

AN OVERLAPPING MIN-SUM LDPC DECODER FOR IEEE 802.11n

Sekson Timakul^{}, Itsara Tanyanon[†] and Somsak Choomchuay^{††}*

^{*}College of Data Storage Technology and Applications, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

Tel./Fax: + 66-2-326-4731, E-mail: sekson.timakul@gmail.com

^{†,††} Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
Tel: +66-2-329-8344 Ext.114, Fax: +66-2-329-8346, [†]Email: itsara@elec.kmitl.net, ^{††}E-mail: kchsomsa@kmitl.ac.th

ABSTRACT

This paper presents overlapping techniques designed for a compact hardware LDPC decoder with MS algorithm. The design is applicable to IEEE 802.11n standard. We elaborate how to reduce hardware and cycle time between row and column operation. The hardware utilization can be better enhanced and 16-40% cycle time reduction compared to a non-overlapping decoder can be achieved.

1. INTRODUCTION

As Low density parity check (LDPC) codes have been proved to perform very close to the Shannon limit, more researchers are interesting using them in many kind of applications. LDPC codes were firstly introduced by Gallager [1] in the 1960's and re-discovered by Mackey [2] in 1990's. Quasi-cyclic LDPC (QC-LDPC) codes of which parity check matrices consist of circulant matrices are a subclass of LDPC codes. This subclass of codes has low-complexity in encoding and decoding methodology. With such regards, QC-LDPC codes have been adopted by IEEE for wireless communications and appear in both 802.11n and 802.16e standards.

IEEE 802.11n-2009 [3] (WLAN) is a recent amendment enhanced for higher throughput upon the previous 802.11n standards by adding multiple-input multiple-output (MIMO) and many other newer features. When the guard interval is assumed, the data rate is more than 150 Mb/s. The high data rate also implies the faster decoding speed of the FEC section. To get the higher data rate decoder, Y. Chen and K.K. Parhi [4] has proposed an overlapped message passing technique. I. Park and S. Kang [7] proposed the scheduling algorithm for partially parallel architecture. That technique is basically a matrix permutation. Shih et. al. [5] proposed H-matrix re-ordering and applied to IEEE802.16.

Based on the works mention above, it is fairly obvious that the fully parallel LDPC decoding architecture can deliver high decoding throughput, but at the cost of higher hardware complexity caused by a large set of processing units and complex interconnections. We, hereby in this paper, present overlapping techniques between CNU and

VNU processes. We are particularly looking into Min-Sum algorithm and focusing to IEEE802.11n system.

The rest of the paper is organized as follows. In section 2 explains briefly about the min-sum decoding algorithm. In section 3 we present overlapping techniques that hardware can be effectively used. In section 4 we will look into the suitable architecture. In section 5 we show some obtained results according to the algorithm used and the corresponding architectures. Finally in section 6, we conclude the paper.

2. MIN-SUM ALGORITHM

The Min-Sum algorithm is an approximated version of belief propagation (BP). In brief, the decoding steps are given as follows.

$$\text{Let } L(c_i) = \begin{pmatrix} \Pr(c_i = 0 | y_i) \\ \Pr(c_i = 1 | y_i) \end{pmatrix},$$

Initialization: Messages at variable nodes are set to:

$$L(q_{ji}) = L(c_i) = y_i \quad (1)$$

Step 1: Check node update (performed by CNUs):

$$L(r_{ji}) = \left(\prod_{i \in R(i) \setminus j} \text{sign}(L(q_{i'j})) \right) \min_{i \in R(i) \setminus j} |L(q_{i'j})| \quad (2)$$

Step 2: Variable node update (performed by VNUs):

$$L(q_{ij}) = L(c_i) + \sum_{j \in C_i \setminus i} L(r_{ji}) \quad (3)$$

Step 3: Soft Decision:

$$L(Q_i) = L(c_i) + \sum_{j \in C_i} L(r_{ji}) \quad (4)$$

Step 4: Hard Decision:

$$\hat{c}_i = \begin{cases} 0 & \text{if } L(Q_i) < 0 \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

Step 5: Parity Check:

$$H \cdot (\hat{c}_1, \hat{c}_2, \dots, \hat{c}_n)^T = 0 \quad (6)$$

3. OVERLAPPING TECHNIQUE

In this section, the overlapping computation resulted from matrix H re-ordering is outlined. Each row of the LDPC codes can be viewed as a concatenation of each layer similar to observations made for architecture-aware LDPC codes noted in [4], [7]. After the check nodes processing are finished for one row grouping, the messages are immediately used to update the variable nodes, whose results are then used to process the next row grouping of check nodes.

The parity check matrix H with codeword 648 bit and code rate of 1/2 defined in wireless IEEE 802.11n system is shown in Fig. 1. This H matrix produces a systematic code, i.e., it encodes an information block of size k , into a codeword, c , of size n , by adding $n-k$ parity bits. The code still satisfies $H \times c^T = 0$, where H is of the size $(n-k) \times n$. The base matrix H includes elements of blank, zero and non-zero integer of 12 rows by 24 columns. The element of blank (-) can be expanded as a 27×27 zero matrix. The element of non-zero integer can be expanded as a cyclically-shifted version of a diagonal 27×27 matrix. It is worthwhile to note that this matrix contains many zero sub-matrices. The zero-shift sub-matrices are arranged in a stair-case structure. With this note, the matrix can be re-ordered, and the overlap computation can be obtained.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
1	0	-	-	0	0	-	-	0	-	-	0	-	-	0	1	0	-	-	-	-	-	-	-	-	-
2	22	0	-	-	17	-	0	0	12	-	-	-	0	0	0	-	-	-	-	-	-	-	-	-	-
3	6	-	0	-	10	-	-	-	24	-	0	-	-	0	0	-	-	-	-	-	-	-	-	-	-
4	2	-	-	0	20	-	-	-	25	0	-	-	-	0	0	-	-	-	-	-	-	-	-	-	-
5	23	-	-	3	-	-	0	-	9	11	-	-	-	0	0	-	-	-	-	-	-	-	-	-	-
6	24	-	23	1	17	-	3	-	10	-	-	-	-	-	0	0	-	-	-	-	-	-	-	-	-
7	25	-	-	8	-	-	-	7	18	-	-	0	-	-	-	-	0	0	-	-	-	-	-	-	-
8	13	24	-	-	0	-	8	-	6	-	-	-	-	-	-	-	-	-	0	0	-	-	-	-	-
9	7	20	-	-	16	22	10	-	-	23	-	-	-	-	-	-	-	-	-	0	0	-	-	-	-
10	11	-	-	-	19	-	-	-	13	-	3	17	-	-	-	-	-	-	-	-	0	0	-	-	-
11	25	-	8	-	23	18	-	14	9	-	-	-	-	-	-	-	-	-	-	-	-	0	0	-	-
12	3	-	-	-	16	-	-	2	25	5	-	-	1	-	-	-	-	-	-	-	-	-	0	-	0

Fig. 1. The parity check matrix suggested in IEEE 802.11n standard

The row reordering of a matrix makes no change on codeword. In contrast, the column reordering makes the bit-level re-permutation in one codeword. To preserve the original meaning, we cannot swap round the column without swapping the corresponding row. Re-ordering the matrix can lead to more efficient computation arrangement. However, re-ordering (or in another word, permutation) can result in some extra buffers required for input sequence permutation and output right order retrieval. Shih et. al. [5] has proposed an attractive arrangement as the one shown in Fig. 2 below.

In a conventional LDPC decoding with min-sum algorithm, it is assumed that all rows must be processed before starting column processing. Consider the matrix shown in Fig. 1, the non-overlapping scheme is shown in Fig 3 a) where 3×27 CNU and 4×27 VNU are assumed. This scheme takes 10 cycles per iteration. When overlapping is considered (see also Fig. 3 b), while performing R7-R12 we can compute C14-C21 in the same time since row computation has no effect to the column

computation. However with matrix re-ordering as discussed above, a possible resulted matrix is shown in Fig.2. While computing row 9-12 one can also perform column 1-8 computation. Similarly row 1-4 can be processed in parallel with column 17-24. Lower number of rows and/or columns parallel processing are also made possible.

New	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		
Old	16	17	18	14	15	7	19	20	1	3	4	5	9	10	11	12	2	6	8	13	21	22	23	24		
1	3	0	-	-	0	-	-	-	6	0	-	10	24	-	0	-	-	-	-	-	-	-	-	-	-	
2	4	0	0	-	-	-	-	-	2	-	0	20	25	0	-	-	-	-	-	-	-	-	-	-	-	
3	5	-	0	0	-	-	-	-	23	-	-	3	0	-	9	11	-	-	-	-	-	-	-	-	-	
4	6	-	-	0	-	-	3	0	-	24	23	1	17	10	-	-	-	-	-	-	-	-	-	-	-	
5	1	-	-	0	-	-	-	-	0	-	-	0	0	-	0	-	0	-	1	-	-	-	-	-	-	
6	2	-	-	0	0	0	-	-	22	-	-	17	12	-	-	0	-	0	-	-	-	-	-	-	-	
7	7	-	-	-	-	0	0	25	-	-	8	7	18	-	-	-	-	-	0	-	-	-	-	-	-	
8	8	-	-	-	-	8	-	0	13	-	-	0	6	-	-	24	-	-	0	-	-	-	-	-	-	
9	9	-	-	-	-	-	-	7	-	16	22	23	-	-	-	20	10	-	-	0	0	-	-	-	-	
10	10	-	-	-	-	-	-	-	11	-	-	19	13	-	3	17	-	-	-	-	-	0	0	-	-	
11	11	-	-	-	-	-	-	-	25	8	-	23	9	-	-	-	-	18	14	-	-	-	-	0	0	
12	12	-	-	-	-	-	-	-	3	-	-	16	25	5	-	-	-	-	2	1	-	-	-	-	0	-

Fig. 2. Matrix re-ordering that enables row operation and column operation to be overlapped [5]

According to the overlapped computation (or parallel on another hand), the number of cycles per iteration is reduced (8 cycles per iteration in this case). A bit more wisely, if we can arrange proper sequences of rows and columns we can then have higher degrees of overlapping. The computing units (VNUs and CNUs) can be made busy most the times. This is shown in Fig. 3c).

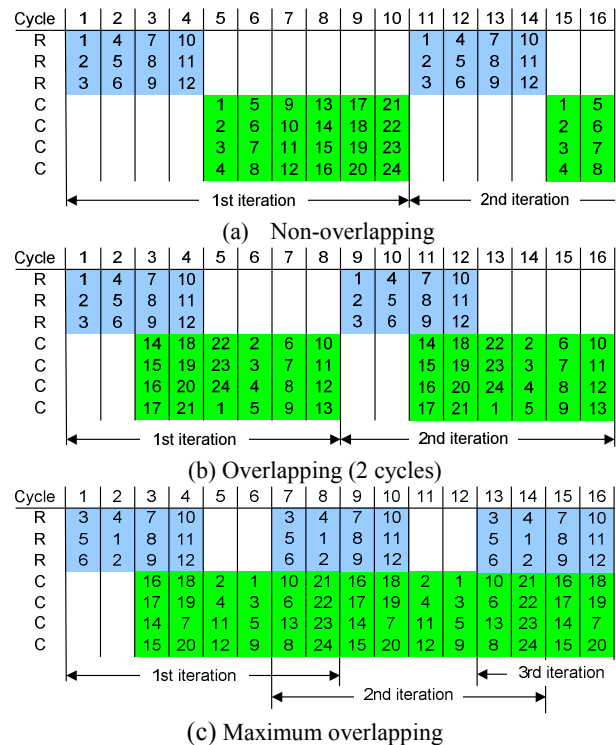


Fig. 3. Overlapping timing diagrams by 3 rows and 4 columns

Although Shih et.al. have firstly applied the re-ordering technique to IEEE802.16e system, their technique can be

TABLE I
Comparison of Hardware Utilization and Corresponding Operation Cycles (20 iterations)

Arrangement (CNU:VNU)	Shih [5]			This paper		
	4:8	3:6	2:4	4:8	3:4	2:4
Non-overlapped utilization	50%	50%	50%	50	50%	50%
Overlapped utilization	75%	66%	75%	60%	CNU 100% VNU 66%	85.7%
Non-overlapped Cycles	27*6*20	27*8*20	27*12*20	27*6*20	27*10*20	27*12*20
Overlapped Cycles	27*(4*20+1)	27*(6*20+1)	27*(8*20+2)	27*(5*20)	27*(6*20+2)	27*(7*20+2)
Reduce ratio	32%	24%	32%	16%	39%	40%

easily applied to IEEE802.11n. Their arrangement is shown previously in Fig. 2. Since we are focusing on the highly parallel hardware for high throughput, the best performance cannot be obtained for the lower number of computing units. We hereby propose a different arrangement that could be an alternative solution, in particular when hardware size is a case and limited factor. Our re-ordered matrix is shown in Fig. 4. Be noted that Shih’s design can support 4:8, 3:6, and 2:4; CNU:VNU while ours supports 4:8, 3:4, and 2:4; CNU:VNU.

of cycles. We cannot beat [5] in the case of 4:8 arrangement. However, what is out performed for our arrangement is that we can make more effectively use of the processor. The lower number of cycles implies the higher number of overlapping cycles. Therefore, the processors are always busy. The exact numbers are in Table I.

4. ARCHITECTURES

Min-sum decoders typically required 4 hardware units; CNU, VNU, memory, and control units. The performance of the implementation is usually addressed by number of VNUs and CNUs as well as their computational arrangement. The word length also affects the result precision. We use the word length of 6 bits in this work. Memory organization and control unit are omitted in this paper.

A. CNU Architecture

Check node update or row operation basically implements eq. (2) shown in the previous section. A block diagram of a CNU is shown in Fig. 6. To find the minimum value of input data, the data are converted 2’s complement number system. To ease the magnitude comparison, sign bits are considered separately. The sign is included again at the final state.

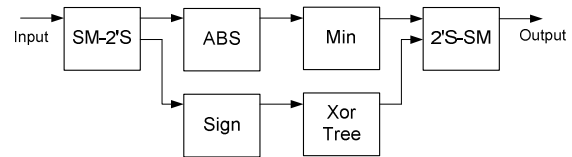


Fig. 6. CNU Block Diagram

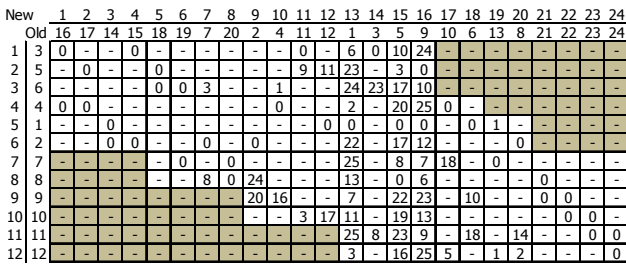


Fig. 4. Re-ordered H Matrix proposed by this paper

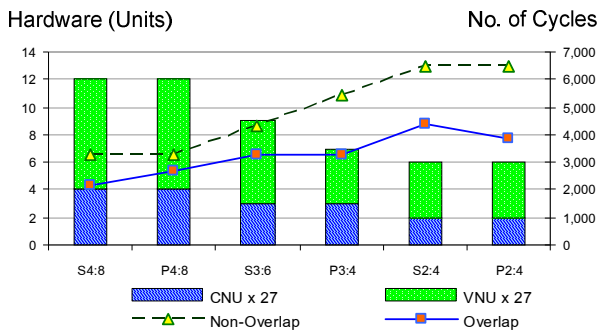


Fig. 5. Comparable hardware unit and cycle time proposed by Shih’s [5] and the technique proposed in this paper

Shown in Fig. 5 are the comparative summary of our design and a similar design proposed in [5] where bar graph denotes the hardware counts while line graph denotes cycle counts. Toward each arrangement, obviously our 3:4 requires lower number of VNUs when compared to its candidate 3:6 proposed by [5]. Both arrangements have the same number of cycles when they are used in the none-overlapping mode. In the overlapping mode our arrangement can have similar or better number

For minimum value searching, we borrowed the one proposed by Zarubica et.al. in their ASIC design [6] for our FPGA implementation. Shown in Fig. 7, we use twenty 2-to-1 comparators. Each row of the matrix shown in Fig. 1 may hold 7 or 8 non-zero contents. For 7 non-zero contents, one input is forced maximum and its corresponding output is ignored.

Sign update circuit is fairly simple as it has to determine all signs multiplication. Basically it is checking whether number of positive signs and negative signs,

which one is greater. This can be implemented using only XOR gates.

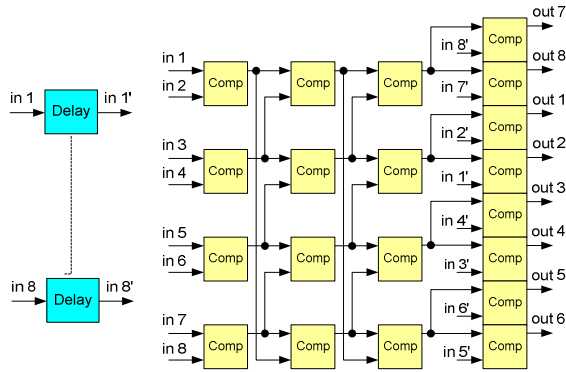


Fig. 7. A check node magnitude update schematic (max.= 8 contents)

B. VNU Architecture

The architecture of a VPU (column operation) is shown in Fig. 8 below. For hardware regularity 12 inputs (max. number of vertical contents) are all utilized. There are summed and added to $L(c_i)$. The considered output must be subtracted out by the corresponding input.

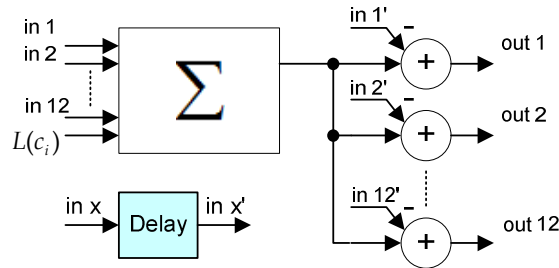


Fig. 8. VNU Block Diagram

5. FPGA EVALUATION

According to the architectures proposed in the previous section, the CNU and VNU were modeled in Verilog and simulated using ModelSim. Subsequently, the designs are synthesized and performed place and route using Xilinx ISE 10.1 software package. The FPGA utilization summarized in table II is based on the Xilinx report. Our result is not fully completed since memory and control are not included. Hence VNU and CNU are given in details.

TABLE II
HARDWARE SUMMARY

XILINX UTILIZATION SUMMARY (3s400tq144-5)		
	VNU	CNU
Number of slices	226	259
Number of slice flip flops	408	344
Number of 4 input LUTs	355	468
Maximum frequency (MHz)	230.23	198.92

6. CONCLUSION

In this paper we have presented an alternative overlapping technique for a LDPC codes with min-sum decoder. The basic idea of this implementation is based on the specific structure of IEEE802.11n parity check matrix. The proposed technique can offer superior performance over [5] in particular when the arrangement holds lower degree of parallelism. For the processing units to be better synchronized, VNU and CNU could be further tuned for hardware complexity and/or operating speed.

7. ACKNOWLEDGMENTS

This work is partially supported by Industry/University Cooperative of Data Storage Technology and Applications Research Center (DSTAR, KMITL), King Mongkut’s Institute of Technology Ladkrabang and National Electronic and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSTDA) under scholarship HDD-01-52-05D.

REFERENCES

- [1] R. G. Gallager, “Low-density parity check codes,” *IRE Trans. Info. Theory*, vol. IT-8, pp. 21–28, 1962.
- [2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low-density parity check codes," *Electron. Lett.*, vol. 32, pp. 1645, 1996.
- [3] IEEE, “IEEE Standard for Information technology -- Telecommunications and information exchange between systems -- Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput”, *IEEE 802.11n-2009*.
- [4] Y. Chen and K.K. Parhi, “Quasi-cyclic low-density parity-check coded multi-band-OFDM UWB systems” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 51, no. 6, pp. 1106-1113, Jun. 2004
- [5] X. Y. Shih, C. Z. Zhan, C.H. Lin and A.Y. Wu, “An 8.29 mm² 52 mW Multi-Mode LDPC Decoder Design for Mobile WiMAX System in 0.13 μm CMOS Process” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 672-683, Mar. 2008
- [6] R. Zarubica, S. G. Wilson, and E. Hall, “Multi-Gbps FPGA-based Low Density Parity Check (LDPC) Decoder Design” *IEEE Global Telecommunications Conference (GLOBECOM) 2007*, pp. 548-552. Nov. 2007
- [7] I. Park and S. Kang, "Scheduling Algorithm for Partially Parallel Architecture of LDPC Decoder by Matrix Permutation" *IEEE Int. Symp. on Circuits and Systems (ISCAS) 2005*, vol. 6, pp. 5778-5781, May. 2005
- [8] G. Malema, “A Flexible and Overlapped QC-LDPC Decoder” *Signal Processing, 2008. ICSP 2008. 9th International*, 2008, pp. 1739 – 1743. 2008.
- [9] Hiroyuki SHIMAJIRI, Akifumi HAYASHI and Takeo YOSHIDA, “RTL design of LDPC decoder for IEEE802.11n WLAN” *ISCIT 2009. 9th International Symposium on Communications and Information Technology, 2009*, pp. 302-307. Sep. 2009