

Iterative Decoding in Analog CMOS

Saied Hemati and Amir H. Banihashemi¹

Broadband Communications and Wireless Systems (BCWS) Centre
Ottawa-Carleton Institute for Electrical and Computer Engineering (OCIECE)
Department of Systems and Computer Engineering, Carleton University
1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6
Tel: +1 613 520-2600 ext. 8026, Fax: +1 613 520-5727
{hemati, ahashemi}@sce.carleton.ca

ABSTRACT

In this paper, a novel current-mode approach is proposed for implementing basic building blocks of an analog iterative decoder. The decoder is based on the so-called min-sum algorithm (also referred to as max-sum or max-product) and can be used to decode powerful coding schemes such as low-density parity-check (LDPC) codes and turbo codes. The proposed circuits can be implemented by standard CMOS technology, which means lower fabrication cost and/or simpler design compared to previously reported analog iterative decoders that are based on BiCMOS or sub-threshold CMOS technology. To demonstrate the functionality of the proposed design, simulation results based on TSMC 0.18 μ m CMOS technology for a (7,4) Hamming code are also presented.

Categories and Subject Descriptors

B.7.1 [Integrated Circuit]: Types and Design Styles – *algorithms implemented in hardware, VLSI.*

General Terms

Design

Keywords

Analog Circuit, Analog CMOS, Iterative Decoding, Low-Density Parity-Check Codes, Turbo codes, Min-Sum Decoding, Soft Decoding, Asynchronous Iterative Decoding, Analog Iterative Decoder.

1. INTRODUCTION

In the recent years, there has been much excitement in digital communications and coding community caused by a class of error correcting codes which are very powerful and yet can be decoded practically using iterative algorithms. Two well-known examples are turbo codes [1] and low-

density parity-check (LDPC) codes [4]. The iterative algorithms can be most naturally described using graph representations of the codes [3,11]. The execution of the algorithms can be described as performing certain computations at the nodes of the graph and passing real messages (soft information) along the edges, in both directions and iteratively. There are only a small number of different types of nodes in terms of their computations. In fact, this number is only two and three for LDPC codes and turbo codes, respectively.

The need for floating point computations and the iterative nature of these decoding algorithms have motivated some very recent research on the analog electronic implementation of the algorithms [6,9,12]. In analog implementation, each node in the graph acts as a computational module which communicates asynchronously with other nodes of the graph through edges. Iterations are thus eliminated and are replaced by the settling behavior of the system. This is projected to improve the ratio of speed to power consumption by two orders of magnitude [9].

There are a wide variety of iterative decoding algorithms, each resulting in a different performance/complexity tradeoff. Among these, *sum-product* (SP), also referred to as *belief propagation*, performs the best but considered to be the most complex to implement. In its original form, SP requires basic operations of addition and multiplication of real numbers [3], and thus the name “sum-product.” While implementing adders is rather straightforward and does not require much power and area in an integrated circuit, implementing a large number of high precision digital multipliers is more challenging. In fact, this is one of the reasons of shifting from digital to analog for implementing iterative decoders. In analog circuits, Gilbert differential multiplier is a well-known solution for implementing multiplication and is widely used in mixer and PLL circuits [5]. In the Gilbert multiplier, bipolar transistors play a key role. In fact, all the previously reported analog decoders for iterative schemes have been based on SP algorithm, have used the Gilbert multiplier as the core circuit, and have mostly been implemented by BiCMOS technology [6,9]. BiCMOS technology has the advantage of using fast

¹ This work has been supported in part by Communication and Information Technology Ontario (CITO).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GLSVLSI'03, April 28-29, 2003, Washington, DC, USA.
Copyright 2003 ACM 1-58113-677-3/03/0004...\$5.00.

bipolar transistors together with low power and small size MOSFETs. However, it has a more complex fabrication process, which translates to higher cost, compared to standard CMOS technology.

There have been also some attempts for implementing SP with MOSFETs in weak inversion mode (sub-threshold region) where transistors are partially turned on [12], and behave like slow bipolar transistors [7]. Sub-threshold circuits in general have the advantage of consuming very small amount of power, but often suffer from low speed, mismatch problems and low accuracy [7].

In this work, we show that iterative analog decoders can be implemented using the favorable full CMOS technology. For this, we focus on another iterative decoding algorithm, called “min-sum” (MS) [3], which can be regarded as an approximation of SP. Other common names for min-sum algorithm are “max-sum” and “max-product.” Although, for many codes, the performance of MS is slightly (a few tenths of a dB) worse than that of SP, it has lower complexity, and unlike SP, does not require an estimate of noise power. More recently, it has also been shown that MS is more robust against quantization noise than SP [13]. In fact, MS can be implemented digitally by using only 4 bits compared to 6 bits for a similar implementation of SP [13]. Moreover, there are simple modifications of min-sum that can perform very close to sum-product algorithm [13], [2]. Because of imperfections in the fabrication process, the problem of mismatch in large analog integrated circuits becomes very important and the higher robustness in min-sum can also greatly help in mitigating this problem.

It is worth emphasizing that all previously reported analog decoders fail to use conventional CMOS technology or conventional biasing methods. They are either BiCMOS designs or sub-threshold CMOS designs. This leads to an increase in overall cost or low speed operation. Also, mismatch problem is more severe in sub-threshold circuits. The main advantage of our circuits is that they can be implemented by standard CMOS technology and use conventional biasing methods, which makes them more practical and affordable.

In addition, all modules in previous designs are based on Gilbert multiplier and each block can just have two inputs. Modules with more than two inputs can then be constructed by cascading the two-input blocks. This increases the processing delay. In the proposed circuits however, modules with large number of inputs can be fabricated easily and simulations show that increasing the number of inputs does not increase the delay as much.

In the rest of the paper, we first explain min-sum algorithm briefly. We then propose current-mode CMOS circuits for implementing the computational modules of MS. Finally, we present simulation results for a (7,4) Hamming code to demonstrate the functionality of our circuits.

2. MIN-SUM DECODING ALGORITHM

To explain the min-sum algorithm, we focus on binary LDPC codes and the bipartite graph that represents the code through its parity-check equations, and is referred to as “Tanner graph” [11]. There are two sets of nodes in a Tanner graph: 1) variable nodes representing code bits, and 2) check nodes representing parity-check equations. To implement the min-sum algorithm in such a graph, two types of computational modules are required: one for variable nodes (VAR) and the other for check nodes (CHK). The operations performed in VAR and CHK modules depend on the representation of soft information (messages that are passed among CHK and VAR nodes through edges). A message along an edge e , which connects a variable node v to a check node c , in general reflects the conditional probabilities of v being zero (p_0) or one (p_1), with $p_0 + p_1 = 1$. In this work, we choose log-likelihood ratio (LLR) as the message, i.e.,

$$M = \ln \left(\frac{p_0}{p_1} \right) \quad (1)$$

The operations in VAR and CHK modules will then be

$$VAR : M_{v \rightarrow c} = \sum_{i \in N(v) \setminus c} M_{i \rightarrow v} + M_v^{(0)} \quad (2)$$

$$CHK : M_{c \rightarrow v} = \prod_{i \in N(c) \setminus v} \text{sgn}(M_{i \rightarrow c}) \cdot \min_{i \in N(c) \setminus v} |M_{i \rightarrow c}| \quad (3)$$

where $M_v^{(0)}$ is the initial local message at variable node v , $N(\cdot)$ represents the set of neighboring nodes, and $\text{sgn}(\cdot)$ returns 1 or -1 depending on its argument being non-negative or negative, respectively. The initial message $M_v^{(0)}$ is computed by replacing the conditional probability density functions of the channel output y corresponding to v in (1). It can be shown that for the binary input AWGN channel, $M_v^{(0)}$ can in fact be simplified to the received value y [13].

In its conventional synchronous form, the algorithm starts by all the variable nodes passing $M_v^{(0)}$ to their neighboring check nodes. Then iterations start by each iteration consisting of all the check nodes passing messages to variable nodes according to (3), and subsequently all variable nodes sending messages to check nodes based on (2). At each iteration, the hard decision assignment for each bit v is decided based on

$$M = \sum_{i \in N(v)} M_{i \rightarrow v} + M_v^{(0)} \quad (4)$$

and by

$$\hat{v} = \frac{1 - \text{sgn}(M)}{2} \quad (5)$$

The algorithm then stops if the assignment is a codeword, or a maximum number of iterations is reached.

3. IMPLEMENTATION OF CHK AND VAR MODULES

To implement min-sum algorithm, one should first design VAR and CHK modules, and construct a reliable cell library. Any min-sum decoder can then be implemented by simply selecting suitable VAR and CHK modules from the cell library and making connections among these modules according to the graphical representation of the code.

In this section, for the sake of simplicity, only the basic structure of VAR and CHK modules are described. In the actual implementation, more sophisticated circuits, with higher accuracy have been used. For instance, cascode configuration is used for current conveyers to improve the accuracy and overcome the short channel effects [7]. Also, wide swing current mirrors are used for implementing fixed current sources [7], and to reduce the on resistance in analog switches simple pass transistors have been replaced by transmission gates [10]. Moreover, in our simulations we have currently ignored the mismatch problem.

3-1 CHK Modules

In order to reduce the complexity of the parity check modules (CHK), we use two new variables W and X to represent the message M by its absolute value and sign:

$$W = |M|, \quad X = \frac{1 - \text{sgn}(M)}{2} \quad (6)$$

Clearly X is a binary variable, and X=0 and 1 correspond to positive and negative signs, respectively. Using W and X, to implement the CHK computation in (3), we only need to implement min(.) function that computes the minimum value of some non-negative signals, and XOR function of X variables. Fig. 1 shows a minimum Winner-Take-All circuit. It is actually a modified version of a well-known maximum Winner-Take-All circuit (max WTA) which is shown in Fig. 2 [8]. To convert the max WTA to min WTA, we have first subtracted the input currents ($I_{in(1)}$, $I_{in(2)}$, ..., $I_{in(m)}$) from a reference constant current (I_{bias}) which should be always greater than the minimum input current (it is easy to show that if an input current is greater than I_{bias} , it can not affect the output). Then these currents are conducted to the max WTA circuit given in Fig. 2 and the maximum input current is reproduced at the output. By subtracting the output of max WTA circuit from I_{bias} , the minimum input current is obtained.

Unfortunately, for deep submicron MOSFETs, short channel effects degrade the accuracy of the max WTA circuit in Fig. 2 and consequently the accuracy of the min WTA will be degraded too. This is due to the fact that short channel effects degrade the output impedance of MOSFETs. For many applications, it is possible to improve the output impedance and consequently the accuracy by

using cascode configurations. However, cascode configurations usually decrease the swing voltage in inputs and output. This might not be tolerable if a low voltage power supply is to be used. As an example, in 0.13 μ m CMOS technology, supply voltage can not be greater than 1.3 volts and conventional cascode configurations could be unacceptable. However, for 0.18 μ m CMOS technology, since supply voltage is not that small, cascode configurations are quite practical and for our simulation purposes in this paper, we have used conventional cascode circuits satisfactorily.

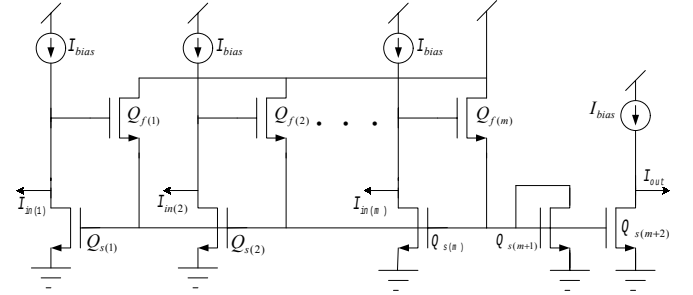


Figure 1. A current-mode min WTA circuit (I_{bias} is a constant current greater than the minimum input current $I_{in(1)}$, ..., $I_{in(m)}$. Also, $I_{in(1)} = C W_1$, ..., $I_{in(m)} = C W_m$, where C is a constant (in our circuit $C=50$ nA)).

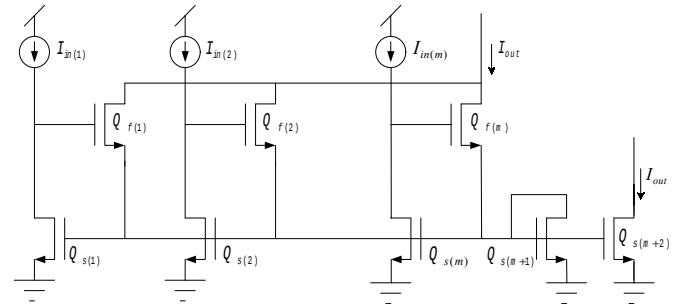


Figure 2. A basic max WTA circuit [8]. $I_{in(1)}$, ..., $I_{in(m)}$ are input currents and I_{out} is equal to the maximum input current.

For implementing the XOR function, there are many different logic families that can be used. We here used Differential Cascade Voltage Switch Logic (DCVSL) XOR gates [10]. Structure of a simple 2-input DCVSL XOR is shown in Fig. 3.

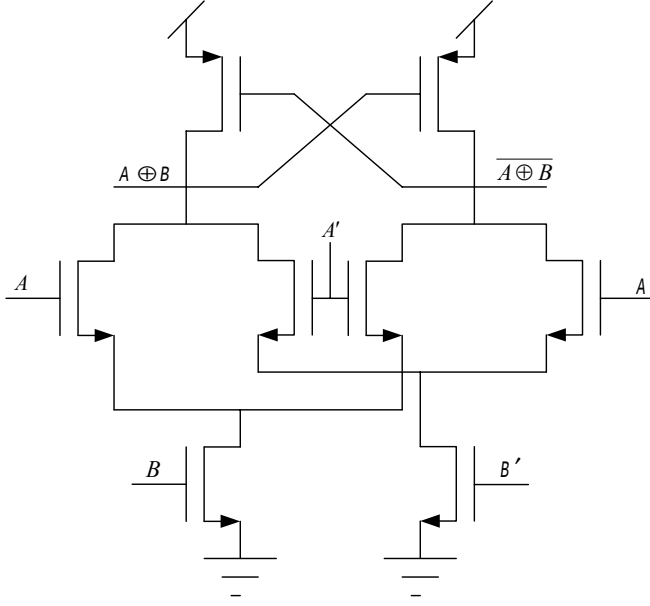


Figure 3. A 2-input DCVSL XOR gate.

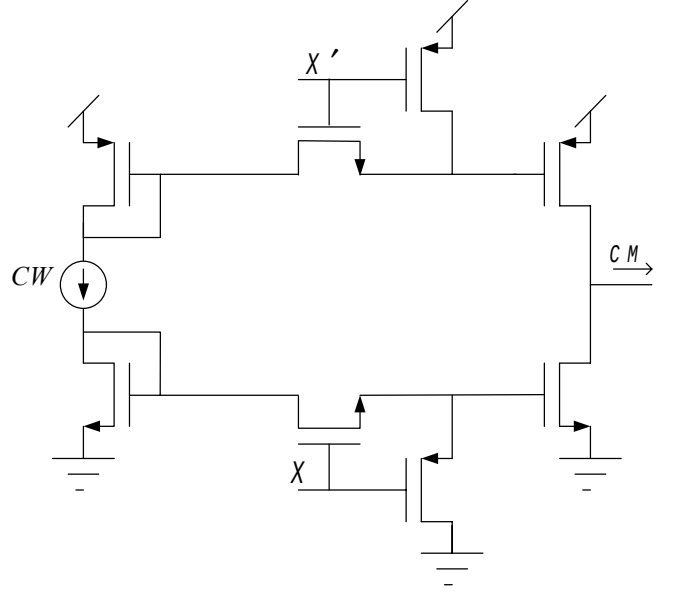


Figure 4. A circuit for obtaining M from W and X.

3-2 VAR Modules

In VAR modules, it is easier to work directly with M and for this reason, first input pairs of (W, X) are converted back to messages $M = (-1)^X W$ using the circuit of Fig. 4. This circuit consists of two diode-connected transistors, with a drain current equal to W. Two simple analog switches (pass transistors) depending on the value of X connect the gate of the output transistors to either a gate of a diode-connected transistor at the input, to form a current mirror, or to a proper voltage to turn them off. Therefore the direction of the output current changes according to the value of X (for instance, if $X=0$, a current equal to CW leaves the circuit). Messages M are then added simply based on the Kirchoff current law and the output can be positive or negative. To feed the CHK modules, we need to obtain the absolute value and the sign of the output current. For the absolute value, a current rectifier consisting of p-channel and n-channel current mirrors is used (see Fig. 5). The rectifier gives the absolute value of the final current in a fixed direction. The sign of the output can also be computed, simply by checking the active current mirror as shown in Fig. 5.

In Figures 4 and 5, the constant C is a design parameter that scales W and M. Parameter C can be changed for optimizing speed and/or power consumption and the dynamic range. In general, at lower currents, circuits dissipate less power but become slower. In our circuits, C is chosen 50 nA.

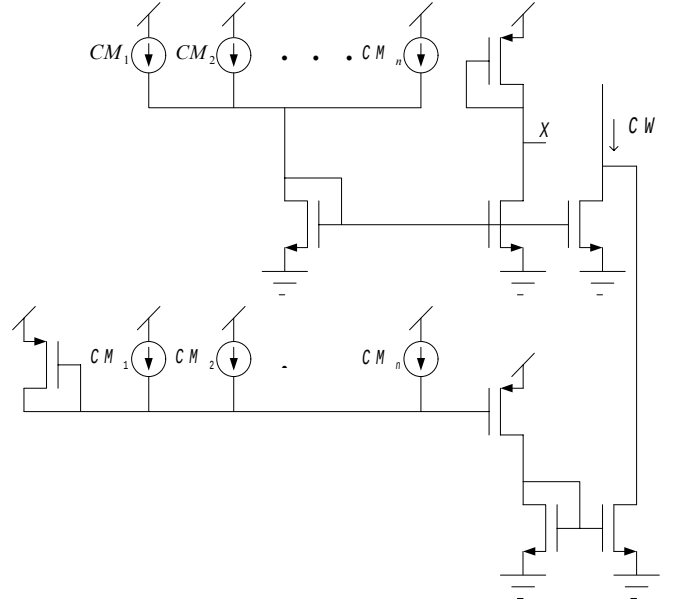


Figure 5. A current mode circuit for VAR module.

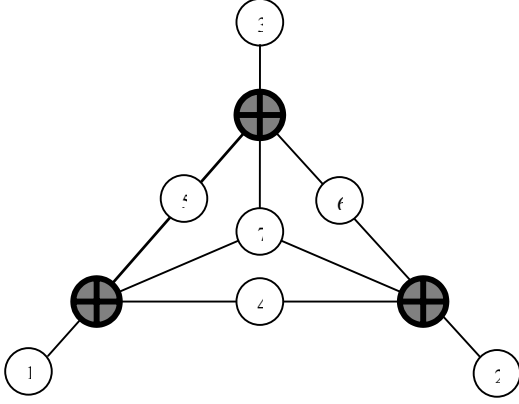


Figure 6. Tanner Graph of the (7,4) Hamming code. Variable nodes are distinguished by their index.

4. SIMULATION RESULTS

The functionality of the proposed circuits for implementing CHK and VAR modules is tested by simulating the MS decoder for a (7,4) Hamming code. As can be seen in the Tanner graph of the code in Fig. 6, the decoder has three CHK nodes with degree four (4-input CHK module) and seven VAR nodes with degrees 1, 2 and 3. To examine the functionality and the transient response of the decoder, as an example, we consider the following received block at the input of the decoder:

$$y = [1 \quad 2 \quad 0.5 \quad 1 \quad 2 \quad 2 \quad -1.7]$$

It is easy to see that the maximum-likelihood (ML) decoding, which finds the closest codeword to the received block, results in:

$$c = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$$

The same received block is now applied to our full CMOS min-sum decoder and the transient Cadence Spectre tools and TSMC 0.18 μ m CMOS technology are used for the simulation. Initial states of the edges in the decoder are all forced to a very small positive value, not to degrade the decoding.

Figures 7 and 8 show how absolute values of the outputs (W's) change, and Fig. 9 shows the variation of the sign bits (X's). As it can be seen, the final hard-decision assignment from Fig. 9, is in fact the ML codeword. It is interesting to note that after applying the inputs to the decoder, it takes about 3.5 μ s until the absolute values of the third, the forth and the seventh outputs reach zero, then the sign for these bits change and then the absolute values keep increasing until they settle down to their final values after about 10 μ s.

According to our simulations, these delays can vary substantially with signal-to-noise ratio (SNR). At higher SNRs, the delay can be even less than 0.1 μ s. While we used a small parameter C to keep the power dissipation low, it is possible to get much faster decoding by increasing C at the expense of higher power.

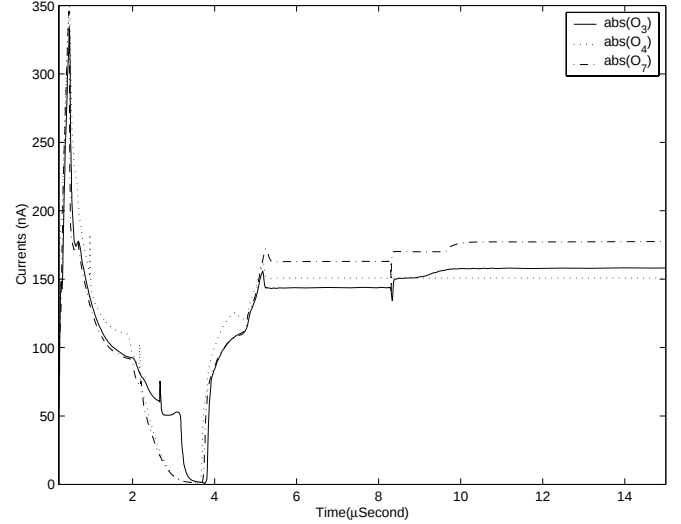


Figure 7. Variation of absolute values of outputs (W's) for 3rd, 4th and 7th bits.

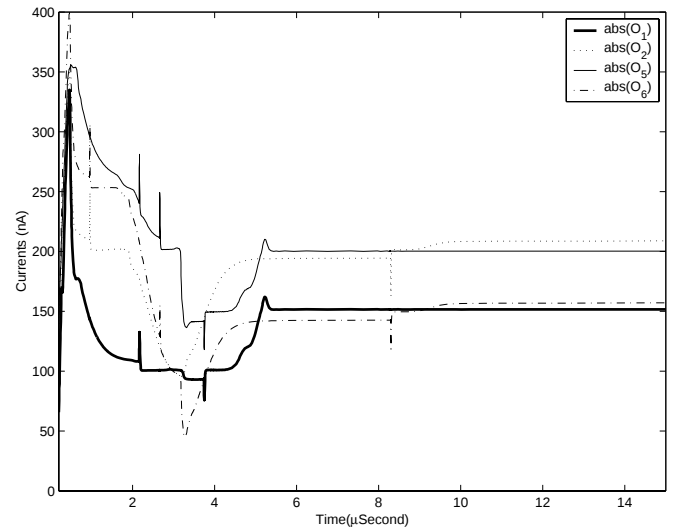


Figure 8. Variation of absolute values of outputs (W's) for 1st, 2nd, 5th and 6th bits.

5. CONCLUSION

Current-mode full CMOS circuits are proposed for implementing the min-sum decoder. The circuits can be

fabricated by standard CMOS technology. This means lower fabrication cost and simpler design compared to the previously reported analog iterative decoders. Although, for many codes, the performance of min-sum is slightly worse than that of sum-product algorithm, it has lower complexity and unlike sum-product, does not require an estimate of noise power. There are also very simple modifications that can be applied to min-sum algorithm [10, 11], and improve the performance more or less to that of sum-product. These modifications can be incorporated in the designed analog MS decoder with minimal effort. Moreover, higher robustness is another advantage of using min-sum for analog VLSI decoders, in which imperfections in the fabrication process lead to mismatch, and parameter variations due to the change of temperature is also a major problem.

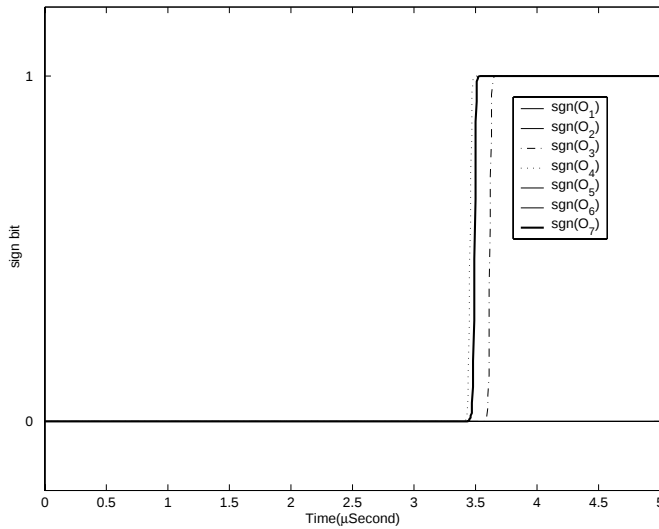


Figure 9. Variation of sign bits (X's) at the output.

Modules with large number of inputs can be fabricated easily and with less delay compared to previously reported analog iterative decoders. The proposed modules can also be used as building blocks for implementing the decoders for other iterative coding schemes than LDPC codes. These include turbo convolutional and block codes.

6. REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding: turbo codes," in *Proc. IEEE ICC'93*, Geneva, Switzerland, 1993, pp. 1064-1070.
- [2] J. Chen, and M. P. C. Fossoreir, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Comm. Lett.*, vol. 6, no.5, May 2002, pp. 208-210.
- [3] G. D. Forney Jr, "On iterative decoder and two-way algorithm," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, Sept. 1997, pp. 12-25.
- [4] R.G. Gallager, "Low-density parity check codes," *IRE Trans. Inform. Theory*, vol. 8, Jan. 1962, pp.21-28.
- [5] P.R. Gray, and R.G. Meyer, *Analysis and Design of Analog Integrated Circuits*, third edition, John Wiley & Sons, 1993.
- [6] J. Hagenauer, M. Moerz, and E. Offer, "Analog turbo networks in VLSI: the next step in turbo decoding and equalization," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, Sept. 2000, pp. 209-218.
- [7] D. A. Johns, and K. Martin, *Analog Integrated Circuit Design*, first edition, John Wiley & Sons, 1997.
- [8] J. Lazzaro, S. Ruckebusch, M.A. Mahowald, and C.A. Mead, "Winner-take-all networks of O(n) complexity," In *Advances in neural information processing systems*, volume 2, pages 703-711, San Mateo - CA, 1989. Morgan Kaufmann.
- [9] H. A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarkoy, "Probability propagation and decoding in analog VLSI," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, Feb. 2001, pp. 837-843.
- [10] J. Rabaey, *Digital Integrated Circuits*, first edition, Prentice Hall, 1996.
- [11] R. M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, Sept. 1981, pp. 533-547.
- [12] C. Winstead, J. Dai, S. Little, K. Yong-Bin, and W. Jin Kim, "Analog MAP decoder for (8,4) Hamming code in subthreshold CMOS," in *Proc. Advanced Research in VLSI conference*, 2001, pp. 132-147.
- [13] F. Zarkeshvari, and A. H. Banihashemi, "On implementation of min-sum algorithm for decoding low-density parity-check (LDPC) codes," in *IEEE Globecom Conf. 2002*, Taipei, Taiwan, Nov. 2002.