Loosely Coupled Memory-Based Decoding Architecture for Low Density Parity Check Codes

Se-Hyeon Kang, In-Cheol Park

Department of EECS, Korea Advanced Institute of Science and Technology

Abstract

Parallel decoding is required for low density parity check (LDPC) codes to achieve high decoding throughput, but it suffers from a large set of registers and complex interconnections due to randomly located 1's in the sparse parity check matrix. This paper proposes a new LDPC decoding architecture to reduce registers and alleviate complex interconnections. To reduce the number of messages to be exchanged among processing units, two data flows that can be loosely coupled are developed by allowing duplicated operations. In addition, intermediate values are grouped and stored into local storages each of which is accessed by only one processing unit. In order to save area, local storages are implemented using memories instead of registers. To verify the proposed architecture, a 1024 bit rate-1/2 LDPC decoder is implemented using a 0.18 um CMOS process. The decoder runs correctly at the frequency of 200 MHz, which enables almost 1Gbps decoding throughput. Since the proposed decoder occupies an area of 10.08 mm^2, it is less than one fifth of area compared to the previous architecture.

1. Introduction

The LDPC code, which was first introduced by Gallager in 1962 [1], is a kind of binary linear block codes. A (n, γ, ρ) LDPC code means that its block length is n and the column and row weight of its parity check matrix are γ and ρ, which represent the number of 1's in a column and a row, respectively. The column and row weight are much smaller than n to achieve a sparse matrix **H**. Fig. 1 shows the factor graph of a (12, 3, 6) LDPC code, which consists of two sets of nodes: i.e. variable nodes, $\{v_j\}$, and check nodes, $\{c_i\}$.

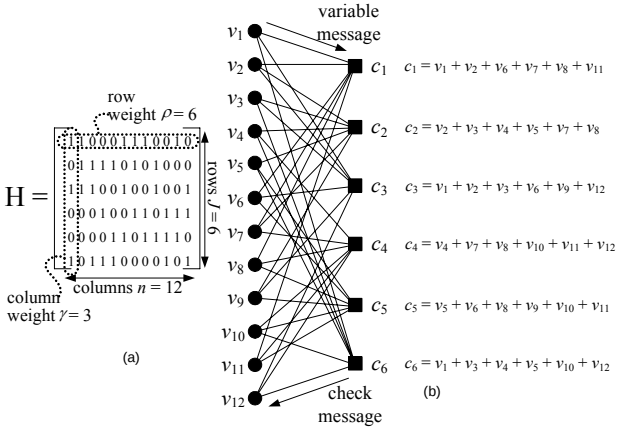

Fig. 1. a (12, 3, 6) LDPC code. (a) parity check matrix. (b) factor graph.

Recent simulation results show that the LDPC code can achieve a performance that is within 0.04 dB of Shannon limit [2]. Despite of these advantages, when the LDPC code was first introduced, it made little impact on the information theory community because enormous storage is required in encoding and the large computational complexity in decoding. Modern VLSI technology, however, is so advanced that it enables parallel architectures exploiting the benefit of inherently parallel LDPC decoding algorithms. Blanksby et al. implemented an1-Gb/s fully parallel decoder in which the message passing algorithm is directly mapped [4]. This architecture, however, requires a large number of complex routings between concurrent processing units (*PU*s) each of which corresponds to a node of the factor graph of the code, leading to the average net length of 3mm and the total die size of 52.5mm^2. On the other side, Yeo et al. proposed an area-efficient architecture that serializes the computations by sharing *PU*s [5]. Consequently, one iteration takes about ten-thousand cycles and wide-input multiplexers are required to select one out of several thousand intermediate values to be fed into the shared *PU*s. These two counter examples show that high throughput LDPC decoding architectures should exploit the benefit of parallel decoding algorithms while reducing the interconnection complexity.

This paper proposes a new architecture to reduce registers and alleviate complex interconnections. The rest of this paper is organized as follows. Section 2 proposes a new LDPC decoding architecture based on loosely coupled two data flows and an efficient scheduling algorithm. The performance of the proposed LDPC decoder is summarized in Section 3. Finally, Section 4 addresses some concluding remarks.

2. LDPC Decoding Architecture

Although the fully parallel implementation of the message-passing algorithm is straightforward and results in a high throughput decoder, it faces with complex interconnections caused by a quite large number of irregular edges. The interconnection complexity can be reduced by employing the serial architecture in which a shared *PU* computes all the rows or columns one after another. The complex interconnection problem, however, is transformed to how to read and write the messages aligned in different ways, requiring complex index generation to access the storage elements.

a) Proposed Loosely Coupled Processing

To resolve the complex interconnection problem, the proposed architecture delivers only the row and column summation values, Δ_j and Λ_i, to the neighbor nodes as shown in Fig. 2. To derive the individual messages from the summation value, a *PU* has to include additional operations computed in the neighbor *PU*s in the previous architecture. The proposed architecture consists of two data flows to minimize the number of messages to be exchanged, leading to an area-efficient decoder. Instead of sending all the messages, only the minimal information is exchanged between the

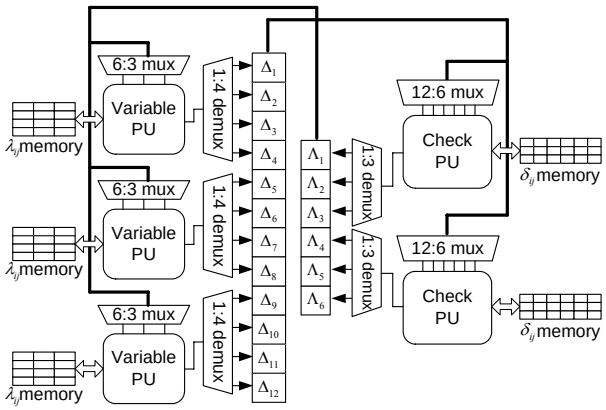

Fig. 2. Partially parallel LDPC decoding architecture.

This work was supported by Institute of Information Technology Assessment through the ITRC, and by IC Design Education Center (IDEC) Samsung 0.18um CMOS process.

two data flows. Then each data flow reconstructs the original messages using the minimal information. Furthermore, the intermediate values are stored into local memories each of which is accessed by only one *PU*. As δ_{ij} and λ_{ji} are not delivered to other type of *PU*s, the number of interconnections is reduced significantly. The duplicated operations increase the hardware complexity of *PU*s. This overhead is inevitable to reduce the interconnection complexity that is more serious than the logic complexity in today's deep submicron technology.

To reduce the overhead, the proposed architecture can exploit the partially parallel architecture in which each *PU* takes in charge of several rows or columns. In addition, since a *PU* processes a row or column at a time, the intermediate values, $\{\delta_{ij}\}$ and $\{\lambda_{ji}\}$, processed by a *PU* can be grouped and stored into a local memory instead of flip-flops to save area. This architecture can reduce the number of interconnections further by sharing the multiplexers required to access the row and column summation values.

b) Overlapped Scheduling

Traditionally, the *CTV* operations start only after the entire *VTC* step finishes completely, and vice versa. A well-scheduled sequence of the message calculations can reduce the latency, because the *CTV* step can start in the course of the *VTC* step if the corresponding row summation values are available, and vice versa. Fig. 3 shows an example scheduling for the (12, 3, 6) LDPC code, assuming that the numbers of check *PU*s and variable *PU*s are two and three, respectively.

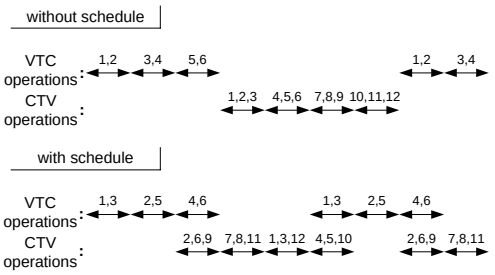

Fig. 3. An example of overlapped processing.

It takes a considerable amount of time to find an optimum schedule that minimizes the overall cycles because of the large number of rows and columns. We proposed a new scheduling algorithm developed for the partially parallel LDPC decoding architecture based on the concept of the matrix permutation. The algorithm makes the row sequence and column sequence that can result in empty spaces in the lower left and upper right corners of the permuted matrix when the matrix **H** is rearranged according to the sequences. The well-scheduled sequence in Fig. 3 can be obtained using the proposed algorithm as shown in Fig. 4 (b). For various LDPC codes, the proposed algorithm saves more than 25% cycles on the average. More details about the scheduling algorithm and its performance results can be found in [5].

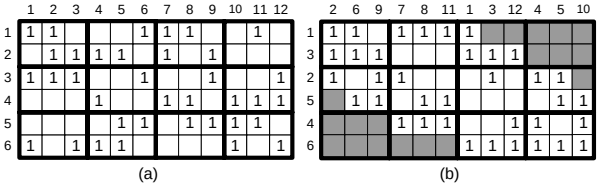

Fig. 4. Scheduling by the permutation of matrix H. (a) Matrix H. (b) Permuted matrix.

3. Implementation Results

We designed a (1024, 3, 6) LDPC decoder based on the proposed partially parallel architecture using a 0.18 um 4-Metal CMOS process. The performances of the decoders which has 32 check *PU*s and 64 variable *PU*s are summarized in TABLE I. The decoder occupies an area of 10.08 mm^2 and provides almost 1Gbps decoding throughput at the frequency of 200MHz. The performance of the second decoder is comparable to the fully parallel architecture proposed by Blanksby et al., but the proposed decoders achieve significant area reduction. Fig. 5 shows the layout of the proposed LDPC decoder. Although the iteration number in Blanksby's implementation is fixed to 64 due to the scan-chain like I/O mechanism, there is no restriction in the proposed architecture. We chose 8 iterations to provide sufficient BER performance.

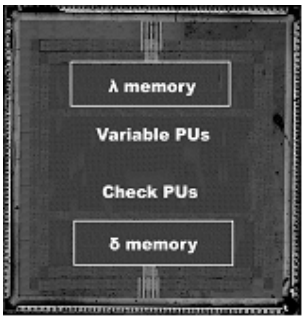

Fig. 5. Die photo of the proposed LDPC decoder chip.

TABLE I
COMPARISON OF LDPC DECODERS

	Blanksby [3]	Proposed
Technology	0.16 um	0.18 um
Bit rate	1 Gbps	985 Mbps
Frequency	64MHz	200MHz
Gate counts	1750K	543K
Area	52.5 mm^2	10.08 mm^2

4. Conclusion

This paper has presented a new LDPC decoding architecture proposed to relax the interconnection complexity and reduce area. In the proposed architecture, only the row and column summation values are exchanged among check and variable *PU*s to reduce the interconnection complexity. To recover the individual messages from the summation values, the function of a *PU* is restructured to include some operations that are traditionally performed in the neighbor *PU*s. In addition, intermediate values accessed by a *PU* are grouped and stored into a local memory instead of registers, since other *PU*s do not access them. The memory-based architecture is area-efficient in the sense that the memory takes much less area compared to the register if both have the same size. We have extended the proposed architecture for the partially parallel architecture to further reduce area by increasing memory usage, and have proposed an efficient algorithm that schedules the processing order of the partially parallel architecture to save the overall processing cycles by overlapping *CTV* and *VTC* steps.

Reference

[1] R. G. Gallager, "Low density parity check codes," IRE Trans. Info. Theory, vol. IT-8, pp. 533-547, Jan. 1962.

[2] S. Chung, D. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the Shannon limit," IEEE Comm. Letters, vol. 5, pp. 58-60, Feb. 2001.

[3] A. Blanksby and C. Howland, "A 690-mW 1-Gb/s, rate-1/2 low-density parity-check code decoder," IEEE J. of Solid-State Circuits, vol. 37, pp. 404-412, Mar. 2002.

[4] E. Yeo, P. Pakzad, B. Nikolić and V. Anantharam, "VLSI architectures for iterative decoders in magnetic recording channels," IEEE Trans. Magnetics, vol. 37, pp. 748-755, Mar. 2001.

[5] I. C. Park and S. H. Kang, "Scheduling Algorithm for Partially Parallel Architecture of LDPC Decoder by Matrix Permutation," *IEEE International Conference on Circuits and Systems*, May 2005.