Scaling of analog LDPC decoders in sub-100nm CMOS processes

Meysam Zargham, Christian Schlegel, Jorge Ernesto Perez Chamorro, Cyril Lahuec, Fabrice Seguin, Michel Jezequel, Vincent Gaudet

Scaling of analog LDPC decoders in sub-100 nm CMOS processes

M. Meysam Zargham [a,b,*], Christian Schlegel [b], Jorge Pérez Chamorro [c], Cyril Lahuec [c], Fabrice Seguin [c], Michel Jézéquel[c], Vincent C. Gaudet [b]

[a] *Electrical Engineering Department, University of Toronto, Canada*
[b] *Electrical Engineering Department, University of Alberta, Canada*
[c] *Electronics Department of Telecom Bretagne, France*

Analog implementations of digital error control decoders, generally referred to as analog decoding, have recently been proposed as an energy and area competitive methodology. Despite several successful implementations of small analog error control decoders, little is currently known about how this methodology scales to smaller process technologies and copes with the non-idealities of nano-scale transistor sizing. A comprehensive analysis of the potential of sub-threshold analog decoding is examined in this paper. It is shown that mismatch effects dominated by threshold mismatch impose firm lower limits on the sizes of transistors. The effect of various forms of leakage currents is also investigated and minimal leakage current to normalizing currents are found using density evolution and control simulations. Finally, the convergence speed of analog decoders is examined via a density evolution approach. The results are compiled and predictions are given which show that process scaling below 90 nm processes brings no advantages, and, in some cases, may even degrade performance or increase required resources.

1. Introduction

The key performance metric in a communication system is data integrity, which can be ensured through sophisticated error control coding techniques. For this reason, decoder circuitry that detects and corrects errors is an essential part of any receiver. Recently low-density parity-check (LDPC) codes have shown near optimum performance with reasonable decoder complexities [1–3]. LDPC codes are decoded using a class of iterative decoding algorithms known as message passing algorithms [4,5]. The core of these iterative algorithms consists of exchanging probabilistic information inside the decoder using a graphical description of the code known as a Tanner graph [6].

Such graph-based decoding algorithms can be efficiently implemented using analog circuitry [7–9]. Most analog decoders exploit the basic exponential characteristics of microelectronic devices to implement fundamental decoder functions. This exponential behavior is found in bipolar transistors as well as CMOS transistors in the sub-threshold region of operation. These decoders often have a better area utilization compared to their digital counterparts. The messages in analog decoders are passed between nodes using one or two wires whereas in a typical digital decoder messages are generally represented by word lengths of at least four binary digits [10]. This means that in a parallel implementation of a decoder, each edge connection of the graph requires at least four wires. Reducing this number requires use of bit-serial arithmetic [11,12] or stochastic computational techniques [13,14]. Simulation and implementation of small analog decoders have shown promising results in terms of energy efficiency or throughput [15–28].

While analog implementations of the computational primitives are elegant, they suffer from physical non-idealities such as mismatch, thermal noise and short-channel effects. These undesirable effects become more severe as the technology scales down. The presence of different short-channel effects causes the transistor model to deviate from the ideal exponential form. As a result these effects can seriously impact the performance of decoders that use short channel length technologies. Currently, there are no reported analog decoders in sub-100 nm processes in the literature. In this study we discuss the feasibility of scaling sub-threshold analog decoders down to the sub-100 nm processes and propose guidelines for transistor sizing to mitigate

* Corresponding author at: Electrical Engineering Department, University of Toronto, Canada.
E-mail addresses: zarghamm@eecg.utoronto.ca (M. Meysam Zargham), schlegel@ualberta.ca (C. Schlegel), jorge.perez@invia.fr (J.P. Chamorro), cyril.lahuec@telecom-bretagne.eu (C. Lahuec), fabrice.seguin@telecom-bretagne.eu (F. Seguin), michel.jezequel@telecom-bretagne.eu (M. Jézéquel), vgaudet@ece.ualberta.ca (V.C. Gaudet).

short-channel effects. We have used the popular BSIM 4 [29] models and studied the behavior of threshold voltage and the sub-threshold swing parameter in nano scale transistors. As a proof of concept several example decoder computational nodes were designed and simulated in 90 nm CMOS process. The post-layout simulation (PLS) results are presented in this paper.Using our findings supported by mathematical analysis of large-scale Tanner graphs using density evolution principles [4,30], we present quantitative performance predictions of what can be expected from sub-threshold analog decoders in sub-100 nm technology.

Preliminary work regarding the mismatch effects in analog decoders was published by Lustenberger et al. who studied the effect of mismatch on decoder performance in bipolar transistors [31]. They assumed that the current factor mismatch [32] has the largest contribution on transistor mismatch and further showed that the effect of such mismatch on the performance of short error control codes was minimal. Frey et al. studied the effect of mismatch on CMOS analog decoders [33]. Results were based on measurements from eleven different semi-discrete short analog decoders. A first analytical mismatch analysis of large codes was given by Winstead and Schlegel [34]. They used density evolution (DE) to predict the behavior of large codes in the presence of mismatch. A current factor mismatch model was used as the mismatch source. In this paper, we present the effect of mismatch on small, moderate and large LDPC decoders using the well established and more appropriate threshold voltage mismatch model [32]. As we will see in Section 4, threshold mismatch is the dominant source of current mismatch in the sub-threshold region of operation.

Aside from mismatch and short-channel effects, scaling transistors introduces increased levels of leakage currents in MOSFETs. Leakage currents enforce a limit on the maximum computational range in a decoder, potentially compromising decoder performance, especially in the low-error regime. Sections 7 and 8 of the paper are devoted to system and circuit level effects of leakage currents on the performance of a decoder.

In Section 9 we investigate thermal noise effects. Dai showed that thermal noise is negligible in sub-threshold analog decoders [35], but unfortunately the noise model he used is valid only in strong inversion. We apply an appropriate sub-threshold noise model to show that the effects of thermal noise are indeed negligible.

The best metric for comparing and evaluating the performance of digital processors is the energy consumption per bit. In order to predict this value, the convergence speed of the decoder needs to be known. In Section 10 we employ DE to estimate the convergence time for analog LDPC decoders. We also discuss the relationship between the input signal to noise ratio and the convergence speed. The rest of the paper is organized as follows. Section 2 reviews background information on analog LDPC decoders. Sections 3–6 discuss the various non-idealities of nano-scale transistors, and Section 11 concludes the paper.

2. Analog LDPC decoders

The LDPC decoding procedure operates on a graphical representation of the code dependencies, on which the sum-product algorithm is executed [5]. In this algorithm probabilistic (soft) information is passed between two types of nodes in the graph: variable and check nodes. These nodes perform local processing on soft information that represents signal reliability. This soft information is generally represented in the form of a log-likelihood (LLR) ratios. LLRs are defined with respect to a binary digit x as

$$\mathrm{LLR}(x) = \ln\left(\frac{P(x=0)}{P(x=1)}\right). \tag{1}$$

The operation at the variable node is to sum the incoming LLRs from the channel and the check nodes that are connected to it. Eq. (2) shows the operation at the variable node:

$$\mathrm{LLR}_{v\to c} = \mathrm{LLR}_{\mathrm{channel}} + \sum_{c'\in C_v\setminus\{c\}} \mathrm{LLR}_{c'\to v}, \tag{2}$$

where $\mathrm{LLR}_{v\to c}$ is the LLR message sent from the variable node v to check node c at each round of the algorithm and $\mathrm{LLR}_{c'\to v}$ is the message sent from the check node c' to variable node v at each round. The set C_v is the set of check nodes connected to variable node v. Note that the sum in (2) excludes the self message from c to v. The message $\mathrm{LLR}_{\mathrm{channel}}$ is the LLR value from the channel. The check nodes perform soft exclusive-or operations on the incoming messages. Eq. (3) describes the check node operation both in LLR and probability form:

$$\begin{aligned}\mathrm{LLR}_{c\to v} &= 2\,\mathrm{atanh}\left(\prod_{v'\in V_c\setminus\{v\}} \tanh(\mathrm{LLR}_{v'\to c}/2)\right)\\ &= \ln\left(\frac{\mathbf{P}(v'_1\oplus v'_2\oplus\cdots\oplus v'_{d_c-1}=0):\forall v'_i\in(V_c\setminus\{v\})}{\mathbf{P}(v'_1\oplus v'_2\oplus\cdots\oplus v'_{d_c-1}=1):\forall v'_i\in(V_c\setminus\{v\})}\right),\end{aligned} \tag{3}$$

where d_c is the number of check node connections. The core circuit for analog implementations of both of these operations in sub-threshold CMOS technologies is the Gilbert multiplier cell [36]. Fig. 1 shows a two-input Gilbert multiplier circuit. Eqs. (4)–(7) hold between the current values in Fig. 1. The circuit produces the normalized product of the input currents. The output currents are normalized by the total current flowing to the first input:

$$I_{x_0y_0} = \frac{I_{x_0}\times I_{y_0}}{I_{y_0}+I_{y_1}}, \tag{4}$$

$$I_{x_0y_1} = \frac{I_{x_0}\times I_{y_1}}{I_{y_0}+I_{y_1}}, \tag{5}$$

$$I_{x_1y_0} = \frac{I_{x_1}\times I_{y_0}}{I_{y_0}+I_{y_1}}, \tag{6}$$

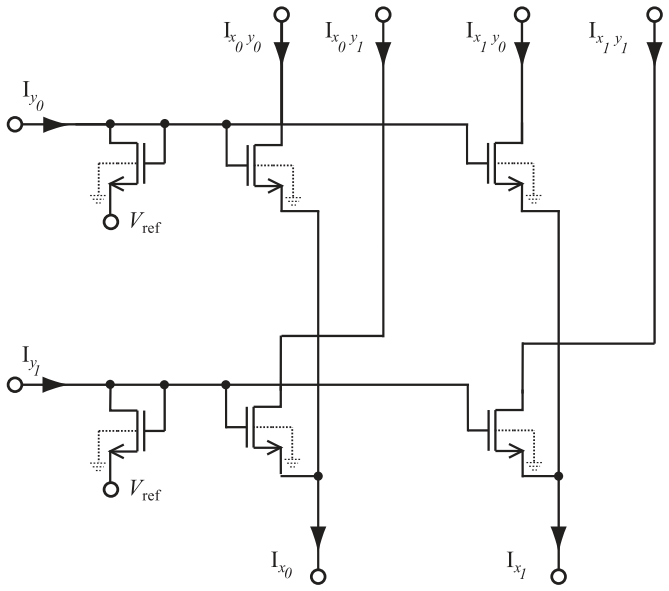

Fig. 1. Two input Gilbert current multiplier.

$$I_{x_1y_1} = \frac{I_{x_1} \times I_{y_1}}{I_{y_0} + I_{y_1}}. \tag{7}$$

These expressions are derived with the assumption of the sub-threshold mode of operation. The drain current I_{DS} of a MOSFET in the sub-threshold region assuming a perfect exponential behavior is

$$I_D = I_0 \exp\left(\frac{V_{GS} - V_t}{nV_T}\right), \tag{8}$$

where I_0 depends on different process parameters, transistor sizing, drain to source voltage and temperature. V_{GS} is the gate to source voltage, V_t is the threshold voltage, V_T is the thermal voltage, and n is the sub-threshold swing parameter.

In sub-threshold analog decoders, currents represent probabilities and voltages represent LLR values. These currents are normalized to the "unity" current I_U. The value of this normalizing current is a design parameter. Probabilities $P(x=0,1)$ are now represented by the currents

$$I_{x_0} = I_U \times P(x=0), \tag{9}$$

$$I_{x_1} = I_U \times P(x=1). \tag{10}$$

We can use the Gilbert multiplier as the core circuit for both types of nodes in our graph. In order to clarify the relationship between decoder nodes and the Gilbert multiplier, here we assume a two input variable node and derive its output in terms of currents in the circuit. In (11) we present the construction of a two-input variable node, i.e.,

$$\begin{aligned} \mathrm{LLR_V} &= \ln\left(\frac{P(x=0)}{P(x=1)}\right) + \ln\left(\frac{P(y=0)}{P(y=1)}\right) = \ln\left(\frac{P(x=0)P(y=0)}{P(x=1)P(y=1)}\right) \\ &= \ln\left(\frac{I_{x_0}I_{y_0}}{I_{x_1}I_{y_1}}\right) = \ln\left(\frac{I_{x_0y_0}}{I_{x_1y_1}}\right), \end{aligned} \tag{11}$$

where $\mathrm{LLR_V}$ is the output LLR of the variable node. The output of a two input check node, $\mathrm{LLR_C}$, is given by

$$\begin{aligned} \mathrm{LLR_C} &= \ln\left(\frac{P(x=0)P(y=0)+P(x=1)P(y=1)}{P(x=0)P(y=1)+P(x=1)P(y=0)}\right) = \ln\left(\frac{I_{x_0}I_{y_0}+I_{x_1}I_{y_1}}{I_{x_0}I_{y_1}+I_{x_1}I_{y_0}}\right) \\ &= \ln\left(\frac{I_{x_0y_0}+I_{x_1y_1}}{I_{x_0y_1}+I_{x_1y_0}}\right), \end{aligned} \tag{12}$$

where we assume a perfect exponential characteristic for the transistors in the circuit.

3. Scaling the decoder

Semiconductor devices are scaled aggressively for high performance and integration. Scaled transistors tend to operate faster but their behavior deviates from long-channel models and undesired effects arise. The gate loses its control over the channel while the drain becomes more prominent. There are many different short-channel effects, primarily channel length modulation, drain-induced barrier lowering (DIBL), punchthrough, velocity saturation and hot carrier effects [37]. In the sub-threshold region of operation the threshold voltage V_t and sub-threshold swing nV_T play the most important roles. This is due to the exponential dependence between the current and these parameters. Therefore, the only short-channel effects that we consider here are the ones affecting the threshold voltage or the sub-threshold swing. In the first sub-section we will briefly introduce short channel effects. Readers who prefer to focus on the results can skip this sub-section without a loss of continuity.

3.1. Short channel theory

In this section we are considering the size dependence of the threshold voltage and nV_T and propose a number of design rules for building LDPC computational nodes in the nano regime.

In the long-channel MOSFET equations, the threshold voltage is only a function of process parameters and source-to-bulk voltage. As the channel length of the MOSFET is scaled down, the threshold becomes a complicated function of the applied voltages, transistor sizing as well as the process parameters.

The classic equation for the threshold voltage in a MOS transistor is [29]

$$V_t = V_{FB} + \Phi_s + \gamma\sqrt{\Phi_s + V_{sb}} = V_{FB} + \Phi_s - \frac{Q_B}{C_{ox}}, \tag{13}$$

where V_{FB} is the flat band voltage, Φ_s is the surface potential, C_{ox} is the oxide capacitance, and Q_B is the charge in the depletion region. The last term in (13) represents the voltage across the depletion region. The parameter γ is the body bias coefficient, given by

$$\gamma = \frac{\sqrt{2q\varepsilon_{si}N_{substrate}}}{C_{ox}}, \tag{14}$$

where $N_{substrate}$ is the doping level of the substrate, and ε_{si} is the permittivity of silicon. Φ_s is a function of channel doping and temperature. In short-channel transistors, channel doping is no longer uniform. The variations in channel doping are both vertical and lateral. The lateral doping change is due to a process called pocket (Halo) implant [38]. During this process, the doping near the source and drain regions is increased. Therefore, as the channel becomes shorter, the average effective concentration becomes higher and increases the surface potential. Thus, V_t becomes a strong function of L. This dependency is modeled in BSIM 4 by a term added to the classic threshold value. Assuming no body effect the threshold voltage in the presence of pocket implants is [29,39]

$$V_t = V_{FB} + \Phi_s - \frac{Q_B}{C_{ox}} + K_1\left(\sqrt{1 + \frac{\mathrm{LPE0}}{L}} - 1\right)\sqrt{\Phi_s}, \tag{15}$$

where K_1 and LPE0 are process parameters. Thus threshold voltage in nano-scale processes is a *decreasing* function of effective length. The width of a transistor also plays a role in its effective threshold. In a long-channel MOS transistor, the source and drain regions are far away from the channel. Therefore, the fields can be assumed to be vertical in most of the channel. In short-channel devices, with channel lengths less than 0.35 μm, the fabrication process follows a pattern known as shallow trench isolation (STI). This isolation technique is used to prevent leakage currents between adjacent devices. As a result of this process, a big portion of gate electric fields, called fringing fields, terminate on source and drain ends. Part of the depletion charge is due to these fringing fields. In order to calculate the exact threshold value, these charges have to be taken into consideration. The new threshold voltage in an STI process, assuming no body effect is

$$V_t = V_{FB} + \Phi_s - \frac{Q_B}{C_{ox} + 2C_F}, \tag{16}$$

where C_F is the fringing field capacitor and is only a function of L. Since Q_B is a function of $W \times L$, the threshold voltage increases with larger device width and smaller channel lengths. This higher threshold voltage is desirable for sub-threshold applications as it provides a wider range of accurate operation.

Another important phenomenon that affects V_t is the DIBL effect. In long-channel transistors the drain can only affect a small part of the channel. As the channel shrinks, field lines from the drain can influence the charge and potential throughout the

channel and force V_t to be a function of drain voltage. Drain voltage increases Φ_s and hence lowers the potential energy for electrons [37]. In analog decoders, we do not have control over the drain to source voltage V_{DS}. Therefore, DIBL is undesirable as it changes V_t during the course of the decoder operation. V_{DS} in a sub-threshold analog decoder typically ranges from 100 to 400 mV. For these small values of V_{DS}, DIBL can be modeled with a first order linear term added to the threshold [40,37]:

$$\Delta V_t = -\eta V_{DS}. \tag{17}$$

The parameter η has a $1/L$ relationship with the channel length. Ideally we want η to be zero. Therefore longer channel lengths are less sensitive to V_{DS} variations.

Short-channel effects can also affect the sub-threshold swing parameter n. This parameter plays an important role in the accurate operation of the Gilbert multipliers. Also, conversion from LLRs to probabilities is directly proportional to this value. Ideally we want this number to be constant throughout the weak inversion region. In small channel lengths the gate control over Φ_s decreases. Therefore, the variations in Φ_s due to variations in the gate potential are smaller [41]. This relationship indicates a bigger sub-threshold swing. According to BSIM 4 models:

$$n \propto \frac{1}{\cosh(L/l_t)-1}, \tag{18}$$

where l_t is a process parameter. Such behavior implies a strong relationship between n and the effective length of the transistor in small transistor lengths. This strong dependence is undesirable in sub-threshold analog decoders. This is because due to the imperfections in fabrication processes, the size of the fabricated transistors are different from the designed size. In addition, large values of n are generally undesirable as they tend to slow down the circuit. The value of n also determines how fast the drain current changes with regards to variations in the gate voltage. Therefore, using longer transistors is advantageous in reducing the uncertainty in the sub-threshold swing.

3.2. Design rules

From the above discussions, we know that there exists a tradeoff in choosing the length and the width of the transistors. Smaller lengths tend to increase the threshold voltage and hence our region of operation. In addition, the gate to bulk capacitance and the output resistance in the device are proportional to device sizing. Using smaller transistor sizes reduces the energy consumption and increases speed of the nodes. However, small transistor lengths amplify the unwanted short-channel effects such as DIBL and length dependency of the sub-threshold swing. They also increase the value of the sub-threshold swing. Large transistor widths are desirable for higher threshold voltage and achieving smaller inversion coefficient. Therefore, using large transistor widths, we can afford large transistor lengths to combat the short-channel effects while the threshold voltage is still high enough due to the large transistor width. On the downside, they slow down the circuit and increase the leakage currents in the transistor. In fact the real trade off is between size, power/speed versus accuracy of the circuit.

As a proof of concept and to study the effect of these non-idealities in decoders, we implemented a variable node employing these rules.

The circuit adds the two incoming differential voltages that represent LLRs in the decoder. The check node circuit is very similar to the variable node. The only difference is the way the Gilbert multiplier outputs are connected to each other. Therefore we are only presenting variable node results here. The post-layout simulation results presented are based on Cadence using STMicroelectronics BSIM 4 models.

The variable node is a two-input function. Therefore in order to be able to compare these different implementations and compare their accuracy we generated an error surface for the set of all meaningful inputs with a 1 mV step size. The error surface contours show the absolute difference between the circuit output and the ideal output which is the addition of the two input LLRs. Therefore each point on the error surface represents the error using, $\mathrm{LLR}_{error} = |\mathrm{LLR}_{in1} + \mathrm{LLR}_{in2} - \mathrm{LLR}_{Circuit}|$. Where, $\mathrm{LLR}_{in1,2}$ are the input LLRs and $\mathrm{LLR}_{Circuit}$ is the LLR output from the circuit simulations. The Gilbert multiplier in the computational nodes can be implemented using either pMOS or nMOS transistors. The choice of pMOS transistors over nMOS was previously recommended by [42]. pMOS transistors give the designer the luxury of connecting the bulk to the source and thus, eliminating the threshold variations due to the change in the source voltage. A pMOS based variable node was implemented in 90 nm generic CMOS process. The choice of the transistors lengths and widths are shown in Table 2 and part of the node layout is shown in Fig. 3. The surface error for the implemented node is shown in Fig. 2.

The implemented node has an absolute error of less than 0.055 LLR units for LLR input range between -1 and +1. This error increases to 0.15 LLR units for input LLR swing of -4.5 to +4.5. The designed node is accurate for small input LLRs but the error increases as high as 1.4 LLR units for larger input LLR values. This increase in the error at higher LLR swings is mainly due to the clipping of LLRs. Fortunately, the clipping of the output LLR is tolerated by most codes. These ideas are discussed in Section 7.

Another important characteristic of the designed node is its average speed and power consumption. Previous studies [44] have shown that the speed of the decoder is independent of the variance of the delay distribution and is a function of the average delay. Therefore for the purposes of speed comparison in this section and convergence time estimation in Section 10, we have used the average speed of the node. In our simulations, we simulate the speed and energy consumption of the implemented circuit under 10 different input conditions. In each test setup we change one or both inputs in the circuit and wait for the output to reach 95% of its final value. We then calculate the time and energy in each case. Table 1 presents these different test setups. In this table "High" is an input differential voltage of 600 mV and

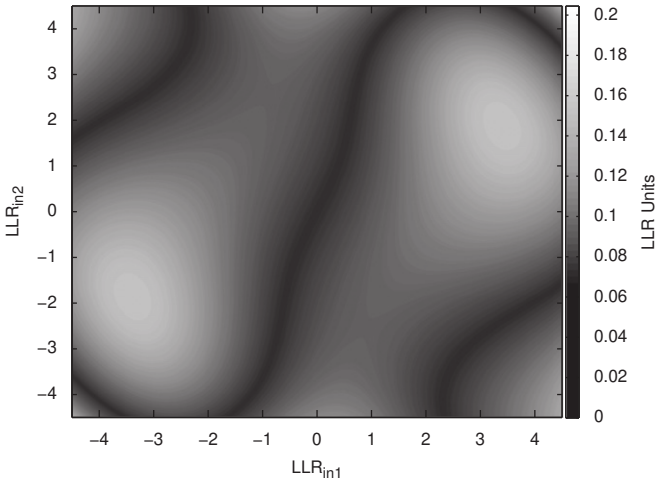

Fig. 2. Absolute error surface, $LLR_{error} = |\mathrm{LLR}_{in1} + \mathrm{LLR}_{in2} - \mathrm{LLR}_{Circuit}|$, of a two-input pMOS variable node designed in 90 nm CMOS. The inputs and the error are both in terms of LLR values.

Table 1
Test conditions for power/speed measurements.

Tests	Input one		Input two	
	Initial	**Final**	**Initial**	**Final**
Test1	Zero	High	Zero	Zero
Test2	High	Zero	Zero	Zero
Test3	Zero	Zero	Zero	High
Test4	Zero	Zero	High	Zero
Test5	Zero	High	Zero	High
Test6	High	Zero	High	Zero
Test7	Zero	High	Zero	−High
Test8	High	Zero	−High	Zero
Test9	Zero	−High	Zero	High
Test10	−High	Zero	High	Zero

Table 2
PLS results of the average energy/speed and transistor sizing of the implemented node.

Technology	90 nm
Length (μm)	0.2
Width (μm)	2.0
Energy (pJ)	0.38
Settling time (μs)	0.05
Circuit simulations	Fig. 2
BER simulations	Fig. 4

"−High" is an input differential voltage of −600 mV. We further verified that adding more test cases did not change the average delay significantly. Table 2 shows these values as well the transistor sizing we use for the Gilbert multiplier.

3.3. Decoder performance under short-channel effects

Now the natural question is how a decoder performs using nodes with non-idealities due to short-channel effects. True simulation of large fully parallel analog decoders using computer simulations is not possible with the current computing powers. Different simulation techniques have been proposed in the literature to increase the accuracy of simulations [26,45,46]. Here we are performing system-level C analysis while incorporating post-layout simulation data from Cadence. LDPC decoder consists of many identical nodes (Variable and Check node) that are connected in a certain fashion. Therefore, neglecting mismatch (which is introduced in the next section), the response of each node is similar to the other nodes in its group. We took advantage of this structure to perform system level C/MATLAB simulations while including the post layout simulation data. We characterized a node by generating a large lookup table for every meaningful input/output (30% beyond output clipping range) with a quantization level of 1 mV. In the C simulations, the node was modeled as a black box that functioned according to the generated lookup table. We use a (3,6)-regular length-816 LDPC code from [47] and perform Monte-Carlo simulations with 50 iterations. An iteration is a round of message passing between variable, check nodes and back. Fifty frame errors are generated for every data point. Binary phase shift keying (BPSK) modulation is used over an additive white Gaussian noise (AWGN) channel. We verify the bias independence of the results using all-zero, all-one and random codewords. The simulations represent a successive substitution Monte-Carlo simulation of the decoder, and hence some behavior in real, continuous-time decoders is lost.

Fig. 4 shows the bit error rate (BER) plot. This plot was generated using the Monte-Carlo simulations we proposed. The implemented node result is very close to the floating-point

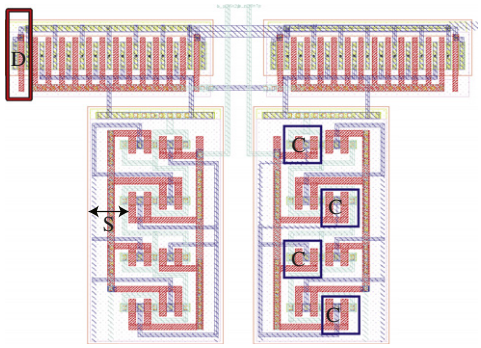

Fig. 3. Part of the layout of the 90 nm implemented node. In this layout dummy gates where used to reduce the STI effects. Also common-centroid and spacing techniques were used to reduce the well proximity effects. Examples of dummy gates in the layout are shown with letter "D", common-centroid transistors with "C" and more than three gate length spacings with "S" [43].

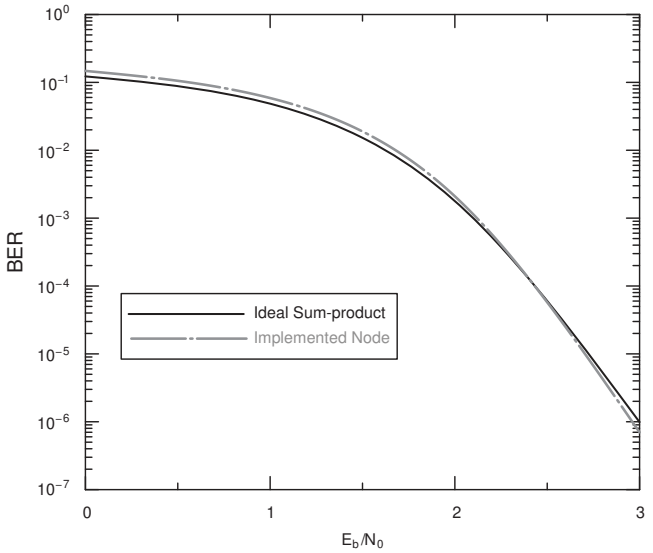

Fig. 4. Bit Error curve for (3,6)-regular length-816, LDPC code using the implemented node.

sum-product algorithm. It is very interesting that the 90 nm circuit has a better performance than the ideal simulation at higher E_b/N_0 values. This is mainly due to the fact that the sum-product algorithm is not optimal. Such behaviors have previously been observed in analog decoders [44]. Nevertheless the loss in the performance is negligible up to a BER of 10^{-6}. Based on these post-layout simulation results we conclude that the short-channel effects can be controlled in sub-threshold analog decoders. If we ignore the mismatch effects and the high leakage currents, they can still decode with reasonable performance in nano-scale processes. The next section is dedicated to the limitations of mismatch on decoder performance.

4. Matching properties of CMOS transistors

Modern CMOS processes are highly sensitive to process variations. Phenomena such as local V_t variations can make results deviate from circuit simulators. Mismatch can be caused by many different variations in the fabrication process. The fabricated width and length of the transistors can be different from the implemented dimensions and channel doping varies in local areas. Also the oxide thickness varies across the die. The local threshold variations of a transistor are a function of the L, W,

the surface charge variations and the channel doping [48]. In short-channel CMOS processes, different short-channel effects can increase mismatch [49]. For example, the effective length, the effect of halo implants and charge sharing due to DIBL effect, can strongly increase the threshold mismatch [50,51]. It has been shown through many different measurements that the mismatch factor A_{Δ}, which will be introduced in this section, does not improve for processes smaller than 100 nm [50].

The local V_t variations contribute to an approximate Gaussian distribution of the threshold voltage with variance $\sigma^2_{V_t}$ [32,52]:

$$\sigma_{V_t} = \frac{A_{\Delta}}{\sqrt{WL}}. \tag{19}$$

A_{Δ} is a process parameter with units mV μm. The value of this parameter depends on the fabrication process and can be found in the process documentation. The value of A_{Δ} decreases with oxide thickness and hence, scales with technology [53,54]. However, this behavior does not extend to small processes. The improvement in A_{Δ} is limited for sub-100 nm processes and in some cases even reverses and A_{Δ} increases with scaling [41]. Therefore, scaling alone will not help us much if at all in smaller processes. Mismatch is also present in μ, C_{ox}, W and L, where μ is the mobility. The net effect of these variations on current factor, $\beta = \mu C_{ox} W/L$, also has a Gaussian distribution and affects the drain current mismatch. These variations are usually due to edge roughness in transistor size as well as local variations in C_{ox} and μ. The statistic of the current factor mismatch is given by

$$\sigma_{\Delta_\beta/\beta} = \frac{A_{\beta}}{\sqrt{WL}}. \tag{20}$$

In the sub-threshold region of operation, where most analog decoders operate, values of I_{DS} are small, A_{β} is a small number on the order of 1% and it is negligible with respect to threshold mismatch. In addition mismatch variations in the sub-threshold swing parameter, n, are also negligible [55], making V_t variation the dominant effect [41]. Hence, in this section our focus is on V_t variations and their effects on sub-threshold current mismatch. The typical value for A_{Δ} in nano-scale processes is around 4 mV μm [41]. Using this value the total estimated threshold mismatch for a minimum-sized transistor in a typical 65 nm CMOS is $\sigma_{V_t} = 35$ mV which can have severe effects on the performance of sub-threshold analog decoders. We investigate these effects in the next section.

5. Statistics of LLR mismatch in analog decoders

In this section we first derive the output current equations of the Gilbert multiplier under mismatch. We then apply the result to the computational nodes of LDPC decoders and derive error statistics in these nodes.

5.1. Gilbert multiplier in the presence of mismatch

As seen in the previous section, the threshold voltage of a transistor varies with mismatch, which causes the drain current to vary as

$$I_{DS,Circuit} = I_{DS} \cdot \exp\left(-\frac{\Delta_{V_t}}{nV_T}\right), \tag{21}$$

where Δ_{V_t} is a Gaussian variable with zero mean and variance $\sigma^2_{V_t}$. Using (21) we derive the mismatch versions of (4)–(7) as

$$I_{x0y0} = \frac{I_{x0} I_{y0} \varepsilon_1 \varepsilon_4}{I_{y0}\varepsilon_1\varepsilon_4 + I_{y1}\varepsilon_2\varepsilon_3}, \tag{22}$$

$$I_{x1y1} = \frac{I_{x1} I_{y1} \varepsilon_2 \varepsilon_5}{I_{y0}\varepsilon_1\varepsilon_6 + I_{y1}\varepsilon_2\varepsilon_5}, \tag{23}$$

$$I_{x0y1} = \frac{I_{x0} I_{y1} \varepsilon_2 \varepsilon_3}{I_{y0}\varepsilon_1\varepsilon_4 + I_{y1}\varepsilon_2\varepsilon_3}, \tag{24}$$

$$I_{x1y0} = \frac{I_{x1} I_{y0} \varepsilon_1 \varepsilon_6}{I_{y0}\varepsilon_1\varepsilon_6 + I_{y1}\varepsilon_2\varepsilon_5}, \tag{25}$$

where

$$\varepsilon_i = \exp\left(\frac{\Delta_{V_{t(i)}}}{nV_T}\right), \tag{26}$$

represents the threshold mismatch in transistor M_i, from Fig. 1 and has a log-normal distribution.

5.2. Variable node behavior in the presence of mismatch

The functionality of a two-input variable node is described by (11). Therefore, the output LLR of the circuit under mismatch using (22)–(25) is

$$\mathrm{LLR}_{\mathrm{variable}} = \mathrm{LLR}_i + \ln\frac{I_{y_0}(\varepsilon_1^2\varepsilon_4\varepsilon_6) + I_{y_1}(\varepsilon_1\varepsilon_2\varepsilon_4\varepsilon_5)}{I_{y_1}(\varepsilon_2^2\varepsilon_3\varepsilon_5) + I_{y_0}(\varepsilon_1\varepsilon_2\varepsilon_4\varepsilon_5)}, \tag{27}$$

where LLR_i is the ideal output LLR. Using (26) we can rewrite the second term in (27) in terms of the input probability $P_{y0} = I_{y_0}/I_U$,

$$\lambda_{\mathrm{mismatch}} = \ln\frac{1 + P_{y0}(e^{\Delta_{V_{t1}}/nV_T + \Delta_{V_{t6}}/nV_T - \Delta_{V_{t2}}/nV_T - \Delta_{V_{t5}}/nV_T} - 1)}{1 + P_{y1}(e^{\Delta_{V_{t2}}/nV_T + \Delta_{V_{t3}}/nV_T - \Delta_{V_{t1}}/nV_T - \Delta_{V_{t4}}/nV_T} - 1)}. \tag{28}$$

New variables can be defined as, $\alpha = ((\Delta_{V_{t1}}/nV_T) - (\Delta_{V_{t2}}/nV_T))$, $\beta = ((\Delta_{V_{t5}}/nV_T) - (\Delta_{V_{t6}}/nV_T))$, $\theta = ((\Delta_{V_{t3}}/nV_T) - (\Delta_{V_{t4}}/nV_T))$. These new variables are Gaussian with $\sigma^2_{\mathrm{new}} = 2 \times \sigma^2_{V_t}$ and mean zero. Re-writing the mismatch in terms of α, β and θ we obtain

$$\lambda_{\mathrm{mismatch}} = \ln\frac{1 + P_{y0}(e^{\alpha-\beta} - 1)}{1 + P_{y1}(e^{\theta-\alpha} - 1)}. \tag{29}$$

5.3. Check node behavior in the presence of mismatch

Eq. (12) describes the function of a check node with two inputs and one output. Using (22)–(23) in (12) we obtain

$$\mathrm{LLR}_{\mathrm{check}} = \ln\left(\frac{\varepsilon_1^2\varepsilon_4\varepsilon_6 P_{x0} P_{y0}^2 + \varepsilon_1\varepsilon_2\varepsilon_4\varepsilon_5 P_{y0} P_{y1} + \varepsilon_2^2\varepsilon_3\varepsilon_5 P_{x1} P_{y1}^2}{\varepsilon_1^2\varepsilon_4\varepsilon_6 P_{x1} P_{y0}^2 + \varepsilon_1\varepsilon_2\varepsilon_3\varepsilon_6 P_{y0} P_{y1} + \varepsilon_2^2\varepsilon_3\varepsilon_5 P_{x0} P_{y1}^2}\right). \tag{30}$$

Eqs. (27) and (30) are the LLR output functions for the statistics of the output LLR for a two-input variable node and check node, respectively. A multi-input node consists of several such basic blocks cascaded together. In a sub-threshold analog decoder, the output of each one of these blocks is normalized before passing to the next node for processing. This normalizing block can be realized by making sure that the output current probabilities always add up to I_U.

6. Mismatch performance of the decoder

Monte-Carlo decoder simulations are presented for three different codes: an (8,4) Hamming code, a moderate size (3,6)-regular length-816 LDPC code and a large (3,6)-regular code of length 4000. The two LDPC codes were taken from [47]. Seventy frame errors are simulated for every data point. A maximum of 50 iterations were performed for each frame. The simulations are performed under AWGN channel assumptions with BPSK modulation. In order to consider the circuit non-idealities, we use the post-layout simulation results of a pMOS based, 65 nm implemented node. Therefore the computations used in the simulations

are based on the non-ideal circuits presented in the previous section. As a result, the following mismatch curves represent both the circuit non-idealities and the mismatch effect. The mismatch is added using the relationships derived in the previous section.

Fig. 5 presents the results of an (8,4) Hamming code in the presence of mismatch, and shows that the mismatch has a minor effect on a small (8,4) Hamming code which is consistent with measurement results from [56]. Fig. 6 shows the result for the length-816 LDPC code and finally Fig. 7 represents the bit error rate curves of the length-4000 LDPC code. σ_V in these figures represents the standard deviation of the variable node and σ_C is standard deviation in the check node. We use different threshold mismatch variances for the variable node and check nodes. This is because the decoder is far more sensitive to mismatch in the check node than the variable node. This observation is demonstrated in Fig. 7. This in an important observation since it will allow for savings in area for the variable node. We also performed mismatch simulations using ideal computational nodes. Our simulations show that the circuit imperfections have a minor effect compared to mismatch. These graphs are not presented in this paper since the results were very similar to circuit based mismatch simulations we presented in Figs. 5–7. Table 3 maps the threshold standard deviation values to the transistor sizing. In order to generate the data for this table, the parameter A_Δ is set to 4 mV μm and we assume $W=2\times L$ which ensures an inversion coefficient, I_C smaller than 0.1 for the normalizing current, $I_U = 1$ μA.

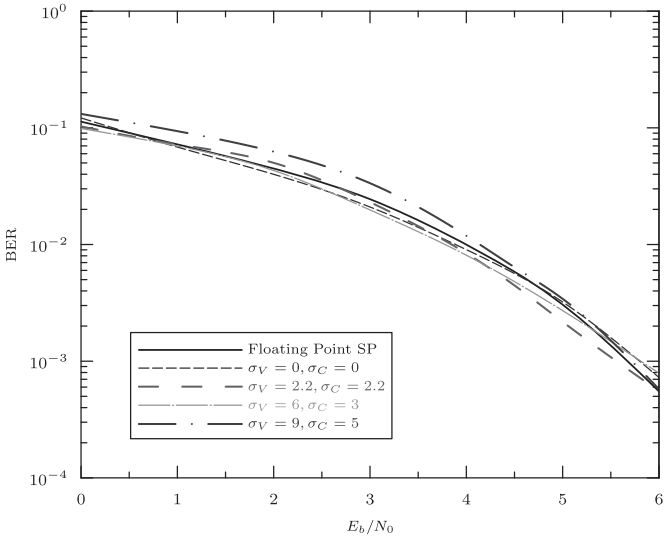

Fig. 5. Hamming (8,4) code in the presence of mismatch.

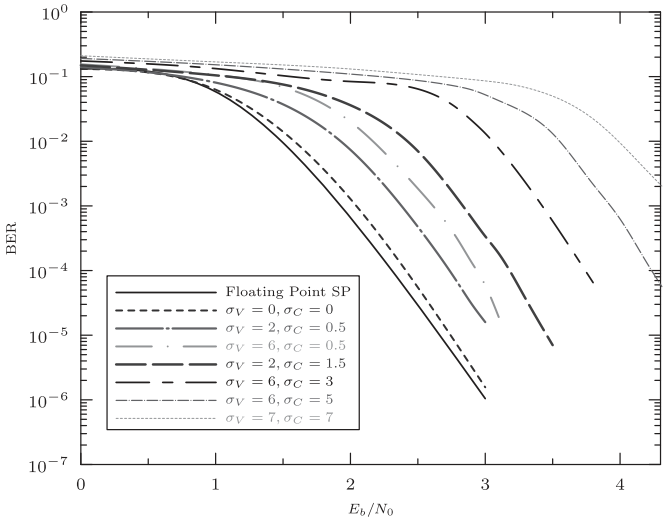

Fig. 6. Length-816, (3,6)-regular code in the presence of mismatch.

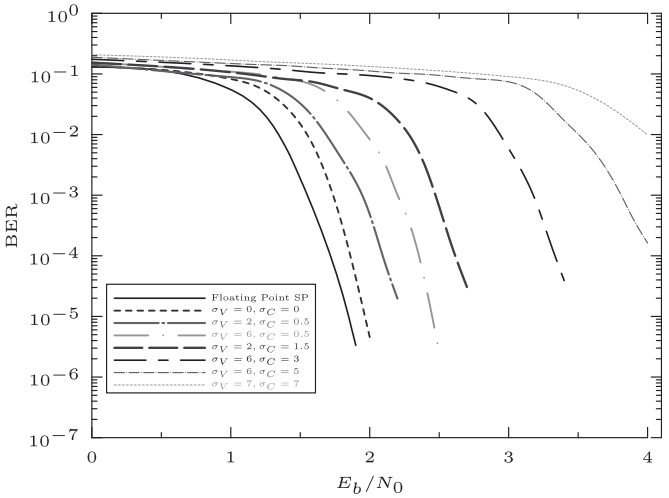

Fig. 7. Length-4000, (3,6)-regular code in the presence of mismatch.

Table 3
Mapping of the threshold standard deviation to transistor sizing.

σ_V	W (μm)	L (μm)	$W\times L$ (μm)^2
0.25	22.64	11.32	256.3
0.50	11.32	5.66	64.1
2.00	2.84	1.42	4.0
5.00	1.14	0.57	0.6
9.00	0.64	0.32	0.2

Table 4
PLS results of the average energy/speed for computational nodes with different transistor sizes.

Transistor $W\times L$ (μm)^2	**Node energy** (pJ)	**Settling time** (μs)	σ_V
nMOS-0.2 × 0.7	0.25	0.02	10.7
pMOS-0.2 × 2.0	0.38	0.05	6.3
nMOS-2.3 × 1.69	0.66	0.09	2.0
pMOS-4.53 × 1.86	3.53	0.75	1.3

Mismatch can severely affect the performance of analog LDPC decoders. These effects become more prominent in larger codes and are barely noticeable in smaller ones. Therefore, building large sub-threshold analog decoders even in sub-100 nm processes requires transistor over-sizing to overcome the mismatch. Large transistors are power hungry and introduce large capacitance that will slow down the decoder. To demonstrate how the transistor sizing would affect the speed and energy consumption of the decoder, Table 4 provides a few data points for the post layout simulation results of computational nodes with different transistor sizes.

Our simulations also demonstrate that check-node mismatch has a stronger impact on the decoder than the variable node mismatch. As a result it becomes clear that analog LDPC decoders based on sub-threshold CMOS transistors cannot take full advantage of small transistor sizes.

7. Leakage currents in MOSFET

The currents in a real transistor can never be fully turned off. As we decrease the gate to source voltage of a transistor, we

ideally expect all the currents to go to zero. In a real transistor, however, these currents reach a lower limit caused by leakage currents. These unwanted currents increase as we scale the transistors. The leakage currents enforce a lower bound on the probabilities and consequently an upper bound on the maximum possible LLRs in the decoder that we refer to as "Clipping". In this section we review the sources of leakage currents in CMOS circuits and their impact on the performance of decoders.

Depending on the context, leakage currents are often interpreted differently. Digital circuit designers refer to leakage as any current flowing when gate to source voltage is less than the threshold voltage, $V_{GS} < V_t$. It is obvious that in the context of analog decoding we have a different definition for leakage currents. Fig. 8 better explains this. The graph shows the drain current of a MOSFET transistor in a sub-100 nm CMOS process for two different drain voltages.

The main mechanisms that contribute to leakage currents in the MOSFET are the pn junction leakage, gate induced drain leakage (GIDL) and the gate leakage. The pn junction leakage is the classic leakage current in MOS transistors. The source of this current is the reverse-bias voltage applied to the pn junction formed by drain (n-type) and substrate (p-type) in an nMOS transistor and is a function of the transistor width [57]. In short-channel devices and in the presence of high drain currents, this leakage converts to band-to-band tunnelling (BTBT). BTBT is the effect of electrons tunnelling from the valence band in the p-region to the conduction band of the n-region [58]. This current is much higher than the classic pn junction leakage and can increase the leakage by more than an order of magnitude. Another form of tunnelling leakage current in MOSFETs is the GIDL current. Such leakage can be very problematic in sub-threshold analog decoders since with GIDL the drain current increases exponentially with lower values of gate to source voltage. GIDL happens when the negative gate voltage causes the p-region near drain and source to become heavily doped. In the presence of the gate voltage we get strong fields which enables tunnelling currents. Tunnelling occurs between the depleted region in n and the heavily doped substrate. This tunnelling current is a strong function of drain voltage and the negative gate voltage [40]. The last type of leakage in the MOSFET is gate leakage. In the context of sub-threshold analog decoders, however, gate leakage is not very severe. The reason is that in the sub-threshold mode of operation the voltage drop across the oxide is small and the gate leakage is negligible. Analog decoders can tolerate small gate

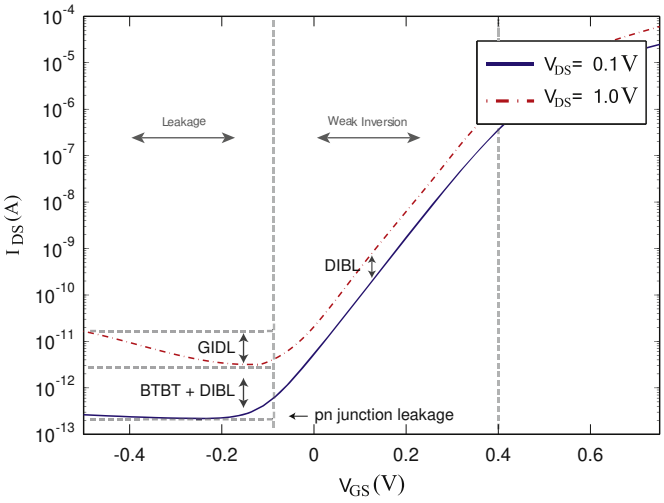

Fig. 8. Simulation of the drain currents in a sub-100 nm CMOS process for two different drain voltages.

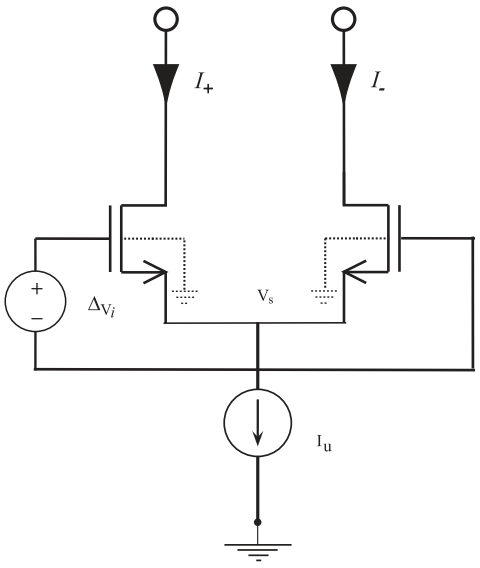

Fig. 9. Differential pair circuit.

leakages. The bipolar decoders that have been previously reported [15,16] are a good example of this fact. The base current in a bipolar transistor is much larger than the MOSFET sub-threshold gate leakage.

BTBT and GIDL tunnelling currents require high drain voltages. Therefore, as long as we keep drain voltage small, we are mainly dealing with pn junction leakage.

The issue of leakage currents links directly to LLR clipping in the decoder. The clipping of LLRs only happens in variable nodes. This is because the check node output is never larger than its inputs. In order to fully understand the behavior of decoder in the presence of leakage currents, we derive the output of a variable node under leakage conditions. A variable node consists of two input differential pairs connected to a Gilbert multiplier. Fig. 9 shows the differential pair. The input LLRs are of the form $\lambda_i = \Delta_{V_i}/nV_T$, where Δ_{V_i} is the input differential voltage. Ideally we want the output of the variable node to be the sum of the two input LLRs:

$$\lambda_{out} = \lambda_1 + \lambda_2. \tag{31}$$

Assuming all transistors use the same transistor width the pn junction leakage current is roughly the same for all transistors. This leakage can be modeled with a constant leakage current $I_{Leakage}$, added to the drain current. Using this model, the currents in Fig. 9 follow these equations:

$$I_+ + I_- = I_U, \tag{32}$$

$$I_+ = I_0 \exp\left(\frac{V_+ - V_S}{nV_T}\right) + I_{Leakage}, \tag{33}$$

$$I_- = I_0 \exp\left(\frac{V_- - V_S}{nV_T}\right) + I_{Leakage}, \tag{34}$$

where I_0 is the sub-threshold current at zero gate source drive. We have ignored the leakage current in the normalizing current source. Using these equations we can rewrite the current in terms of the differential voltage drive (Δ_{V_i}), leakage current and the normalizing current I_U,

$$I_+ = \frac{\exp(\Delta_{V_i}/nV_T)(I_U - I_{Leakage}) + I_{Leakage}}{1 + \exp(\Delta_{V_i}/nV_T)}, \tag{35}$$

and

$$I_- = I_U - I_+. \tag{36}$$

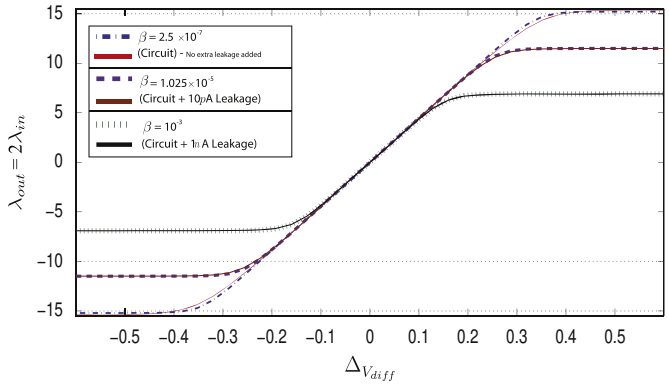

Fig. 10. Circuit simulations and analytical plots of output LLR versus differential input voltage for different leakage current values.

Applying the leakage model to the Gilbert multiplier in Fig. 1 and using the output currents from the differential pair we obtain the output of the variable node under leakage conditions:

$$\lambda_{\text{out}} = \ln\left(\frac{(1+e^{\lambda_2}-2e^{\lambda_1+\lambda_2})\times\beta+e^{\lambda_1+\lambda_2}}{1+(-2+e^{\lambda_2+\lambda_1}+e^{\lambda_1})\times\beta}\right), \tag{37}$$

where $\beta = I_{\text{Leakage}}/I_{\text{U}}$ is the ratio between the leakage current and the normalizing current. In order to verify the result we compare (37) with circuit simulations from a 65 nm CMOS process. An ideal 1 μA current source was used as the normalizing current.

In order to achieve higher leakage currents in the circuit level we artificially add current sources between the drain and bulk terminal of the transistors. In order to be able to connect these blocks directly, we use nMOS-based differential pairs and a pMOS-based Gilbert multiplier. We apply the same voltage to both inputs and calculated the natural logarithm of the ratio between the output currents from the Gilbert multiplier. For conversion between voltages and LLRs we use a value of the sub-threshold swing parameter n which best matches the circuit results. The simulation results are presented in Fig. 10. The figure shows the close match between the circuit simulations from Cadence and the analytical results for three different values of β. Using (37) we can derive the maximum output LLR for different values of β. Larger values of β represent nano-scale CMOS technologies. One way to avoid large β values is to use a higher normalizing current source. The drawback is that this leads to higher power consumption. Also, very big currents force the circuit to move from the sub-threshold region to strong inversion and reduces the accuracy of the circuit.

8. The effect of clipping on the sum-product algorithm

In order to study the effect of clipping on large codes we use density evolution (DE). DE has previously proved to be useful in predicting the code threshold and the average behavior of LDPC codes [4,3]. DE predicts the behavior of an infinitely large code but it is well known that even in finite sized codes, the LLRs behave closely to the DE results [4,30]. We modify DE to predict the performance of the decoder under limited LLR constraints. For each of eight code ensembles in Table 5, we clip the internal LLRs passing from variable nodes to check nodes, at three different values, chosen according to a beta of 0.001; 0.005 and 0.01. These ratios correspond to maximum LLR values of "IBound" = 6.90 5.29 and 4.6. When the internal LLRs are clipped, we cannot allow large input LLRs from the channel which could otherwise override any internal computation and cause high error floors. In other words the inputs from the channel should also be bounded so that the small check node outputs would be able to correct any channel LLR value. Therefore, we also clip the channel input LLRs at LLR value of CBnd, where "CBnd" is optimized for best performance in each case and is given in Table 5. Two thousand iterations are run at the target bit error rate of 10^{-8}. The results of this simulation are shown in Table 5. The term "Th" is the new threshold value for the code with LLR limitations. The deviation of the code from its original threshold is an indicator of the performance loss. The results of this simulation are shown in Table 5.

Table 5
DE analysis of clipping in LDPC codes.

Code ensemble/threshold	IBound: 6.9		IBound: 5.29		IBound: 4.6	
	CBnd	Th	CBnd	Th	CBnd	Th
(5,10)-2.008	7.00	2.008	5.30	2.028	4.7	2.090
(4,8)-1.538	6.95	1.538	6.00	1.557	4.6	1.615
(3,4)-0.957	8.00	0.957	5.29	0.960	4.65	0.975
(4,10)-1.729	7.00	1.729	5.29	1.769	–	–
(3,5)-0.888	7.80	0.888	5.35	0.903	–	–
(3,6)-1.101	7.00	1.102	5.30	1.135	–	–
(3,9)-1.747	7.00	1.752	–	–	–	–
(3,15)-2.586	–	–	–	–	–	–

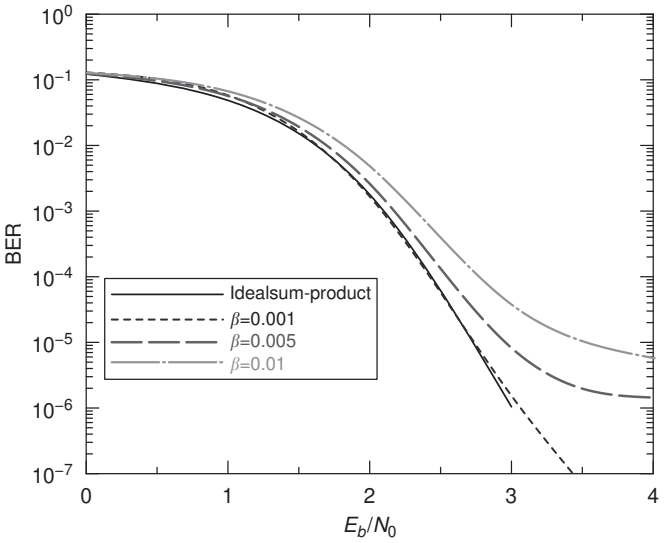

Fig. 11. The effect of hard clipping the internal LLRs in a (3,6)-regular length-816 code.

The "–" sign indicates that the error did not go asymptotically to zero even at very high threshold values. Hence we cannot define a threshold value for the code.

To further verify the results, Monte-Carlo simulations are carried out for different clipping scenarios. A moderate size (3,6)-regular length-816 code is used and runs for 50 iterations. The internal and channel LLRs are clipped according to Table 5. The results confirm the outcome from the DE analysis (see Fig. 11).

The graph confirms the DE results that clipping values of less than about 7 result in serious degradation of performance.

In order to make the simulations closer to the actual circuit behavior, we run the Monte-Carlo simulations using (37) for $\beta = 0.001, 0.005$ and 0.01. Fig. 12 shows the simulation results for the (3,6)-regular code. The leakage-included curves show worse performance compared to simple clipping of the LLRs.

Our simulations show that for values of β smaller than 0.001 the effect of LLR clipping is minor for most code ensembles up to bit error rate of 10^{-6}. Therefore as long as we can control the

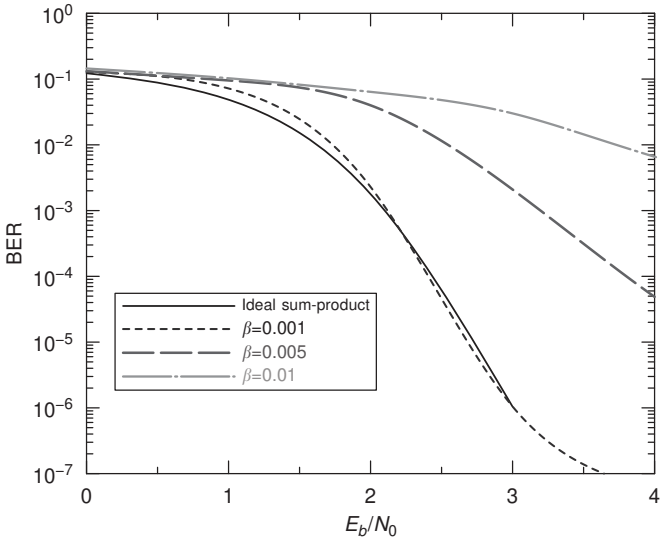

Fig. 12. The effect of LLR limitation due to leakage currents in a (3,6)-regular length-816 code, evaluated using (37).

leakage currents or use a large enough normalizing current source, the effect of clipping can be controlled.

9. Thermal noise

The variance of the drain current noise in weak inversion is given by [37]

$$\sigma_{I_n}^2 = 2qI_{DS}(1+e^{-V_{DS}/V_T})\Delta f, \tag{38}$$

where q is the electron charge. In order to calculate the maximum value of $\sigma_{I_n}^2$, we need to know the current I_{DS} and the bandwidth Δf. The maximum current in the decoder is the normalizing current I_U. A reasonable value for the normalizing current that we used in this paper is 1 μA. Using the average convergence speed of the designed nodes, the average bandwidth is 20 MHz. From these values σ_{I_n} is 2.53 nA. Such a change in current can be caused by a threshold variation with $\sigma_{V_t} = 0.063$ mV which according to mismatch analysis is negligible. Therefore we can ignore the effect of thermal noise.

10. Convergence speed

Unlike discrete time decoders, analog decoding happens asynchronously. The information adds up until the decoder converges. Hemati et al. have previously studied this problem using a methods of relaxation in a Monte-Carlo simulation setting [44,45]. In this paper, we use Gaussian DE to predict the speed of analog decoders. In an analog LDPC decoder, part of the delays stem from the interconnects which connect variable node processors to check nodes using long on-chip wires and part of the delay is due to the internal node delays. The node delays are a function of transistor sizing and are mainly constrained by mismatch sizing rules. The interconnect part of the delay is a function of the number computational nodes, N, in the decoder. Assuming these nodes are densely-packed on a square grid, the length of one side scales with $\sqrt{N}$. In a random code, since wires should be randomly-placed, average wire length scales with $\sqrt{N}$. Both the resistance and the capacitance of a wire are proportional to the wire length. Hence, the delay associated with these interconnects is proportional to N.

The total delays in the circuit can be modeled with a first order $\tau = RC$ delay. Similar concept has been explained in more details in [45]. Eq. (39) shows the change in the output after Δ_t using the RC delay model.

$$\text{Output}(t_0+\Delta_t) = \text{Output}(t_0) + (\text{Input}(t_0) - \text{Output}(t_0))(1-e^{-\Delta_t/\tau}), \tag{39}$$

where Output(t_0) is the value of Output before changing the input and Input(t_0) is value of the input step function at t_0. Using this model for both a check node and a variable node and assuming the inputs from the channel are already present, we have

$$\text{LLR}_{v\to c}(i\Delta_t) = \frac{2}{\sigma^2}(1-e^{-i\Delta_t/\tau}) + (1-e^{-\Delta_t/\tau})(\text{dv}-1)\sum_{j=2}^{i}\text{LLR}_{c\to v}(j\Delta_t)(e^{-(i-j)\Delta_t/\tau}) \tag{40}$$

$$\text{LLR}_{c\to v}((i+1)\Delta_t) = (1-e^{-\Delta_t/\tau})\sum_{j=1}^{i}\mu(\text{LLR}_{v\to c}(j\Delta_t))(e^{-(i-j)\Delta_t/\tau}), \tag{41}$$

where i, j are integer numbers larger than 0, $2/\sigma^2 = -4/N_0$ is the mean of the probability density function of the log-likelihood messages at the receiver in a Gaussian channel with noise variance $\sigma_{\text{noise}}^2 = N_0/2$, $\text{LLR}_{v\to c}(t)$ is the message sent from a variable node to check nodes at time instance t, $\text{LLR}_{c\to v}(t)$ is the message sent from a check node to variable nodes at time instance t and $\mu(x)$ represents the check node operation from [59, p. 270].

Using this delay-based DE we can compare the convergence time between discrete (synchronous) decoders and continuous (asynchronous) decoders. Fig. 13 shows DE results for both cases at $E_b/N_0 = 1.17$ dB. We used the parameter $\Delta_t/\tau = 1/5$ for the continuous case. The simulations show that the analog DE requires approximately $2\times\tau\times$ number of discrete, Δ_t steps to achieve the same LLR mean.

It is interesting to notice that as we increase the signal to noise ratio and move further from the threshold, both the required number of discrete iterations and continuous DE decrease while their ratio remains constant. Simulations show that for $E_b/N_0 = 1.27$ we require 37 discrete iterations, 22 for $E_b/N_0 = 1.47$ and 13 for $E_b/N_0 = 1.97$ and an analog decoder will correspondingly converge faster at higher signal-to-noise ratios. Many of the state of the art digital implementations of LDPC decoders use between 8 to 15 iterations [60–62]. Therefore based on the DE analysis an analog decoder requires $2\times\tau\times 15$ seconds for decoding a frame.

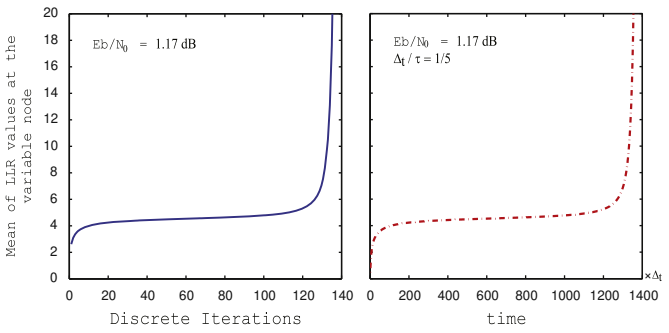

Fig. 13. DE analysis of a (3,6)-regular LDPC code under Gaussian channel assumption.

In order to estimate this value, we used post-layout simulation results from our implemented LDPC nodes. We simulated a three input variable node, driving a six input check node. The connection between the variable node and check node is through a 1 mm wire. The choice of wire length was based on parallel decoder architectures found in the literature for practical codes [10,21,63].

Using post-layout simulations we calculated the average value of τ. This number is $\tau = 0.0258\,\mu s$ for small [Each Transistor $= 0.14(\mu m)^2$] nMOS based 90 nm node and equals, $\tau = 0.523\,\mu s$ for the large [Each Transistor $= 3.90(\mu m)^2$] nMOS based circuit in 65 nm. These values translate to throughputs between $N \times 1.3$ Mb/s to $N \times 64$ kb/s. The node delay depends on many different factors such as transistor sizing, wire length, normalizing current and node architecture. Larger nodes are immune against mismatch but have slower response. Increasing the normalizing current reduces the convergence time but increases the power consumption. In addition, different node architectures result in different convergence times. While the dependence of accuracy and speed stems from many design parameters, using the faster designs discussed in this paper, and tolerating higher mismatch values, energy per bit consumption values of less than 10 pJ/bit can be computed as achievable.

11. Conclusion

In this paper we studied different phenomena that affect the scaling of sub-threshold analog decoders down to the nanometer regime. Here we focused on LDPC decoders, but the approach should be applicable to any analog decoder (e.g. Turbo or Turbo Product decoder) that employs Gilbert Multipliers. The effect of different short-channel effects as well as leakage currents and mismatch on sub-threshold analog decoders was discussed in detail. We showed that mismatch has the largest impact and imposes a lower limit on transistor sizing. Hence, process miniaturization is not likely to be beneficial for larger codes, as it results in approximately the same size decoders with the same speed and energy per bit values. That is, potential benefits of sub-threshold analog decoding flatten out around the 90 nm technology, as the value of A_Δ does not scale anymore. However, small code sizes can still benefit from scaling in CMOS technologies. Since the threshold mismatch effect is more severe in the sub-threshold region, moving to strong inversion is likely to alleviate this issue. Here we note the work in analog min-sum decoders [21], where transistors operate in strong inversion, and where scaling may be easier to achieve. Further investigation of this issue would be required.

Another possible solution is a hybrid analog-digital decoder, where the analog decoder would perform a fast, power-efficient initial decoding phase, and after some point an "accurate" digital decoder would take over to correct any residual errors. Since most of the incorrect bits have already been corrected by the analog decoder, the dynamic power consumption of the digital decoder is reduced. These digitally assisted solutions require further investigation and are still at a preliminary stage.

References

[1] R.G. Gallager, Low-density parity-check codes, IRE Transactions on Information Theory IT-8 (1962) 21–28.

[2] D.J.C. MacKay, R.M. Neal, Near Shannon limit performance of low density parity check codes, IEEE Electronics Letters 33 (1997) 457–458.

[3] S.Y. Chung, G.D. Forney Jr., T.J. Richardson, R. Urbanke, On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit, IEEE Communications Letters 5 (2001) 58–60.

[4] T. Richardson, R. Urbanke, The capacity of low-density parity check codes under message-passing decoding, IEEE Transactions on Information Theory 47 (2001) 599–618.

[5] F.R. Kschischang, B.J. Frey, H.A. Loeliger, Factor graphs and the sum-product algorithm, IEEE Transactions on Information Theory 47 (2001) 498–519.

[6] R.M. Tanner, A recursive approach to low complexity codes, IEEE Transactions on Information Theory 27 (1981) 533–547.

[7] J. Hagenauer, M. Winklhofer, The Analog decoder, in: IEEE International Symposium on Information Theory, 1998, p. 145.

[8] H.A. Loeliger, F. Lustenberger, M. Helfenstein, F. Tarköy, Probability propagation and decoding in analog VLSI, in: IEEE International Symposium on Information Theory, 1998, p. 146.

[9] H.A. Loeliger, F. Lustenberger, M. Helfenstein, F. Tarköy, Probability propagation and decoding in analog VLSI, IEEE Transactions on Information Theory 46 (2001) 837–843.

[10] A.J. Blanksby, C.J. Howland, A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder, IEEE Journal of Solid-State Circuits 37 (2002) 404–412.

[11] T. Brandon, R. Hang, G. Block, V.C. Gaudet, B. Cockburn, S. Howard, C. Giasson, K. Boyle, P. Goud, S. Sheikh Zeinoddin, A. Rapley, S. Bates, D. Elliott, C. Schlegel, A scalable LDPC decoder ASIC architecture with bit-serial message exchange, Integration-the VLSI Journal 41 (2008) 385–398.

[12] A. Darabiha, A. Chan Carusone, F.R. Kschischang, A 3.3 Gbps bit-serial block-interlaced LDPC decoder in 0.13 μm CMOS, in: IEEE Custom Integrated Circuits Conference, 2007, pp. 459–462.

[13] V.C. Gaudet, A.C. Rapley, Iterative decoding using stochastic computation, IEEE Electronics Letters 39 (2003) 299–301.

[14] S. Sharifi Tehrani, W.J. Gross, S. Mannor, Stochastic decoding of LDPC codes, IEEE Communications Letters 10 (2006) 716–718.

[15] M. Moerz, T. Gabara, R. Yan, J. Hagenauer, An analog 0.25 μm BiCMOS tailbiting MAP decoder in: IEEE International Solid-State Circuits Conference: Digest of Technical Papers, 2000, pp. 356–357.

[16] C. Lahuec, G. Le Mestre, F. Seguin, M. Arzel, M. Jézéquel, Design and test of a 0.25 μm−BiCMOS double-binary analogue APP Decoder, in: 5th Analogue Decoding Workshop, Torino, Italy, 2006, pp. 35–38.

[17] C. Winstead, J. Dai, S. Yu, C. Myers, R.R. Harrison, C. Schlegel, CMOS analog MAP decoder for (8,4) Hamming code, IEEE Journal of Solid-State Circuits 39 (2004) 122–131.

[18] D. Vogrig, A. Gerosa, A. Neviani, A.Gi. Amat, G. Montorsi, S. Benedetto, A 0.35-μm CMOS analog turbo decoder for the 40-bit rate 1/3 UMTS channel code, IEEE Journal of Solid State Circuits 40 (2005) 753–762.

[19] V.C. Gaudet, P.G. Gulak, A 13.3-Mb/s 0.35-μm CMOS analog turbo decoder IC with a configurable interleaver, IEEE Journal of Solid-State Circuits 38 (2003) 2010–2015.

[20] M. Arzel, C. Lahuec, F. Seguin, D. Gnaedig, M. Jézéquel, Semi-iterative analog turbo decoding, IEEE Transactions on Circuits and Systems—I: Regular Papers 54 (2007) 1305–1316.

[21] S. Hemati, A.H. Banihashemi, C. Plett, A 0.18 μm CMOS analog min-sum iterative decoder for a (32,8) low-density parity-check (LDPC) code, IEEE Journal of Solid State Circuits 41 (2006) 2531–2540.

[22] M. Frey, On analog decoders and digitally corrected converters, Ph.D. Thesis, ETH Zurich, No. 16536, 2006.

[23] P. Merkli, Message-passing algorithms and analog electronic circuits, Ph.D. Thesis, ETH Zurich, No. 15942, 2005.

[24] A.F. Mondragon-Torres, E. Sanchez-Sinencio, K.R. Narayanan, Floating-gate analog implementation of the additive soft-input soft-output decoding algorithm, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 50 (2003) 1256–1269.

[25] F. Gioulekas, M. Birbas, A. Birbas, G. Bilionis, A high-speed analog turbo decoder with low-energy consumption, IEEE International Symposium on Communications and Information Technology 1 (2004) 33–38.

[26] M. Mörz, Analog signal processing in forward error correction (FEC) decoders, Ph.D. Thesis, Munich University of Technology, 2007.

[27] C. Kong, S. Chakrabartty, Analog iterative LDPC decoder based on margin propagation, IEEE Transactions on Circuits and Systems—II: Express Briefs 54 (2007) 1140–1144.

[28] S. Yang, D. Li, Y. Qiu, The decoder of trellis code implemented by CMOS analog circuits, in: IEEE 7th International Conference on ASIC, 2007 pp. 898–901.

[29] M.V. Dunga, W. Yang, X. Xi, J. He, W. Liu, Kanyu, M. Cao, X. Jin, J. Ou, M. Chan, A. Niknejad, C. Hu, BSIM4.6.1 MOSFET Model User's Manual, Department of EECS University of California, Berkeley, 2007.

[30] T. Richardson, A. Shokrollahi, R. Urbanke, Design of capacity-approaching irregular low-density parity-check codes, IEEE Transactions on Information Theory 47 (2001) 619–637.

[31] F. Lustenberger, H.A. Loeliger, On mismatch errors in analog-VLSI error correcting decoders, IEEE International Symposium on Circuits and Systems 4 (2001) 198–201.

[32] M.J.M. Pelgrom, Matching properties of MOS transistors, IEEE Journal of Solid-State Circuits 24 (1989) 1433–1440.

[33] M. Frey, H.A. Loeliger, F. Lustenberger, P. Merkli, P. Strebel, Analog decoder experiments with subthreshold CMOS soft-gates, in: IEEE International Symposium on Circuits and Systems, 2003, pp. 85–88.

[34] C. Winstead, C. Schlegel, Density evolution analysis of device mismatch in analog decoders, in: IEEE International Symposium on Information Theory, 2004, p. 293.

[35] J. Dai, Design methodology for analog VLSI implementations of error control decoders, Ph.D. Thesis, University of Utah, 2002.

[36] B. Gilbert, A precise four-quadrant multiplier with subnanosecond response, IEEE Journal of Solid-State Circuits 3 (1968) 365–373.

[37] Y. Tsividis, Operation and Modeling of The MOS Transistor, second ed., Oxford University Press, NY, 2003.

[38] K. Cao, W. Liu, X. Jin, V. Green, K. Krick, J. Vrotsos, T. Chenming Hu, Modeling of pocket implanted MOSFETs for anomalous analog behavior, in: International IEDM Technical Digest on Electron Devices Meeting, 1999, pp. 171–174.
[39] T. Ytterdal, Y. Cheng, T.A. Fjeldly, Device Modeling for Analog and RF CMOS Circuit Design, John Wiley and Sons Ltd., England, 2003.
[40] K. Roy, S. Mukhopadhyay, H. Mahmoodi-Meimand, Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits, Proceedings of the IEEE 91 (2003) 305–327.
[41] J.A. Croon, W.M. Sansen, H.E. Maes, Matching Properties of Deep Sub-Micron MOS Transistors, Springer, The Netherlands, 2005.
[42] J. Pérez Chamorro, C. Lahuec, F. Seguin, M. Jézéquel, Design rules for subthreshold MOS circuits, in: 5th Analogue Decoding Workshop.
[43] P.G. Drennan, M.L. Kniffin, D.R. Locascio, Implications of proximity effects for analog design, in: Custom Integrated Circuits Conference, 2006, pp. 169–176.
[44] S. Hemati, A.H. Banihashemi, Convergence speed and throughput of analog decoders, IEEE Transactions on Communications 55 (2007) 833–836.
[45] S. Hemati, A.H. Banihashemi, Dynamics and performance analysis of analog iterative decoding for low-density parity-check (LDPC) codes, IEEE Transactions on Communications 54 (1) (2006) 61–70.
[46] F. Lustenberg, On the design of analog iterative VLSI decoders, Ph.D. Thesis, ETH Zurich, No. 13879, 2000.
[47] D. MacKay, Encyclopedia of sparse graph codes, 2007 〈http://www.inference.phy.cam.ac.uk/mackay/codes/data.html〉.
[48] T. Mizuno, J. Okumtura, A. Toriumi, Experimental study of threshold voltage fluctuation due to statistical variation of channel dopant number in MOSFET's, IEEE Transactions on Electron Devices 41 (1994) 2216–2221.
[49] P.G. Drennan, C.C. McAndrew, Understanding MOSFET mismatch for analog design, IEEE Journal of Solid-State Circuits 38 (2003) 450–456.
[50] R. Difrenza, P. Llinares, G. Ghibaudo, The impact of short channel and quantum effects on the MOS transistor mismatch, Solid-State Electronics 47 (2003) 1161–1165.
[51] R. Difrenza, P. Llinares, G. Ghibaudo, E. Robillart, E.Granger, Dependence of channel width and length on MOSFET matching for 0.18 μm CMOS technology, in: Proceedings of European Solid-State Device Research Conference, 2000, pp. 584–587.
[52] J.P. de Gyvez, H.P. Tuinhout, Threshold voltage mismatch and intra-die leakage current in digital CMOS circuits, IEEE Journal of Solid-State Circuits 39 (2004) 157–168.
[53] R.J. van de Plassche, J.H. Huijsing, W.M. Sansen, Analog Circuit Design: High-Speed Analog-to-Digital Converters, Mixed Signal Design; PLLs and Synthesizers, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.
[54] M.J.M. Pelgrom, H.P. Tuinhout, M. Vertregt, Transistor matching in analog CMOS applications, in: International Electron Device Meeting Technical Digest, 1998, pp. 915–918.
[55] A. Pavasovic, A.G. Andreou, C.R. Westgate, Characterization of subthreshold MOS mismatch in transistors for VLSI systems, Journal of VLSI Signal Processing 8 (1994) 75–85.
[56] M. Frey, H.A. Loeliger, F. Lustenberger, P. Merkli, P. Strebel, Measurements on an analog (8,4,4) hamming code decoder chip, Technical Report No. 200402, Signal and Information Processing Laboratory, ETH Zurich, 2004.
[57] S. Chih-Tang, R.N. Noyce, W. Shockley, Carrier generation and recombination in P-N junctions and P-N junction characteristics, Proceedings of the IRE 45 (1957) 1228–1243.
[58] S. Mukhopadhyay, A. Raychowdhury, K. Roy, Accurate estimation of total leakage in nanometer-scale bulk CMOS circuits based on device geometry and doping profile, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 24 (2005) 363–381.
[59] C. Schlegel, L. Perez, Trellis and Turbo Coding, IEEE press, NJ, USA, 2004.
[60] A. Darabiha, A. Chan Carusone, F.R. Kschischang, Power reduction techniques for LDPC decoders, IEEE Journal of Solid-State Circuits 43 (2008) 1835–1845.
[61] M. Mansour, N. Shanbhag, A 640-Mb/s 2048-Bit Programmable LDPC Decoder Chip, IEEE Journal of Solid State Circuits 41 (2006) 684–689.
[62] T. Mohsenin, B.M. Baas, Split-Row: a reduced complexity, high throughput LDPC decoder architecture, in: IEEE International Conference on Computer Design, 2006, pp. 320–325.
[63] E. Liao, E. Yeo, B. Nikolic, Low-density parity-check code constructions for hardware implementation, in: IEEE International Conference on Communications, vol. 5, 2004, pp. 2573–2577.