

# Equation Based LDPC Decoder for Intersymbol Interference Channels

Zining Wu and Gregory Burd

Marvell Semiconductor, 700 First Ave., Sunnyvale, CA 94089

{zwu,gburd}@marvell.com

**Abstract** - The application of LDPC code to intersymbol interference (ISI) channels requires efficient soft decoding methods for ISI channel as well as outer LDPC code. We introduce a fully pipelined turbo equalization architecture that combines soft decoders for channel and LDPC code. This equation based LDPC decoder uses signed sum product (SSP) decoding algorithm, and stores soft information for each equation, in contrast to each edge in conventional LDPC decoders. The memory size of LDPC decoder is greatly reduced, and the delay of LDPC decoder is absorbed by the delay of soft channel detector.

## I INTRODUCTION

The invention of turbo codes [1] and the re-discovery of low-density parity-check (LDPC) codes [2] lead to increased efforts in utilizing iterative system in various applications. Since plenty of communication channels can be modeled as intersymbol interference (ISI) channels, for example the magnetic recording channel, application of iterative codes to ISI channels is of great interest. Combining iterative codes with ISI channels was first introduced in [3] under the name of “turbo equalization”. The key observation in turbo equalization is that ISI channel can be treated as rate-1 convolutional code, therefore it can be easily decoded by soft-input soft-output (SISO) algorithms and iterated with an outer convolutional or LDPC decoder.

Turbo equalization is well studied and understood (see [4] and references therein). However, hardware implementation of turbo equalizer is a challenging task. Soft decoders for ISI channels, such as BCJR [7], soft-output Viterbi algorithm (SOVA) [8], and decision-aided equalization (DAE) [9], are inherently sequential decoders, while LDPC decoder is preferably implemented in a parallel or semi-parallel architecture. Consequently, large memory is required to store intermediate soft information in order to concatenate LDPC decoder with soft channel decoder, making memory access the bottleneck in designing high speed turbo equalizers.

In this paper, we propose a new equation based LDPC decoder that is well suited for concatenation with ISI channels. In contrast to the simplifications in [5] and [6], where memory size is reduced by breaking down each iteration to several “sub-iterations” and storing soft information only for the latest sub-iteration, our method maintains the integrity of each iteration and produces a lower error floor. We consider a typical system as in Fig. 1, where the ISI channel is treated as a rate-1 inner code and its decoder is iterated with the LDPC decoder. The proposed LDPC decoder stores soft information per equation, instead of per edge in the factor graph. This memory arrangement greatly reduces size and access speed requirements. Moreover, equation based LDPC decoder operates in a serial fashion and is fully integrated with soft decoder for ISI channel. Therefore, the LDPC decoder does not incur additional delay.

The rest of the paper is organized as follows. Section II reviews LDPC decoding algorithms. Section III discusses several candidate soft decoders for ISI channels. Section IV introduces the equation based iterative decoder for ISI channels. Section V provides concluding remarks.

## II LDPC DECODING ALGORITHMS

LDPC codes can be easily decoded using sum product algorithm, which is summarized below for completeness (Fig. 2).

Suppose LDPC code is represented by a bi-partite graph. Each information bit corresponds to a bit node, and each check equation corresponds to a check node in the graph. Bit nodes are indexed by  $l$ , and check nodes are indexed by  $m$ . In the sum product decoding algorithm, there are four types of soft information circulating in the decoder:

1.  $llrP_l$ : soft information from the channel *a posteriori* probability (APP) decoder, one for each bit. Channel APP decoders can be BCJR, SOVA or DAE [7]-[9].
2.  $llrQ_{lm}$ : soft information from bit node  $l$  to equation node  $m$ , one for each edge.
3.  $llrR_{ml}$ : extrinsic information from equation node  $m$  to bit node  $l$ , one for each edge.
4.  $llrAPP_i$ : overall soft information after each iteration, one for each bit.

Each iteration of LDPC decoding consists of three steps (superscripts denote iteration numbers):

1. Each bit calculates the information  $llrQ_{lm}^i$  that it passes to the connecting equations, which is the sum of the extrinsic information  $llrR_{ml}^i$  from the previous iteration with the channel information  $llrP_l^i$  from the current iteration, excluding the extrinsic information from the same equation, i.e.,

$$llrQ_{lm}^i = llrP_l^i + \sum_{m' \neq m} llrR_{m'l}^{i-1}. \quad (1)$$

2. Each check equation (each row in the parity-check matrix) calculates the extrinsic information for each involved bit.  $\sum \oplus$  denotes the “LLRXOR” operation discussed below.

$$llrR_{ml}^i = \sum_{l' \neq l} \oplus llrQ_{l'm}^i. \quad (2)$$

3. Each bit calculates the overall soft information by summing up all the  $llrR_{ml}^i$ 's and the  $llrP_l^i$ 's,

$$llrAPP_l^i = llrP_l^i + \sum_m llrR_{ml}^i. \quad (3)$$

The LLRXOR operation in (2) is defined as

$$\begin{aligned} \text{LLRXOR}(y,z) &= \log(e^y + e^z) - \log(1 + e^{y+z}) \\ &= \max(y, z) + \log(1 + e^{-|y-z|}) \\ &\quad - \max(y+z, 0) - \log(1 + e^{-|y+z|}) \end{aligned} \quad (4)$$

where  $\log(1 + e^{-|a|})$  can be implemented by a look-up table. The operation in (4) requires five additions, two absolute values and two table look-ups. This procedure is referred to as the “sum product” (SP) algorithm.

The LLRXOR operation can be further simplified by eliminating table look-ups, i.e. omitting the  $\log(1 + e^{-|a|})$  term in (4), yielding the following approximation for the LLRXOR operator:

$$\text{LLRXOR}(y,z) \approx -\text{sign}(y)\text{sign}(z)\min(|y|, |z|). \quad (5)$$

(5) requires only absolute value, sign, and comparison operations. The simplification to (5) is referred to as the “signed sum product” (SSP) algorithm.

### III APP DECODING ALGORITHMS FOR ISI CHANNELS

Soft-input soft-output (SISO) algorithms for ISI channels include the BCJR algorithm [7], SOVA [8], DAE [9], etc. The original BCJR algorithm is a maximum *a posteriori* probability (MAP) algorithm and requires a forward iteration and a backward iteration, whereas SOVA and DAE are simplified APP decoders and only require sequential processing to obtain the soft information for each bit. In our concatenated systems, we use SOVA and DAE for channel SISO detection. To standardize the notation independent of

the soft channel decoding algorithm, we assume that soft channel detectors take in samples,  $llrAPP_l^{i-1}$ 's and  $llrP_l^{i-1}$ 's, and outputs  $llrP_l^i$ 's (see Fig. 3).

### IV EQUATION BASED LDPC DECODER

Consider a regular LDPC code with block length  $L$  and column weight  $M$ . A conventional implementation of SP or SSP requires  $L$  units of memory to store  $llrP_l^i$ 's and  $ML$  units of memory to store  $llrR_{ml}^i$ 's,  $llrQ_{lm}^i$ 's and  $llrAPP_l^i$ 's can be computed on the fly and does not need to be stored. Soft information is updated edge by edge within each iteration [5][6]. We call this implementation the “edge based” decoder, which can be implemented in either serial or parallel manner. Parallel architecture can achieve higher speed while serial architecture minimizes hardware size. For ISI channel serial implementation is preferred because soft information from the channel comes sequentially. At each iteration, edge based decoder computes  $llrQ_{lm}^i$  according to (1) after channel decoder calculates  $llrP_l^i$ . The decoder then computes  $llrR_{ml}^i$  according to (2). In a serial implementation, edge based decoder requires roughly  $L$  cycles per iteration for soft channel decoding and calculating  $llrQ_{lm}^i$ , and another  $L$  cycles for calculating  $llrR_{ml}^i$  and  $llrAPP_l^i$ . Therefore the total delay of channel and LDPC decoders is  $2L$  times the number of iterations. It should be noted that the amount of memory needed to store soft information in a fully pipelined decoder is proportional to the decoder latency.

When signed sum product algorithm is used to decode LDPC codes, the magnitudes of the extrinsic information conveyed in each equation are determined by the two  $llrQ$ 's with the smallest amplitudes. Other  $llrQ$ 's only contribute to the sign of the extrinsic information.

Based on the observation that only two  $llrQ$  magnitudes are needed, we propose an equation based architecture that eliminates the  $llrR$  memory and cuts the decoding delay in half. At iteration  $i$ , for equation  $m$ , there is only a need to store two smallest  $llrQ$  magnitudes (denoted  $min1_m^i$  and  $min2_m^i$ ), the signs for all the  $llrQ_{lm}^i$ 's, the position of  $min1_m^i$  (denoted  $I_m^i$ ), and the overall sign of the equations (denoted  $s_m^i$ ). Then,  $llrR_{ml}^i$ 's can then be calculated on the fly from the stored information as follows:

$$llrR_{ml}^i = \begin{cases} -s_m^i \cdot \text{sign}(llrQ_{lm}^i) \cdot min1_m^i, & \text{if } l \neq I_m^i \\ -s_m^i \cdot \text{sign}(llrQ_{lm}^i) \cdot min2_m^i, & \text{otherwise} \end{cases}, \quad (6)$$

where  $s_m^i = -\prod_l \text{sign}(llrQ_{lm}^i)$  is the sign of the  $m^{\text{th}}$  equation at the end of  $i^{\text{th}}$  iteration.

In the new equation based architecture, time cycles are allocated exclusively for the channel decoding, LDPC decoding is done on the fly. As soon as the first iteration channel SISO decoder outputs  $llrP^1_l$  for a given bit  $l$ , it is used to obtain  $llrQ^1_{lm}$  utilizing (1).  $llrQ^1_{lm}$  in turn is sent to update  $min1^1_m$ ,  $min2^1_m$ ,  $I^1_m$ , and  $s^1_m$  for each check node connected to the current bit. Once the channel detector finishes processing the entire codeword, we have all necessary information to compute  $llrR^1_{ml}$ 's, and to perform the second iteration channel decoding. On the second sweep through the bits, iterative decoder first obtains  $llrR^1_{ml}$  using (6). The result is then immediately sent to the SISO channel detectors as the *a priori* information. Once  $llrP^2_l$  becomes available (after a small channel detector latency),  $llrQ^2_{lm}$  is computed using (1) and sent to update  $min1^2_m$ ,  $min2^2_m$ ,  $I^2_m$ , and  $s^2_m$  for the connected check nodes. The rest of iterations proceed in a similar fashion.

The equation based arrangement takes advantage of both the sparseness of the parity-check matrix and the simplicity of the SSP decoding. The memory saving offered by equation based LDPC decoder is proportional to the code rate of LDPC code. This is particularly advantageous for the magnetic recording applications where high rate codes are predominantly used.

Equation based LDPC decoder architecture is now going to be further illustrated on a simple  $(L,k)$  product code whose parity check matrix is given by (7),

$$H = \left[ \begin{array}{cccc} 1 & \dots & 1 & \\ & & 1 & \dots & 1 \\ & & & & 1 & \dots & 1 \\ 1 & & 1 & & 1 & & \\ \dots & & \dots & & \dots & & \\ & & 1 & & 1 & & 1 \end{array} \right] \left. \vphantom{\begin{array}{cccc} 1 & \dots & 1 & \\ & & 1 & \dots & 1 \\ & & & & 1 & \dots & 1 \\ 1 & & 1 & & 1 & & \\ \dots & & \dots & & \dots & & \\ & & 1 & & 1 & & 1 \end{array}} \right\} L-k+1, \quad (7)$$

$\underbrace{\hspace{10em}}_L$

It is easy to see that  $H$  has no short cycles (girth>4), and a column weight of 2. The minimum distance is 4. Fig. 4 shows a fully pipelined architecture for iterative LDPC code decoder. Memories for  $min1$ ,  $min2$ , and  $llrP$  are initialized to 0. The equation based SSP circuit performs the following steps for each iteration (starting from “Eq. RAM 0” in Fig 4):

1. At each time clock,  $llrR^{i-1}_{ml}$  's are calculated using (6) by the “ $llrR$  calculation” block.
2. Each bit calculates the overall soft information  $llrAPP^{i-1}_l$  by summing up all the  $llrR^{i-1}_{ml}$ 's and  $llrP^{i-1}_l$  as an input to the channel SISO detector.

3. The channel SISO detector calculates  $llrP^i_l$ .
4. Each bit calculates  $llrQ^i_{lm}$  that it passes to the connecting equations according to (1).
5. Each bit stores the signs of the  $llrQ^i_{lm}$  's in “ $llrQ$  sign RAM”.
6. Each bit updates the equation information ( $min1$ ,  $min2$ ,  $I$ , and  $s$ ) for the two equations involving the current bit. This operation is performed by the “Equation update” block.

It is obvious that steps 1-6 for each iteration can be finished in the  $L$  clock cycles that is needed for channel SISO decoding, i.e., there is no additional clock cycles required for LDPC decoding.

Fig. 5 is a block diagram of a memory arrangement of the decoder in accordance with the equation based decoding method. As shown therein, memory has a partition for each equation, and each equation has a partition for storing magnitude data ( $min1$  and  $min2$ ), index data ( $I$ ), and sign data ( $s$ ). The equation memory blocks are arranged into two tiers according to the structure of the parity check matrix (7), one for row parities, and one for column parities. There are two copies of memory for each equation - one for the current iteration and one for the previous iteration. These two copies are used by the “equation update” block and the “ $llrR$  calculation” block in a ping-pong fashion. The MUX controls which memory to be read from and which memory to be written into according to an iteration number counter.

Equation based LDPC decoder needs  $L$  units of memory to store  $llrP$ 's,  $ML$  bits to store the sign of  $llrQ$ 's, and  $(L-k)$  units of memory to store equation based statistics. For high rate codes,  $(L-k)$  is small compared to the block size  $L$ . Therefore, memory requirement for the equation based decoder is dominated by the channel soft information  $llrP$ . Besides, since the turbo equalization latency is reduced from  $2L$  to  $L$  clock cycles per iteration, the overall size of equation based LDPC decoder is much smaller than that of conventional edge based decoder.

At the end of each iteration, the parity equations are checked. If all the parity equations are satisfied, we consider the decoding successful and start to output hard decisions that are derived from last iteration's  $llrAPP^i_l$ 's. Checking the equations is easy, because the overall sign for an equation,  $s^i_m$ , is -1 when equation  $m$  is satisfied, and is 1 otherwise. Therefore, there is no additional memory required for equation checking.

## V CONCLUSION

The key aspects of our equation based LDPC decoder are storing and updating soft information according to equations, and computing soft information for each edge on the fly. Doing so reduces the memory size of the LDPC decoder, especially when high rate LDPC codes are used.

Moreover, the proposed LDPC decoder does not require additional latency beyond what the channel SISO decoder needs. Therefore, the equation based decoder is very efficient for decoding LDPC codes on ISI channels.

### REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. on Communications*, Geneva, Switzerland, May 1993.
- [2] D. J.C. MacKay, "Good error-correcting codes based on very sparse matrices", *IEEE Trans. Inform.*, vol 45, pp. 399-431, Mar. 1999.
- [3] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel," in *Proc. Int. Symp. Turbo codes*, Brest, France, Sept. 2000, pp. 371-374.
- [4] M. Tuchler, R. Koetter, and A.C. Singer, "Turbo Equalization: Principles and New Results", *IEEE Trans. Commun.*, vol 50, pp 754-767, May 2002.
- [5] Y. Engling, P. Pakzad, B. Nikolic, and V. Anantharam, "VLSI architectures for iterative decoders in magnetic recording channels," *IEEE Trans. Mag.*, vol 37, pp 748 - 755, Mar. 2001.
- [6] M. M. Mansour and N.R. Shanbhag, "High-Throughput LDPC Decoders," *IEEE Trans. VLSI Sys.*, vol 11, pp 976-996, Dec. 2003.
- [7] L.R.Bahl et al, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform.*, vol. IT-20, pp. 284-287, Mar. 1974.
- [8] J. Hagenauer and P. Hoehner, "A Viterbi algorithm with soft-decision outputs and its applications," In *Proc. IEEE Global Telecomm. Conf.*, 1989, pp 1680-1686.
- [9] Z. Wu, *Coding and Iterative Detection for Magnetic Recording Channels*, Kulwer Academic Publishers, 2000.
- [10] Z. Wu and G. Burd, "LDPC Decoder and Method Thereof," US Patent application, filed 2000.

FIGURE 1. LDPC code with ISI channel

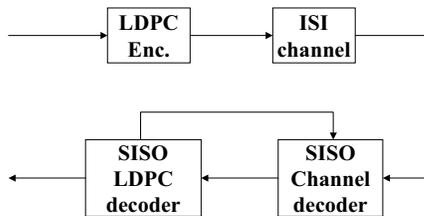


FIGURE 2. Factor graph and sum product decoding

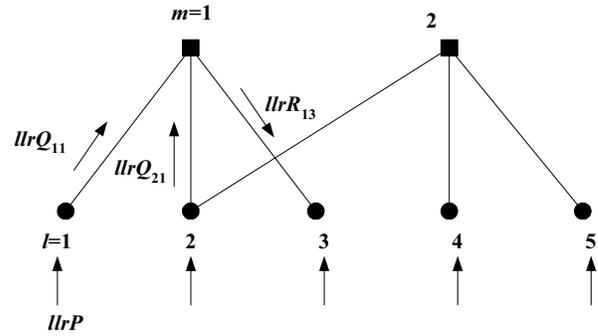


FIGURE 3. Generic channel SISO decoder

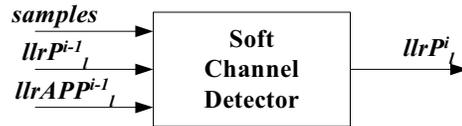


FIGURE 4. Equation based LDPC decoder

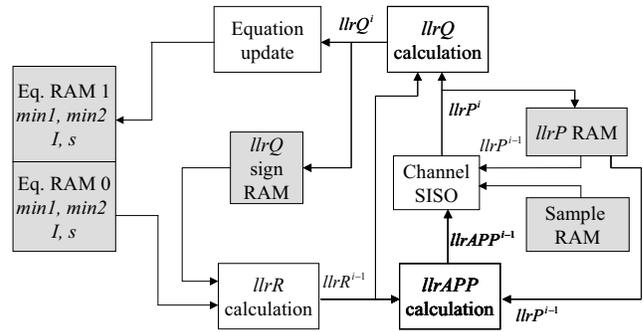


FIGURE 5. Equation memory arrangement

