relation between  $I_C$  and  $I_C^*R$ ?*

Equation (3) described the model collector current  $I_C^*R$  when considering the parasitic emitter resistor. The term  $I_C$  was

misused here and has been replaced by  $I_C^{*R}$ . In the same way, Equations (6) and (7) have been corrected.

*Equation (4) requires more explanations.*

The text around the equation has been modified and we hope it is clearer. We named the Early effect terms  $q_1$  as described in [28]. Simplification is made clearer by separating Equation (4) into two equations.

*Section IV.B: The details of the actual turbo code seem to be missing.*

The description of the whole turbo code has been added in Section IV.C, in particular the interleaver (Table I) was added.

*Section V.B: I do not understand the first two statements.*

This part has been rewritten and should be clearer.

*Section V.C: More details on the "digital counterpart" are needed. (How many iterations? Full precision? What about damping?)* The digital counterpart: 8 iterations, floating point precision. This has been added to Section V.C.

*Reference [14] looks strange: (i) what is "ser."? (ii) no publisher; (iii) why "Ch. 4"?. This was actually a PhD thesis at ETH Zurich.*

This reference has been corrected.

*There are also some typos in other references.*

The references have been corrected.

*However, all these particular points are just examples. The whole text needs to be re-thought.* About the organization and the way the paper is written.

The paper starts with a description of a stand-alone APP decoder designed using BJTs. We show next that there is a problem with the way LLRs are translated into probabilities. In order to evaluate the impact of this imperfection, a behavioral model of the transistor is developed and tested for correctness versus transistor level simulation using Cadence® design software tools. Then, from it, behavioral models of a stand alone APP decoder is developed et simulated to obtain the BER. As the APP decoder performance is seriously degraded by the BJT's parasitics, we simulated a turbo scheme using the previous elementary decoder in, to check if it could overtake that degradation. As the answer is no, we studied the way to compensate this error and came up with a simple solution that does improve the BER as our simulations show (even though it is not perfect). We think this is a valid way of presenting our work.

However, some sections were not well written and organized. They have been rewritten: Introduction, II.D, III first paragraph, III.A, III.E, IV.A, V.B, V.C., Section V.A has been split into Section V.A: Most relevant effect and V.B: Counterbalancing circuit.

Reviewer 4:

*There should be many more citations, particularly on behavioral simulations and alternative architectures.* Many citations have been added (20 references for the first version of the paper vs 31 for the revised version). For instance, the references [9] and [2] have been added in the introduction concerning the behavioral simulations and the references [8] and [10] for alternative architectures.

*The "extended Gilbert cell" topology has been used in a number of analog decoders. Authors should note this and add citations.* Section II.C now clearly state that and references [19] and [26] have been added.

*Authors need to explain more about their simulation methods (see below), and should add related citations where appropriate.*

The Section III. has been partially rewritten to explain more how the three effects have been selected. Information has been added in section IV. concerning the Simulink® model.

*The paper addresses physical effects for a particular BiCMOS process. Are the paper's results expected typical of other current BiCMOS processes?*

This BiCMOS process is a typical process. If another process is used, the values of  $R_E$ ,  $V_{AR}$ ,  $I_K$  and the simplifications should be adapted of course but the results should similar. These explanations have been added in Section IV.A.

*The primary text should state that the Mextram model is used in simulations, and identify the model's version number.*

We added in Section II.D that the Mextram model is used in simulations and the version number (504).

*What simulation method is used to evaluate the model equations given in (3)-(6)?*

Equations (3)-(7) are simply solved. The  $I_C$  current is calculated by the Mextram model by Spectre® simulator. The ratio  $I_C/I_C^*$  is simply plotted in Fig. 7 to assess the precision of each model.

*What is the "forward current",  $I_F$ , mentioned in Sec. III-D? How is it evaluated in the simulations?*

$I_F$  is the ideal forward current of the transistor described in [27]. In the forward active region,  $I_F = I_S \exp\left(\frac{V_{BE}}{V_T}\right) \exp\left(-\frac{R_E I_C}{V_T}\right)$ . This has been added in Section III.D.

*Under what conditions can the secondary effects be minimized? Is it possible to bring the circuit closer to the ideal model by, for example, altering the bias current or the transistor dimensions?*

As stated in section V.A, we do not want to change the transistor dimension nor the bias current for cost and speed reasons. Because the Reverse Early and the Webster effects are negligible in a Turbo decoder (see Fig. 15), we did not investigate further on how the secondary effects can be minimized.

*What simulation procedure is used to obtain the BER results? Did the authors use, say, forward-Euler simulation with a fixed frame time-interval for decoding?*

The solver used for decoding is a high order solver : Runge-Kutta ODE solver with variable time-steps. This has been added in section IV.B.

*BER simulations are not very thorough. Is it possible to at least reach  $10^{-5}$  with the proposed behavioral model?*

The simulation time takes about 2 weeks for a BER of  $10^{-4}$  so it would take about 20 weeks to reach  $10^{-5}$ . The simulation time could be reduced using the importance sampling approach [30] [31]. However, this work concentrates on the model accuracy and not on the simulation time, so importance sampling was not considered.

*The paper should document the computational cost of using the behavioral model in place of the ideal model. For example, the paper could compare the measured simulation time needed to evaluate one forward-Euler iteration using the ideal model, vs the proposed model (6), vs the approximate model (9).*

We added the Section IV.D: Computational cost. This section compares the decoding simulation time of the ideal model vs the proposed model (13) vs the Mextram model.

*Fig. 15 is confusing: Does "Analog turbo decoder without parasitic REs" mean the same thing as "Analog turbo decoder with compensation circuit"?*

No this means "Analog decoder with parasitic  $R_E = 0\Omega$ ". This has been rephrased in Fig. 15.

*The introduction should speak to a broad audience. Say something like, "Many communication channels are modeled by AWGN. In an AWGN channel, the demodulated channel output is proportional to the log-likelihood ratio (LLR) of the corresponding emitted symbol [maybe cite some tutorial paper or book]."*

*"Therefore" is misused in the second sentence. Probability propagation should be defined and identified as a major focus of the paper, with citations to fundamental references.*

The introduction has been partially rewritten.

*The discussion in Sec. III-E is overly verbose. Most of the details are immediately obvious from Fig. 7.*

The second paragraph has been nearly entirely removed.

*You cannot "prove" that the simplifications are "acceptable" (Sec. IV-B) unless there is a quantifiable standard of acceptability. Should reword this.*

Absolutely. This has been reworded.

*This statement (Sec. V) is too strong: "The only parasitic that can be counterbalanced by design is the parasitic emitter resistor." Instead, try saying "The parasitic emitter resistor can be counterbalanced by design," or "There are few design options for counterbalancing parasitic effects."*

This sentence has been rephrased: "The parasitic emitter resistor could be counterbalanced by modifying either the size of the BJTs or the bias current."

*The first paragraph of Sec. V-A is very confusing. Please revise this section carefully.*



This paragraph has been partially rewritten.

*I think "Turbo" is usually capitalized in "Turbo codes" and "Turbo decoder". I could be mistaken.*

It seems you are mistaken. For example, in this paper: "Turbo codes with rate- $m/(m+1)$  constituent convolutional codes" by Douillard, C. and Berrou, C.