

CONVERGENCE-OPTIMIZED VARIABLE NODE STRUCTURE FOR STOCHASTIC LDPC DECODER

Qichen Zhang, Yun Chen, Di Wu, Xiaoyang Zeng*

Yeong-luh Ueng[†]

Fudan University
State Key Laboratory of ASIC and system
Shanghai 200433, CHINA

National Tsing Hua University
Department of Electrical Engineering
Hsinchu, Taiwan

ABSTRACT

By using stochastic computation, a fully-parallel low-density parity-check (LDPC) decoder can be implemented using a lower wire complexity. In order to enhance the decoder performance, probability tracers, such as up/down counters, are added at each edge between variable nodes and check nodes, as described in previous literature. However, this causes a large decoding latency and a high number of decoding failures. In this paper, a convergence-optimized structure for variable nodes is proposed that is able to overcome these issues. As a result, the throughput for the proposed decoder is 20.5Gb/s, which is 101% higher than the original counter-based decoder presented in the previous literature.

Index Terms— LDPC decoder, stochastic decoding, up/down counter, convergence-optimized

1. INTRODUCTION

Low-density parity-check codes (LDPC) are a class of linear block codes which can achieve near-capacity performance. The concept of LDPC codes were initially developed by Gallager in 1963 [1], but were neglected since they were generally considered too difficult to implement in a practical application until his work was revisited by Mackay and Neal in 1996 [2]. Nowadays, LDPC codes are used extensively in a number of communication standards. LDPC codes can be defined using a parity check matrix (PCM). A Tanner graph, which is a kind of bipartite graph, can clearly show the LDPC decoder architecture. In the Tanner graph, each column of the PCM can be expressed as a variable node (VN), and each row as a check node (CN). The Sum-Product Algorithm (SPA), which is an iterative decoding algorithm, exchanges the decoding messages between the VNs and the CNs during

each iteration. The SPA can be used to achieve an outstanding decoding performance.

Stochastic LDPC decoding has evolved from the SPA, and uses Bernoulli sequences to represent decoding messages, where the ratio of 1s in the sequence represents the message probability [3]. For example, if the probability value of a message is 0.5, and the length of the Bernoulli sequence is 50, the number of 1s in the sequence will be about 25. Consequently, stochastic decoding is a form of bit-serial algorithm which can be used to reduce the wire complexity of an SPA-based decoder. Moreover, stochastic decoders have a more simpler computation structure. For instance, a simple AND gate can be used to implement multiplication tasks.

However, decoding by a stochastic LDPC decoder may fall into dead loop due to the correlation introduced by the cycles in the Tanner graph. Edge-memory (EM) [8] was the first structure that was developed to mitigate this problem, but the area that the EM occupies is too large and the throughput is low. Majority-based tracking forecast memories (MTFM) [5] is a modified EM structure. MTFM uses tracking forecast memories (TFM) to replace EM, and combines the TFM that exist in each sub-node into a single TFM. Although an excellent performance and good throughput can be achieved using this method, the area is still a bit too large. In [6], delay stochastic (DS) structure is used. However, although the throughput is excellent and the area is small, performance is sacrificed. When the counter-based VN structure presented in [4] is considered, an implementation based on stochastic decoding can be used to achieve a good performance, a low wire complexity and a small area. However, the decoding latency will increase.

In this paper, a convergence-optimized VN structure is proposed, together with the associated decoder architecture where the termination of the decoding process described in [4] is modified, thereby significantly reducing the decoding latency. The probability tracer is still implemented using a counter, so the area of the decoder remains smaller. The decoder is implemented for the regular (2048, 1723) LDPC code specified in the 10GBASE-T [7] standard. The throughput for the proposed decoder is 20.5Gb/s, which is 101% higher than

*Sponsored by the National Natural Science Foundation of China (Grant No. 61306043), Shanghai Natural Science funding (Grant No. 13ZR1401800). National Natural Science Foundation of China under Grant 61234002. National Science and Technology Major Project of China (Grant No. 2011ZX 03003-003-03).

[†]Sponsored by the Ministry of Science and Technology of Taiwan (Grant No. MOST 103-2622-E-007-029-CC2).

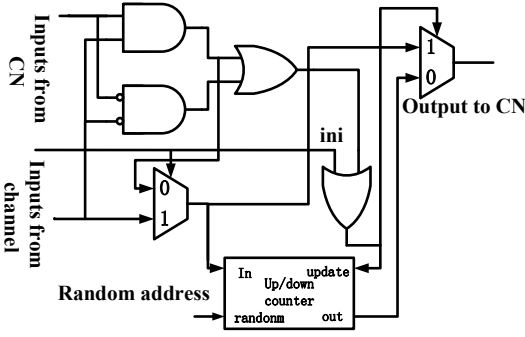


Fig. 1. Structure of 2-input counter-based VN [4]

the original counter-based decoder presented in [4].

2. PRELIMINARY

In the conventional SPA, the CNs and VNs exchange complete decoding messages during each iteration. However, in stochastic decoding, the CNs and VNs only exchange one bit of the message during each decoding cycle (DC), which is represented using the Bernoulli sequence. In order to perform stochastic decoding, channel messages represented using a log-likelihood ratio (LLR) must be converted to Bernoulli sequences. Firstly, Look up Tables (LUTs) are used to convert the channel message to a probability using $P_a = \frac{e^{L_a}}{e^{L_a} + 1}$, where L_a is the channel LLR message for VN v_a . The probability value together with a random number generated using Linear Feedback Shift Registers (LFSRs) are then compared to obtain the Bernoulli sequence. The comparator outputs a value of 1 when the probability value is larger than the random number, and otherwise outputs a value of 0. In the following, the stochastic computation executed at both the VN and the CN is introduced using a degree-2 VN and a degree-3 CN for convenience.

- 1) *VN*: Suppose that VN v_a receives bits from CN c_i together with its channel message, the message probability from v_a to c_k is calculated according to

$$P_{a \rightarrow k} = \frac{P_{i \rightarrow a} P_a}{P_{i \rightarrow a} P_a + (1 - P_{i \rightarrow a})(1 - P_a)} \quad (1)$$

- 2) *CN*: Suppose that CN c_i receives the bits from VNs v_a and v_b , the message probability from c_i to v_c is calculated according to

$$P_{i \rightarrow c} = P_{a \rightarrow i}(1 - P_{b \rightarrow i}) + P_{b \rightarrow i}(1 - P_{a \rightarrow i}) \quad (2)$$

Fig. 1 shows the structure of the 2-input counter-based VN proposed in [4]. The VN receives the bit-serial Bernoulli sequences from both the channel and a neighboring CN. The function of the VN is to verify whether or not the inputs are

the same. If the messages from both the CN and the channel are the same, then the VN will be set to the “unhold” state. In this state, the VN will output the input bit to the other neighboring CN and the counter. If the value of this bit is 1, the value of the up/down counter will be increased by 1 and if the value of the bit is 0, the counter value will be decreased by 1. In contrast, if the value of the two input bits for the VN are different, the VN will remain in the “hold” state, and the value of the counter is not changed. After a number of DCs, the value of the counter will be changed based on the number of “unhold” inputs, so it can be used to represent the probability of the outputs when the VN is in the “unhold” state. A random number generated by LFSR is compared with the counter value during each DC to get the output of the counter. If the counter value is larger than the random number, the output of the counter is set to 1 and otherwise it is set to 0. The VN will output the result of the counter to corresponding CN while the VN is in the “hold” state. The function of the CNs is to check whether or not the input values satisfy the check equations. The CN can be implemented by using a simple XOR gate since the inputs are bit-serial.

3. PROPOSED VN STRUCTURE AND DECODER

The up/down counting mechanism implemented for the decoder presented in [4] causes the decoding time to be lengthy, where the up/down counter acts as a probability tracer and hard decision generator. The sign bit of the counter value is used to generate the hard decision. In order to trace the probability more accurately, the value of the counter is set to a large number, such as ± 256 , which means that when the value of the counter is at one extreme, 256 DCs will be needed before the hard decision can change. For example, if the value of counter is -256 , the output will be 1, but the correct output should be 0, 256 DCs will be required before the value can change from -256 to 0. Actually, after analyzing the decoding process by simulation, it can be observed that when the decoding process is about to terminate, the majority of the code bits are correct, but the values for the counters where the bits are wrong are at the extremes. This means that the decoder requires hundreds of DCs in order to correct a few wrong bits, which is a major waste of decoding time.

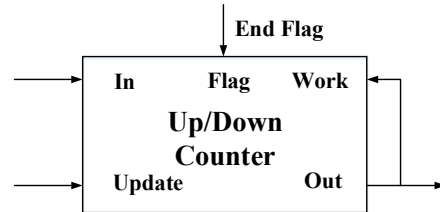


Fig. 2. Block diagram for the proposed up/down counter

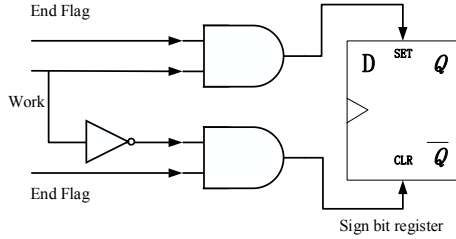


Fig. 3. Control logic for the sign bit register of the counter

3.1. Proposed counter-based VN structure

In this paper, a new up/down counter structure is proposed that is able to mitigate this problem. This type of counter will operate in the same manner as the original counter during the start and the middle period of the decoding process. However, when the decoding process is about to terminate, the sign bit of the counter will be reversed if the input of the counter is different to the output. When the decoding process is about to terminate, the inputs of the counter are believed to be correct. So, directly changing the output is one approach to reducing the decoding time.

The proposed up/down counter contains a primary up/down counter together with a small combinational logic block to reverse the sign bit of the counter. Fig. 2 shows the block diagram for the proposed counter. The counter receives an end flag from the check block, and the output from itself is set as a feedback signal. The end flag is used to indicate that the decoding is about to terminate. The generation of the end flag will be described in the following subsection. As shown in Fig. 3, the combinational logic block will set the SET or CLR pin of the sign bit register of the counter as determined by the end flag and the feedback signal, i.e., the Work signal. When the end flag is 0, the output values for the two AND gates will each be 0, the SET and CLR pin are not operational. When the value of the end flag is 1 and the Work signal is 1, the output to SET will be 1 and the sign bit will be set to 0. Otherwise, when the value of the end flag is 1, and the Work signal is 0, the CLR pin will be triggered and the sign bit will be set to 1. In this way, there is no need for the hard decision generator to wait for the sign value of the counter to change if this counter has tendency to change its sign. If the input of the wrong bit's counter changes correctly, the sign bit for the counter will immediately change, meaning that the decoding time will be significantly reduced.

3.2. Proposed counter-based (2048, 1723) decoder

The proposed VN structure is adopted in order to implement a counter-based decoder for the (2048, 1723) LDPC code in the 10GBASE-T standard. The PCM is regular and the size is 384×2048 , meaning that the proposed decoder contains 384 CNs and 2048 VNs. The degree of each VN and each CN

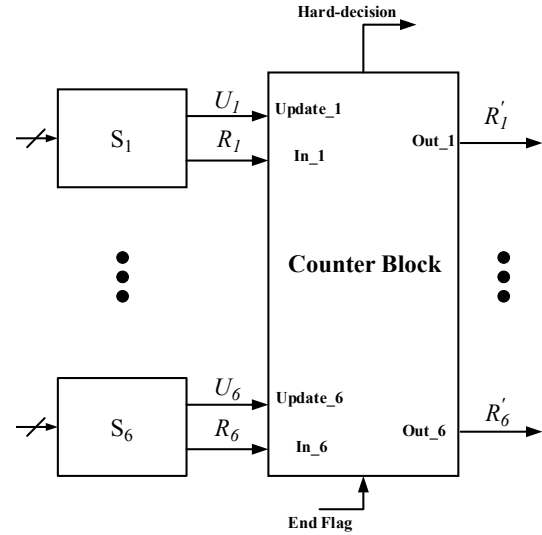


Fig. 4. Block graph for the proposed VN

is 6 and 32, respectively. Fig. 4 shows the VN structure for the proposed (2048, 1723) decoder, where S_i represents the i th sub-node. The VN contains six sub-nodes and receives its channel message and CN messages from its neighboring six CNs. The structure of the sub-nodes is based on the structure proposed in [4], except that the counter is implemented based on the proposed techniques. The counter block contains the proposed up/down counter and output generating logic block. The U_i signal is the update signal from each sub-node. The R_i message will be sent directly to neighboring CNs, i.e., $R'_i = R_i$, if the value of corresponding U_i is 1, otherwise the output from the counter will be sent to the CNs.

Fig. 5 shows the block diagram for the proposed stochastic (2048, 1723) LDPC decoder, where the maximum number of DCs was set to 700. The VN and CN blocks are connected by an interconnection network. The number of LFSRs is 64, every groups of 32 VNs receive a random number from one of the LFSRs. The check block generates the end flag and transmits it to the VNs. The check block receives 2048 hard decisions (decoded bits) that are used to calculate the number of unsatisfied check equations, which is denoted as N_u . Since N_u and the number of incorrect code bits are positively correlated, N_u can be used to indicate whether or not the decoding process is about to terminate. Based on computer simulation, when the value of N_u is below 25, the number of incorrect code bits will be below 10. As a result, 25 is used as the threshold to trigger the end flag. When using the proposed counter-based VNs, the decoder only requires a minimal number of DCs in order to complete the decoding process, meaning that the amount of decoding time required can be significantly reduced. Compared to the original counter-based decoder presented in [4], the proposed decoder only includes the addition of a number of simple combinational

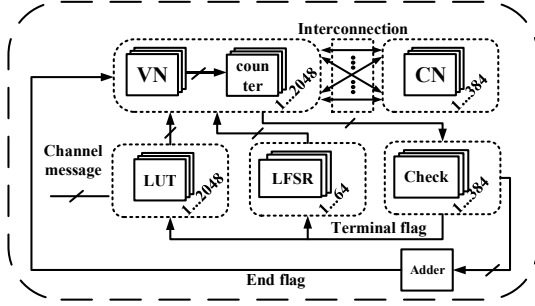


Fig. 5. Block graph for the complete decoder

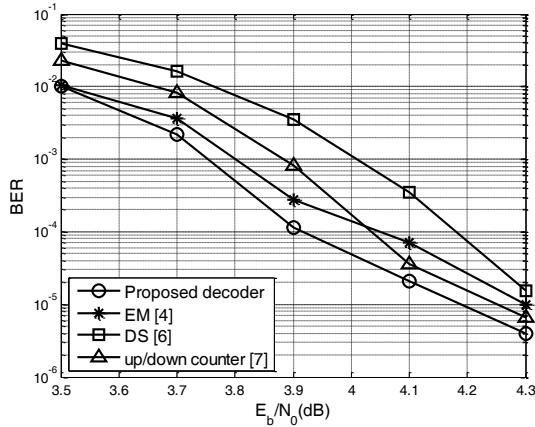


Fig. 6. BER performance for the (2048, 1723) code using a variety of stochastic decoders

logic blocks, so the impact on the associated area overhead is minimal.

4. RESULTS

Fig. 6 shows a comparison of the BER performance for both the proposed decoder and the original counter-based decoder presented in [4]. It can be seen that proposed decoder is able to achieve a better BER performance in the moderate to high SNR regions. When compared to the proposed decoder, the original counter-based decoder is not able to yield a correct codeword with a higher probability in the moderate SNR region, even though the maximum number of DCs is reached. Hence, the performance improvement can be gotten, and is more significant in the moderate SNR region for the proposed decoder. Fig. 7 shows the DC distribution for both the proposed decoder and the original decoder presented in [4] at an SNR of 5.5dB. The average number of DCs required for the proposed and the original decoders is 50.6 and 100.3, respectively. It can be seen that the convergence speed for the proposed decoder is significantly increased, and, hence, the throughput is also significantly increased compared to the original decoder presented in [4].

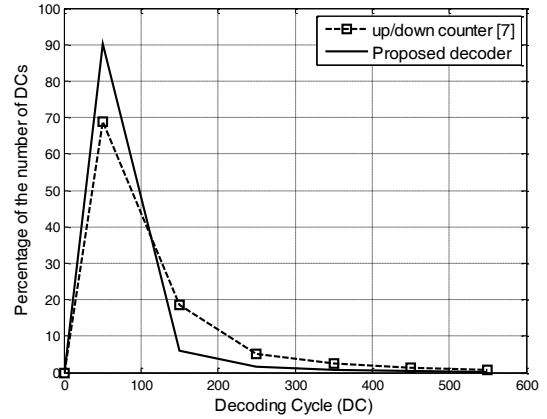


Fig. 7. DC distribution at an SNR of 5.5dB

Table 1. Synthesis results of decoders in SMIC 65nm

	Area (mm^2)	Frequency (MHz)	Throughput (Gb/s)
Proposed	2.7	500	20.5
[4] counter	2.2	500	10.2
[8] EM	46097 Slices	222	1.66
[5] MTFM	3.3	500	61.3
[6] DS	1.97	750	172.4

The proposed decoder was synthesized using the SMIC 65nm 1.0V process. The area, frequency and throughput for the proposed decoder and the decoders described in [8], [5], [6] and [4] are shown in Table I, where the results are all scaled to 65nm. It can be seen that the area of the proposed decoder is smaller than that of the EM [8] and MTFM [5] implementations. Although the DS decoder presented in [6] is able to achieve a higher throughput and a lower area compared to the proposed decoder, it can be seen from Fig. 6 that the proposed decoder is able to achieve a better error rate performance. Compared to the original counter-based decoder presented in [4], the proposed decoder is able to provide a throughput that is 101% higher, with a consequent increase in area of only 22%.

5. SUMMARY

A convergence-optimized variable node (VN) structure for counter-based stochastic LDPC decoders has been presented. The proposed technique is able to significantly reduce the decoding time and, hence, increase the decoder throughput. When compared to the original counter-based decoder presented in [4], the proposed decoder is able to provide a throughput that is 101% higher, with a consequent increase in area of only 22%.

6. REFERENCES

- [1] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, No. 1, pp. 21–28, 1962.
- [2] D. J. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics letters*, vol. 32, No. 18, pp. 1645–1646, 1996.
- [3] B. Gaines, "Advances in information systems science," New York:Plenum, 1969, vol. 2, ch. 2, pp. 37–172.
- [4] Q. C. Zhang, Y. Chen, D. Wu, X. Y. Zeng, and Y. L. Ueng, "An area-efficient architecture for stochastic ldpc decoder," *IEEE International Conference on Digital Signal Processing (DSP)*, pp. 244–247, 2015.
- [5] S. S. Tehrani, A. Naderi, G.-A Kamendje, S. Hemati, S. Mannor, and W. J. Gross, "Majority-based tracking forecast memories for stochastic ldpc decoding," *IEEE Transactions on Signal Processing*, vol. 58, No. 9, pp. 4883–4896, 2010.
- [6] A. Naderi, S. Mannor, M. Sawan, and W. J. Gross, "Delayed stochastic decoding of ldpc codes," *IEEE Transactions on Signal Processing*, vol. 59, No. 11, pp. 4883–4896, 2011.
- [7] IEEE P802.3an 10GBASE-T Task Force, "<http://www.ieee802.org/3/an/>," .
- [8] S. Sharifi Tehrani, S. Mannor, and W. J. Gross, "Fully parallel stochastic ldpc decoders," *IEEE Transactions on Signal Processing*, vol. 56, No. 11, pp. 5692–5703, 2008.
- [9] M. R. Yazdani, S. Hemati, and A. H. Banihashemi, "Improving belief propagation on graphs with cycles," *Communications Letters*, vol. 8, No. 11, pp. 57–59, 2004.