

DVB-T2 LDPC Decoder with Perfect Conflict Resolution

XIONGXIN ZHAO^{1,a)} ZHIXIANG CHEN¹ XIAO PENG¹ DAJIANG ZHOU¹
SATOSHI GOTO¹

Received: May 27, 2011, Revised: September 2, 2011,
Accepted: October 19, 2011, Released: February 21, 2012

Abstract: Currently most of LDPC decoders are implemented with the so-called layered algorithm for its implementation efficiency and relatively high decoding performance. However, not all of structured LDPC codes can be implemented with the layered algorithm directly because of the message updating conflicts within layers in the a-posteriori information memory. In this paper we focus on the resolution of this kind of conflicts for DVB-T2 LDPC decoders. Unlike the previous resolutions, we directly implement the layered algorithm without modifying the parity-check matrices (PCM) or the decoding algorithm. DVB-T2 LDPC decoder architecture is also proposed in this paper with two new techniques which guarantee conflict-free layered decoding. The PCM Rearrange technique reduces the number of conflicts and eliminates all of data dependency problems between layers to ensure high pipeline efficiency. The Layer Division technique deals with all remaining conflicts with a well-designed decoding schedule. Experiment results show that compared to state-of-the-art works we achieve a slight error-correcting performance gain for DVB-T2 LDPC codes.

Keywords: LDPC, DVB-T2, layered decoding, message updating conflict

1. Introduction

Low-density parity-check (LDPC) codes [1], which are firstly invented by Gallager in 1960s, recently as error-correcting codes (ECC), are receiving much attention in both academic and industrial areas because of their excellent error-correcting performance and parallelization with relatively low implementation complexity for decoders. The iterative decoding process of LDPC codes is originally based on the two-phase message-passing algorithm (TPMP) [1], which contains check-node message updating and variable-node message updating. In 2003, Mansour introduced the concept of turbo-decoding message-passing (TDMP) [2], which is also called layered algorithm, for his proposed structured or architecture-aware LDPC (AA-LDPC) codes, enables up to twice convergent speed due to more frequent a-posteriori message updating.

On the other hand, considering the implementation complexity for encoders and decoders, many structured LDPC codes are proposed. Two important types of structured LDPC codes are recently adopted by most of communication systems. The irregular-repeat-accumulate (IRA)-LDPC codes [3], which benefit from linear encoding complexity and relatively simple structure, have already been included in long distance communication systems such as DVB-T2 [4], DVB-S2 [5] and ISDB-S2 [6]. Another type of structured codes, the quasi-cyclic (QC)-LDPC codes [7], have already been adopted in many communication standards not limited to the well-known WiMAX and Wi-Fi standards. Moreover, the structured QC-LDPC codes make it effectively suitable for

partial-parallel decoder implementation with the layered algorithm.

For the structured IRA-LDPC codes, many researchers also attempt to implement with the layered algorithm for efficiently partial-parallel decoding [8], [9], [10], [11], [12]. However, some IRA-LDPC codes cannot be implemented with layered algorithm directly because of the so-called message updating conflict problem which appears while dividing the PCM into layers. Directly ignoring the conflicts will cause a lot of cutting edge problems in layered decoding which will be discussed in Section 2 in detail. The error-correcting performance degradation can be larger than 0.2 dB in SNR (signal-to-noise ratio) compared with conflict-free layered decoding performance [12]. To solve this problem, authors in Ref. [8], [9] decoded the conflict layers with TPMP algorithm while authors in Ref. [10], [11] tried to modify the PCM of conflict layers. The authors in Ref. [12] proposed a selective recalculation strategy applied to the layered algorithm to achieve conflict-free error-correcting performance, but the algorithm is not hardware-friendly.

In this paper, we focus our attention on the resolution of message updating conflicts for DVB-T2 LDPC codes. Unlike the previous resolutions, we can directly implement the layered algorithm without modifying the PCM or the decoding algorithm. Two new techniques are proposed to guarantee conflict-free layered decoding performance. Firstly, the PCM Rearrange technique efficiently reduces the number of conflicts with a reasonable parallelism and eliminates all of data dependency problems between layers to ensure high pipeline efficiency. Secondly, the Layer Division technique solves all remaining conflicts with a well-designed decoding schedule. A synthesizable DVB-T2

¹ Waseda University, Kitakyushu, Fukuoka 808–0135, Japan

^{a)} zxx@moegi.waseda.jp

LDPC decoder architecture is also introduced in this paper to demonstrate detail implementations.

The remainder of this paper is organized as follows: Section 2 briefly describes the layered decoding algorithm and the arising message updating conflict problem when applying to DVB-T2 LDPC codes. In Section 3, the proposed PCM Rearrange and Layer Division techniques are discussed in detail. The proposed decoder architecture is presented in Section 4. Section 5 demonstrates the comparison results of error-correcting performances and implementation results, while Section 6 concludes the paper.

2. Message Updating Conflict in Layered DVB-T2 Decoder

In this section, firstly, the LDPC code structure adopted in DVB-T2 is introduced. Secondly, the layered algorithm for partial-parallel decoding is shown. Thirdly, the arising conflicts while applying layered algorithm to DVB-T2 LDPC codes is discussed.

2.1 LDPC Codes in DVB-T2 Standard

An LDPC code is defined by a sparse parity-check matrix (PCM) H of size $M_H \times N_H$. It can be represent by a bipartite graph which consists of two types of nodes, the variable nodes and check nodes. The check node i is connected to variable node j when element h_{ij} in matrix H is 1. No connection exists when h_{ij} is 0. DVB-T2 is the second generation digital terrestrial television broadcasting system. In such kind of channel conditions, the Forward Error Correction (FEC) technique adopted in the system is designed to provide a “Quasi Error Free” (QEF) quality target, approximately with a bit error ratio $BER < 10^{-9}$. The code lengths for DVB-T2 LDPC codes are 64,800 bits for normal frame and 16,200 bits for short frame. For different channel conditions, there are 6 and 7 kinds of code rates for normal and short frame, respectively.

LDPC codes in DVB-T2 standard belong to systematic IRA-LDPC codes. For DVB-T2 LDPC codes specified with code rate $R = K/N$, the codeword c is defined as $c = (i_0, i_1, \dots, i_{K-1}, p_0, p_1, \dots, p_{N-K-1})$, in which i and p represent information bits and parity bits, respectively. The PCM can be divided into several parts, shown in Fig. 1. Matrix H_A , which represents the connections between information nodes and check

nodes, can be further divided into j M -column sub-matrices. The connections of first column for each sub-matrix H_{A_j} are specified by row j of the corresponding address tables specified in Ref. [4] and the following $M - 1$ columns are generated by cyclically shift a number of q as shown in the figure, which is $q = (N - K)/M$, a code rate dependent constant. It shows the periodicity of M for each sub-matrix H_{A_j} since the first column can be obtained by cyclically shift value q to the M -th column for each H_{A_j} . Matrix H_B , which represents the connections between parity nodes and check nodes, is a staircase matrix.

The periodical feature of the PCM makes it possible for partial-parallel decoding. The fixed periodicity $M = 360$ in DVB-T2 LDPC codes enables decoding parallelism up to 360. Through very simple row reorder based on modulo operations to the row indices, the PCM can be shown as QC-like matrix [9], in which three types of sub-matrices with block size $M \times M$ exist. They are: all-zero blocks, shifted identity blocks and multiple shifted identity (MSI) blocks, as shown in Fig. 2. The MSI blocks with several diagonals, which do not exist in normal QC matrices, leads to message updating conflict problem in layered decoding.

2.2 Partial-parallel Layered Decoding Algorithm

Unlike the TPMP algorithm, the layered algorithm divides each iteration into sub-iterations or layers and utilizes the intermediate a-posteriori probability APP_j for each variable node j to calculate the extrinsic information L_{ij} , which represent messages sent from check node i to variable node j , while in the TPMP algorithm, APP and extrinsic messages are updated only once during each iteration. The frequent updating of APP messages leads to rapid convergence. The decoding procedure of layered algorithm is shown below.

At first, APP_j are initialized with the received channel Log-

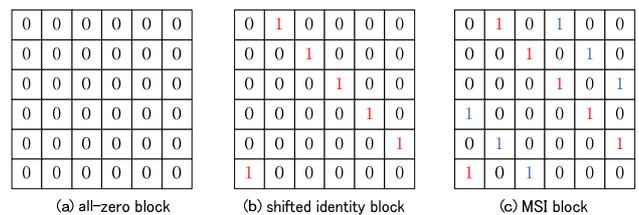


Fig. 2 Three types of blocks in QC-like PCM.

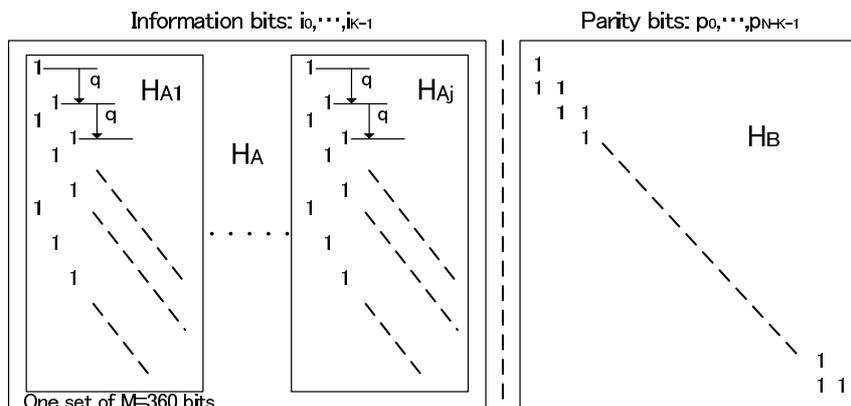


Fig. 1 PCM of DVB-T2 LDPC codes.

likelihood ratio (LLR) and the extrinsic messages L_{ij} are initialized with zeros. For each sub-iteration, the APP_j and L_{ij} messages are updated using the following equations:

$$E_{ij} = APP_j - L_{ij}, \quad (1)$$

$$L_{ij}^{new} = 2 \tanh^{-1} \left(\prod_{j' \in A(i) \setminus j} \tanh(E_{ij'}/2) \right), \quad (2)$$

$$APP_j^{new} = E_{ij} + L_{ij}^{new}, \quad (3)$$

in which E_{ij} are the messages sent from variable node j to check node i while $A(i)$ is the set of variable nodes connected with check node i . Hard decisions of variable node j can be done by evaluating the sign of APP_j after all the layers being processed once (a whole iteration is finished) and then go to the parity-check function. If the decoded codeword \vec{c} satisfies the parity-check function $H \cdot \vec{c} = 0$ or the iteration number reaches the predefined maximum value, the decoding will be terminated. Otherwise the steps with Eqs. (1), (2), (3) will be repeated.

The QC matrices, in which the column weights for each horizontal layer of sub-matrices or “row block” never exceed one, allow paralleled decoding for each row block with appropriate permutation operations to the APP messages of each variable node group [9].

2.3 Message Updating Conflict in DVB-T2

As mentioned above, MSI blocks appear during QC transformation for DVB-T2 LDPC codes. If we consider each row block as one layer in the partial-parallel layered decoding, n diagonals within a block means that APP messages for these variable nodes will be updated n times concurrently. It is equivalent to cut $n - 1$ of n check-to-variable edges for such variable nodes as shown in Fig. 3, which demonstrates an example of cutting edge with both PCM Fig. 3 (a) and bipartite graph Fig. 3 (b) representations for one layer. Check nodes C_0 and C_1 compose one layer and variable nodes V_0 , V_1 and V_2 connect with them. The two red edges shown in Fig. 3 (b) send the updated extrinsic messages L_{00} and L_{10} calculated with Eq. (2) to variable node V_0 for APP_0 updating with Eq. (3) concurrently, but only one of them can participate in the equation. As a result, either of the two edges is cut. This problem is the message updating conflict problem and forces parallel level degradation or performance loss. The error-correcting performance degradation can be larger than 0.2 dB in SNR compared with conflict-free layered decoding performance [12], which is even worse than decoding with TPMP algorithm. To implement the layered algorithm, this problem must be solved first.

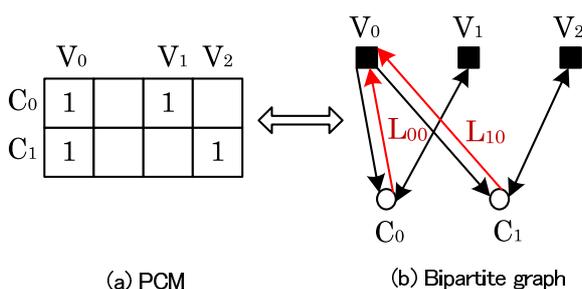


Fig. 3 Example for message updating conflict.

3. Resolution to Message Updating Conflict for DVB-T2

Simply ignoring the conflict problem is undesirable. Many published works showed their solutions by modifying the PCM or decoding algorithm to achieve acceptable decoding performance, which we conclude as approximate resolutions. In this section, we illustrate our resolution to the message updating conflict problem with two techniques. The most important advantage of our proposal is that we exactly apply the layered decoding algorithm and follow the original PCM specified in DVB-T2 for decoder implementations.

3.1 PCM Rearrange

To achieve the minimum decoding throughput of 90 Mbps required by DVB-T2 system, dozens of parallel processors instead of 360 are enough [10]. Splitting the 360×360 blocks is a reasonable way and has already been applied in many works, since it can not only reduce the number of conflict blocks but also simplify the hardware. The periodicity $M = 360$ of DVB-T2 LDPC codes makes it possible to split into various block sizes. For convenient explanation, we introduce parameter ϵ , which is the integer factors of 360, helping us reorder the original PCM into smaller block QC-like matrices, in which the corresponding block size is $P = M/\epsilon$. We can deal with each sub-matrix H_{Ai} separately and finally put them together. Moreover, the barrel shift property of matrix H_{Ai} helps us obtain the simple reordering strategy. It transforms sub-matrix H_{Ai} to $\epsilon \cdot q$ by ϵ seed matrix H_{Ai}^* with block size of $P \times P$. Then let us define j and k as the row and column indices for the blocks in seed matrix H_{Ai}^* and S_{jk} as their corresponding left barrel shift values. Each number X of the corresponding row of the address table generates a number of shifted identities which can be calculated using Eq. (4) as follows:

$$\begin{cases} \text{for } k = 0, 1, \dots, \epsilon - 1 \\ j_k = (X + kq) \bmod (\epsilon q) \\ S_{jk} = \lfloor (X + kq)/(\epsilon q) \rfloor \end{cases} \quad (4)$$

The remaining conflict numbers based on different parameters ϵ for all code rates in DVB-T2 long frame can be counted exactly, shown in Table 1. PCM which includes 3-diagonal conflict blocks are marked with * in the table and each 3-diagonal conflict is counted as three.

Two important properties can be observed from Table 1. One is that with the decrease of block size, the number of conflicts reduces significantly. Another is that even if the block size is reduced to unacceptable values, the conflicts cannot be eliminated. In our design, ϵ is selected with 9 and this special number will help with the resolution of remaining conflicts, which will be dis-

Table 1 Number of conflicts in DVB-T2 normal frame.

R	ϵ											
	1	2	3	4	5	6	8	9	10	12	15	18
1/2	8	4	2	2	0	1	0	2	0	1	0	1
2/3	12	5	5	4	2	2	3	2	0	1	2	1
3/5	38*	19	16	8	8	6	2	4	4	4	4	1
3/4	24*	10	8	3	3	3	3	3	2	0	0	2
4/5	37*	15	15	6	9	5	3	4	2	4	3	1
5/6	44*	21	12	13	11	3	5	2	5	2	6	1

cussed in Section 3.2.

The staircase matrix H_B can be also transformed to QC matrix by applying the same reorder scheme to its columns. Unfortunately, the QC-like PCM generated with Eq. (4) is not friendly for partial-parallel decoder. Firstly, many variable node blocks are shared by some pairs of successive layers which will cause serious data dependency problem in pipelined partial-parallel layered decoding [13]. Secondly, as a common method for the storage of QC PCM information, all the indices of non-zero blocks (j, k) and the shifted values S_{jk} should be memorized. Thanks to the same row weight for each PCM of DVB-T2, row indices j can be neglected. Therefore, for each non-zero block (j, k), $(\lfloor \log_2(N/P) \rfloor + 1)$ bits and $(\lfloor \log_2 P \rfloor + 1)$ bits are required for storing k and S_{jk} , respectively. On the other hand, the number of non-zero blocks is increased ϵ times since the total edge number will not be changed during reorder. As a result, ϵ times of data for PCM information are required. In the case of $\epsilon = 9$, the PCM information size for all six code rates is about 600 kbits, which is even close to the size of extrinsic memory!

To solve the two problems mentioned above, we prefer to further reorder the row blocks in PCM generated with Eq. (4). Considering the ϵ identities generated by Eq. (4), if the information for one identity is calculated, that of other identities can be calculated with it. Furthermore, these ϵ identities are in different rows and columns since the j and k of them are all different. Therefore, storing information for one identity is enough instead of storing all ϵ identities. The main idea of the proposed PCM Rearrange method is that we can put the generated identities in successive ϵ rows to solve the data dependency problem between layers and reduce the huge PCM information to $1/\epsilon$ by only storing PCM information for each one of ϵ layers.

From Eq. (4), by modulo and divide q to the row indices we can derive Eq. (5) as follows:

$$\begin{cases} \text{for } k=0, 1, \dots, \epsilon - 1 \\ j_k^* = j_k \bmod q = [X \bmod (\epsilon q)] \bmod q = X \bmod q \\ r_k^* = \lfloor j_k/q \rfloor = \lfloor [X \bmod (\epsilon q)]/q \rfloor + k \bmod \epsilon \\ \quad = (\lfloor X/q \rfloor + k) \bmod \epsilon \end{cases}, \quad (5)$$

in which we can find that j^* do not depend on k , and r_k^* are different for each k . Therefore, we treat each $\epsilon \times \epsilon$ blocks in the seed matrix H^* as one macro-block (MB) and for a given number of X , the corresponding row index of MB and the positions inside MB for the ϵ identities are defined with j^* and (r^*, k) through Eq. (5), respectively. Finally, for the b -th row of number X in the address table, the indices (J, K) and shift values S_{JK} in the seed matrix H_A^* can be generated with Eq. (6) as follows:

$$\begin{cases} \text{for } k = 0, 1, \dots, \epsilon - 1 \\ J = (X \bmod q) \times \epsilon + (\lfloor X/q \rfloor + k) \bmod \epsilon \\ K = b \times \epsilon + k \\ S_{JK} = \lfloor (\lfloor X/q \rfloor + k)/\epsilon \rfloor \end{cases}. \quad (6)$$

Figure 4(a) shows a non-zero MB example in the rearranged seed matrix H_A^* which is generated in the case of $\epsilon = 9$. In this figure, the blanks are all-zero blocks and the numbers are left shift values for the shifted identities. Furthermore, the data dependency problem between successive blocks due to the staircase

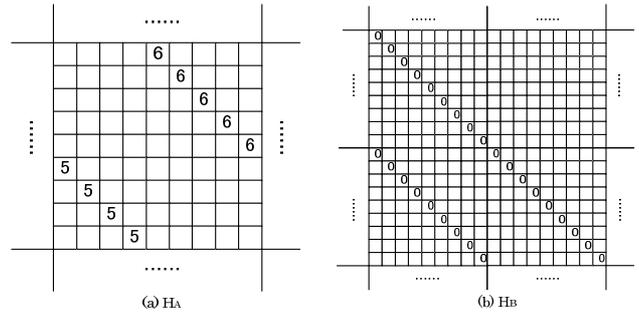


Fig. 4 Seed matrix of DVB-T2 after rearrangement.

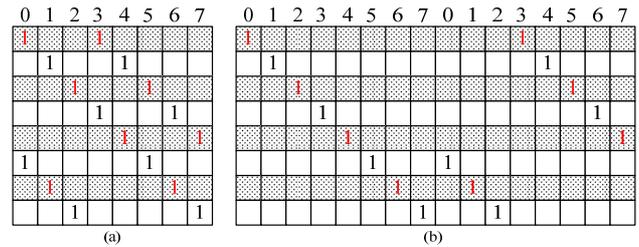


Fig. 5 Layer Division for 8×8 conflict block.

feature is also eliminated in the rearranged seed matrix H_B^* , which is shown in Fig. 4 (b).

With this proposed PCM rearrange scheme, only information of one non-zero block should be memorized for each $\epsilon \times \epsilon$ non-zero MB, and at the same time the data dependency problem between successive layers can be solved to ensure the pipeline efficiency in layered decoding.

3.2 Layer Division

The message updating conflicts cannot be eliminated with rearranging the PCM only. In order to deal with the remaining conflicts after the PCM rearrange scheme, we introduce the Layer Division technique which selectively divides check nodes of the conflict layer into two sub-layers to avoid updating the same APP message concurrently while maintaining the parallelism of decoding.

Let us take 8×8 conflict block which is shown in Fig. 5 (a) as an example to explain the mechanism of this technique. Commonly, in the pipelined partial-parallel decoding, this block should be treated as two shifted identities taking part in the message updating in the corresponding layer, which share the same messages APP_0 to APP_7 , as shown in Fig. 5 (b). The later updated APP messages will overwrite the earlier ones without using them thus it causes message updating conflict problem. By simply dividing the eight check nodes of this conflict layer into two groups which are painted white and dark as shown in Fig. 5, we can ensure that in each group of check nodes the messages APP_0 to APP_7 are just updated once. For the decoding process of these two sub-layers, we firstly decode the dark sub-layer to update the messages APP_0 to APP_7 through red-marked edges, then use these newly updated messages decoding the white sub-layer to obtain final APP messages.

However, in order to apply this technique to solve all the remaining conflicts for DVB-T2 LDPC decoders, some details should be further considered. Firstly, we must ensure that there

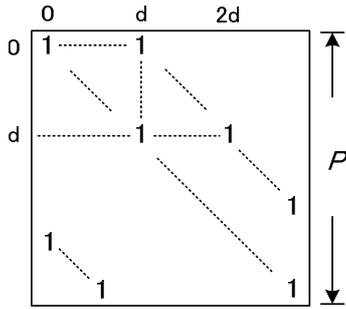


Fig. 6 Example of (P,d) conflict pattern.

are available divisions for all of the conflict blocks in rearranged DVB-T2 matrices. Secondly, the decoding process for the two sub-layers should be maximally overlapped to avoid half of the parallel processors from idling.

A. Division Patterns for Conflict Blocks of DVB-T2

Whether a conflict block can be successfully divided depends on two values: the block size P and the distance of two diagonals which is defined with Eq. (7) as follows:

$$d = |S_1 - S_2|, \tag{7}$$

in which S_1 and S_2 are shift values for the two diagonals. Obviously, P must be an even number to ensure equal divisions and $d \in D = \{1, 2, \dots, P/2\}$ is enough to cover all the cases of conflict patterns. In order to facilitate the explanation, we barrel shift this block to make one of the diagonals to the left as shown in Fig. 6, which will not affect the result.

Let us suppose these P rows can be divided into two sets: SET_1 and SET_2 , and initially put $Row(0)$ into SET_1 . Since both of $Row(d)$ and $Row(0)$ contain elements in $Column(d)$, $Row(d)$ must belong to SET_2 to avoid conflicts. Similarly, $Row(2id \bmod P)$ belong to SET_1 while $Row((2i + 1)d \bmod P)$ belong to SET_2 with increasing integer i . This recursion of allocation will be finished when the coefficient $k = 2i$ or $k = 2i + 1$ can make $kd \bmod P$, which forms a loop of rows related to $Row(0)$ which are included in the two sets. If k is an even number, two sets do not share the same column elements, so that it is an available division. Otherwise, $Row(0)$ will be included by two sets, which means the block with such parameters cannot be successful divided. The remaining rows also form the same kind of loops, and can be allocated with the same strategy. In order to cover all the cases with minimal division patterns, we selectively distribute these rows into the two sets based on the division algorithm as shown in Fig. 7.

U_d is the set of d values which cannot be divided. For the rearranged matrices of DVB-T2 normal frame, simulations are processed based on this algorithm to check whether all remaining conflict blocks can be divided for different ϵ values. The main reason to select $\epsilon = 9$ in our design is that the elements of set U_d do not exist in the remaining conflict blocks when $P = 40$. The three division patterns and their corresponding d values for $P = 40$ are shown in Fig. 8 (a), (b) and (c), which are enough to cover all the conflict blocks.

B. Pipelined Conflict Layer Decoding with Maximally Overlapping

After the conflict layers are successfully divided, the remain-

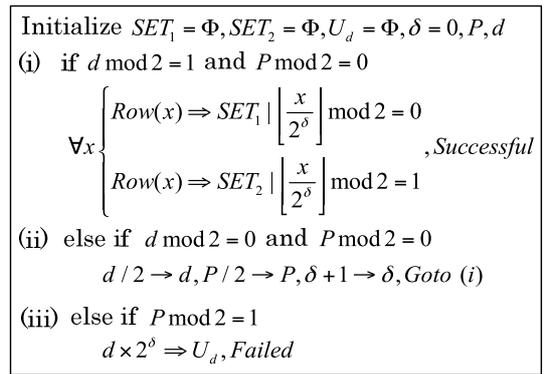


Fig. 7 Division algorithm for (P,d) conflict block.

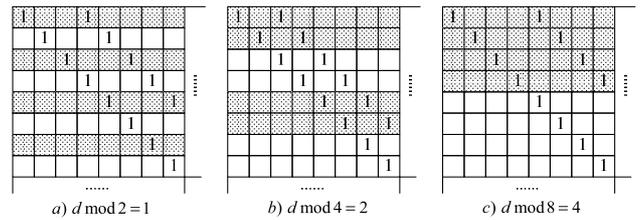


Fig. 8 Three division patterns and corresponding d values for $P = 40$.

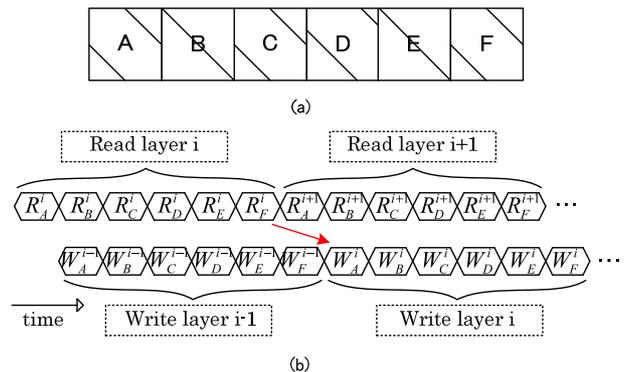


Fig. 9 Timing diagram of fully-overlapped pipeline for normal layers.

ing work is to maximally overlap the pipelined decoding for the two sub-layers. If not, the decoding time for conflict layers will be doubled and lead to serious performance degradation.

Before we demonstrate our resolution, let us review the timing diagram of fully-overlapped pipelined processing with an example of normal QC layer, in which the layer pattern with a row weight of six is shown in Fig. 9 (a). Figure 9 (b) shows the corresponding timing diagram in which the read operation means messages are read into processors and the write operation means messages are written back to memories. There is a one-cycle delay marked with red arrows between read and write operations for the same layer due to the delay of FIFO registers in the processor which is named as the ‘‘FIFO-delay.’’

However, there are data dependencies between the two sub-layers while decoding a conflict layer. The most critical one is that for messages of the conflict block, the second sub-layer must wait for APP updating results updated by the first sub-layer. Therefore, we put the conflict block in the front of the first sub-layer but put it at the end of the second sub-layer to reduce the overall clock cycles for read operations. And we suppose that we can bypass the updated APP messages to the input port of the pro-

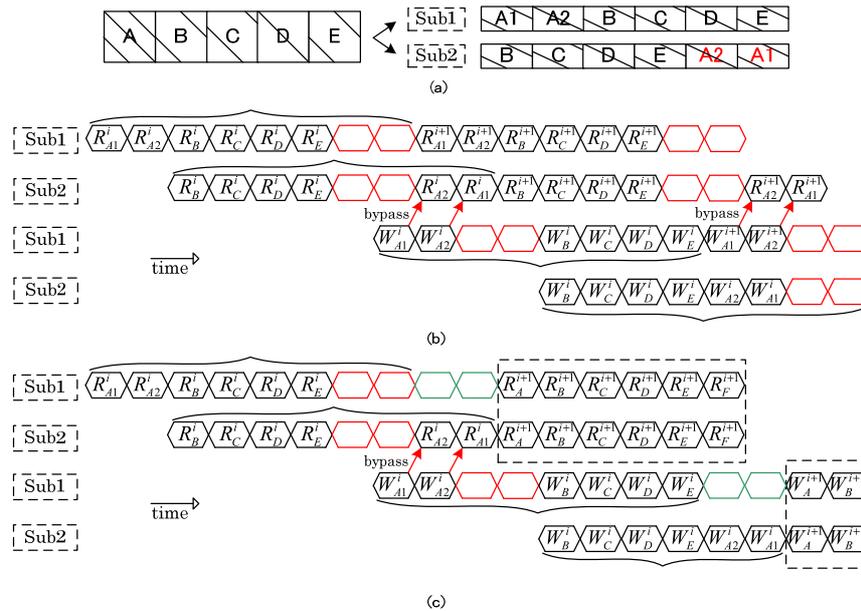


Fig. 10 Timing diagram of overlapped pipeline for conflict layers.

cessor for read operations with one-cycle delay, which is named as the “Bypass-delay.” The *APP* messages required by block A2 in the second sub-layer are just the *APP* messages updated by block A1 from the first sub-layer, and vice versa.

Furthermore, we should process each normal block for the two sub-layers simultaneously to avoid accessing the memories twice for each block. The conflict layer pattern is shown in Fig. 10 (a), in which the blocks A1 and A2 are separated from the conflict block A and the processing orders of them in the two sub-layers are different. Based on all above criteria, the timing diagram for successive conflict layers is shown in Fig. 10 (b), in which the pipelined processing is overlapped. In this figure, the FIFO-delay and Bypass-delay totally lead to a two-cycle delay for the read operations of *APP* messages of blocks A2 and A1 in the second sub-layer, which are marked with red blanks. We should notice that the read operations of blocks A1 and A2 in the first sub-layer for the next layer can be overlapped with the read operations of blocks A2 and A1 in the second sub-layer for the current layer since the latter messages are from the bypass. So that only two extra clock cycles are required for decoding each conflict layer compared to normal layers. For DVB-T2 normal frame codes, the worst case, which occurs in rate 4/5, is 1.2% decoding time redundancy which can be neglected. On the other hand, if a conflict layer is followed by normal layers or conflict layers with different division patterns, extra two clock cycles are required as shown in Fig. 10 (c) marked with green blanks. But this situation at most happens three times for each iteration so that it can be omitted, since the conflict layers can be put together by simply reordering the PCM.

4. Proposed DVB-T2 LDPC Decoder Architecture

In this section we introduce the partial-parallel layered LDPC decoder architecture based on the proposed conflict resolution, which can support all the six code rates for DVB-T2 normal

frame. We will focus on the particular function modules and logic elements which support the proposed decoding strategy.

4.1 Top-level Decoder Architecture

Figure 11 demonstrates the top-level decoder architecture based on the conflict resolution described in Section 3. This 40-parallel layered decoder can be divided into several modules which are listed as follows:

- (1) The controller module (CTRL), which generates all of the control signals and addresses for memories and other function modules, can control the decoding of parallel processors in different statuses.
- (2) The PCM-ROM single-port memory module, which stores all code rates of DVB-T2 PCM information, only requires one ninth of memory size compared to general PCM storing method which benefits from the proposed PCM Rearrange technique.
- (3) The *APP*-MEM dual-port memory module, which is initialized with received channel LLR, stores all the *APP* messages for updating.
- (4) The information generating unit (IGU), which is included in the controller module, recovers all PCM information.
- (5) The pattern decoder (PD), which is also included in the controller module, detects whether a group of nine layers are conflict layers and generates a 40-bit division pattern based on conditions as shown in Fig. 8.
- (6) The permutation networks PN0 and PN1 are 40-way MUX-based barrel shifters. Notice that PN0 is a combinational circuit while PN1 contains flip-flops for *APP* message recovery which is necessary for conflict blocks.
- (7) The processor elements (PE) 0-39 are parallel processors for *APP* and extrinsic message updating.
- (8) The Word-MEM single-port memory with 40 cuts, which is included in each PE, stores the compressed extrinsic messages without sign bits.

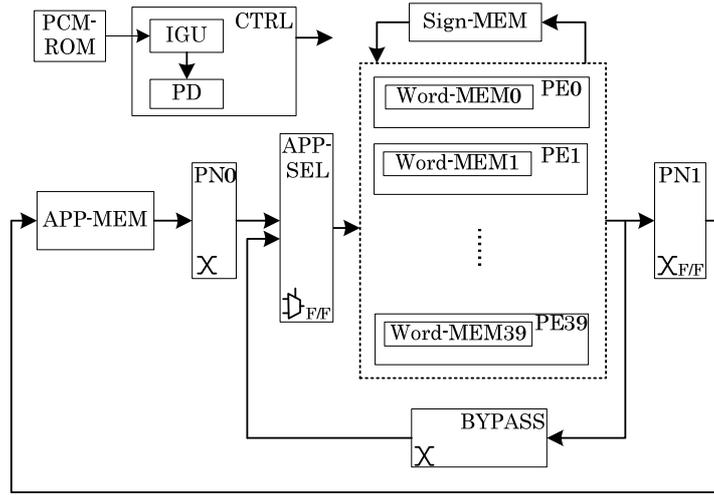


Fig. 11 Top-level decoder architecture.

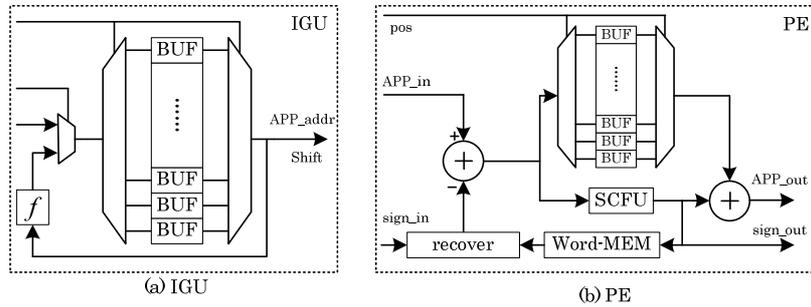


Fig. 12 (a) IGU structure and (b) PE structure.

- (9) The Sign-MEM dual-port memory module stores all the sign bits of extrinsic messages.
- (10) The bypass unit (BYPASS), which follows the assumption in Section 3, contains a 40-way barrel shifter to correctly transfer the *APP* messages from the first sub-layer to the second sub-layer with a shift value equal to $S_2 - S_1$.
- (11) The *APP* selector (APP-SEL) module, which is composed of multiplexers and flip-flops, makes selections of *APP* messages for inputs between PN0 and BYPASS outputs. One level of flip-flops is implemented to balance the critical path while achieving the one-cycle Bypass-delay.

In this design, for investigating the feasibility of proposed decoding strategy, input and output buffers are not contained. The extrinsic message updating is simply based on Normalized Min-sum function as shown in Eq. (8) with the normalized factor equal to 0.75.

$$L_{ij}^{new} = \gamma \times \prod_{j' \in A(i) \setminus j} \text{sign}(E_{ij'}) \times \min_{j' \in A(i) \setminus j} |E_{ij'}| \quad (8)$$

The extrinsic messages are quantized with 6 bits while *APP* messages are quantized with 8 bits to avoid overflows. Instead of storing all the extrinsic messages, memorizing the magnitude of first minimum, the magnitude of second minimum, the position of first minimum and all the sign bits for each check node is enough, in which the first three elements compose a 15-bit word storing in Word-MEM for each processor, based on the quantization and the maximum row weight of 22.

4.2 Particular Function Modules

A. Information Generating Unit (IGU)

The IGU sub-module, which is designed to recover the PCM information of successive nine layers from those of one layer, is shown in Fig. 12 (a). It is composed of 22 buffers for temporarily storing the *APP* addresses A and shift values S for non-zero blocks of one layer, and also a functional node f to calculate the information for the following eight layers. When information of one buffer is fetched, it will be automatically updated through the functional node f with Eq. (9) as follows:

$$(A, S)^{next} = \begin{cases} (A - 8, S - 1), & \text{if } A \bmod 9 = 8 \\ (A + 1, S), & \text{else} \end{cases} \quad (9)$$

B. Processor Elements (PE)

Figure 12 (b) shows the structure of proposed PE, which can exactly follow the timing diagrams as shown in Fig. 9 (b) and Fig. 10 (b), (c) for normal and conflict layers, respectively. As in other designs, a serial check function unit (SCFU) is implemented to calculate the four elements of extrinsic messages. The variable-to-check messages are calculated with the input *APP* messages and extrinsic messages which are recovered by input sign bits and Word-MEM contents. Instead of using FIFO registers, 22 buffers are implemented to temporarily store the variable-to-check messages. In order to deal with the non-input clocks and different decoding statuses of PEs for conflict layer decoding, an input signal 'pos' for each PE, which denotes the position of current input block of the layer or sub-layer, controls the accessing indices of buffers and the calculations in SCFU. When decoding

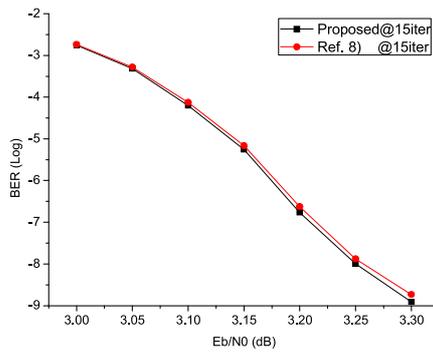


Fig. 13 BER performance comparison of different strategies for rate 5/6 of DVB-T2 normal frame.

to the non-input clocks for conflict layers, dummy information is written into the corresponding buffer controlled by signal ‘pos’ to *push* out the required messages for *APP* updating while the dummy information does not affect the calculations in SCFU.

5. Results and Comparisons

5.1 Performance Simulation Results

Software simulations have been done to demonstrate how much the proposed strategy can improve the BER performance for DVB-T2 LDPC codes compared with that of Ref. [8]. Several state-of-the-art works on DVB-T2 or DVB-S2 LDPC decoding apply the same conflict resolution strategy as Ref. [8] in which the conflict layers are decoded with TPMP algorithm while the other layers are decoded with the layered algorithm. Compared to pure layered decoding which can be only guaranteed by our proposed strategy, the convergent speed is slowed down due to the reduction of *APP* message updating frequency when decoding with TPMP algorithm. **Figure 13** illustrates the simulation results of the BER performance for code rate 5/6 of DVB-T2 LDPC normal frame since it has the most conflict blocks. BPSK (Binary phase-shift keying) modulation and AWGN (Additive white Gaussian noise) channel are simply used throughout the simulations while the quantization and extrinsic message updating function for both strategies are the same as hardware implementation. The maximum iteration number in the simulations is set to 15, which is a relatively small value, since we prefer to clearly demonstrate the differences of performance within acceptable simulation time. From Fig. 13 it can be observed that under the same SNR (Signal-to-noise ratio) or E_b/N_0 values the proposed strategy achieves a slight BER performance gain compared to that of Ref. [8]. At the same channel condition $E_b/N_0 = 3.30$ dB, for totally 200,000 test frames, the number of error bits are 24 and 16 for Ref. [8] and the proposed one, respectively. The difference is not so significant because of two reasons. One is that the conflict layers do not occupy a large proportion in the PCM of DVB-T2 LDPC codes. Another is that the strategy in Ref. [8] still takes the advantage of fast convergence of layered algorithm since the message updating style between layers is still the same as layered algorithm.

5.2 Implementation Results

The decoder core is synthesized based on SMIC 90 nm CMOS technology. Synthesis results obtained with Synopsis Design Compiler are listed in **Table 2**. For code rate 3/5, which has

Table 2 Implementation and comparison results.

Parameter	Proposed	Ref. [9]
Standard	DVB-T2 normal frame	DVB-S2 FEC
Parallelism	40	90
Single-port RAM	553 kb	3.59 Mb
Dual-port RAM	803 kb	
Gate count	193 k	650 k
Max frequency	319 MHz	270 MHz
Max iteration number	25	40
Air throughput	116 Mbps	90 Mbps×2
Check node function	Normalized Min-sum	3-Min
Technology	90 nm	90 nm

the most of edges, the air throughput at the maximum clock frequency can be calculated as 116 Mbps with 25 iterations which meets the 90 Mbps requirement for DVB-T2 system.

It is difficult to compare with other approaches due to different targets, parallelisms or decoding algorithms. There is not any LDPC decoder core design applying the layered algorithm for DVB-T2, so that we compare with the latest DVB-S2 design [9] which also pays attention to solving the message updating conflict problem. In our design, the total gate count is rather small mainly because of the less parallelism since most of the gates are occupied by PE and PN. Compare to Ref. [9] the RAM size is largely reduced because code rates in DVB-T2 LDPC codes are less than DVB-S2 and the extrinsic memory organization in the proposed decoder is also simpler than that of Ref. [9].

6. Conclusion

In this paper, we propose a partial-parallel layered LDPC decoder architecture with a novel conflict resolution to the message updating conflict problem for DVB-T2 LDPC codes. Two techniques are proposed to guarantee conflict-free layered decoding performance with reasonable parallelism and efficient pipeline. Moreover, simulation result shows that compared to state-of-the-art works, the proposed decoder architecture can achieve better error-correcting performance.

Acknowledgments This research was supported by CREST project of Japan Science and Technology Agency.

Reference

- [1] Gallager, R.: Low-Density Parity-Check Codes, MIT Press, Cambridge, MA (1963).
- [2] Mansour, M. and Shanbhag, N.: High-throughput LDPC decoders, *IEEE Trans. VLSI System*, Vol.11, No.6, pp.976–996 (2003).
- [3] Jin, H., Khandekar, A. and McEliece, R.: Irregular Repeat-Accumulate Codes, Second International Conference on Turbo Codes, Brest, France (2000).
- [4] ETSI, *Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2): EN 302 755 V1.2.1* (2010).
- [5] ETSI, *Digital video broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broad-band satellite applications: EN 302 307 V1.1.1* (2005).
- [6] Hashimoto, A. and Suzuki, Y.: A new transmission system for the advanced satellite broadcast, *IEEE Trans. Consumer Electronics*, Vol.54, No.2, pp.353–360 (2008).
- [7] Tanner, R., Sridhara, D., Sridharan, A., Fuja, T. and Costello Jr., D.: LDPC block and convolutional codes based on circulant matrices, *IEEE Trans. Inf. Theory*, Vol.50, No.12, pp.2966–2984 (2004).
- [8] Segard, A., Verdier, F., Declercq, D. and Urard, P.: A DVB-S2 Compliant LDPC Decoder Integrating the Horizontal Shuffle Scheduling, *ISAPCS '06 Proceedings*, pp.1013–1016 (2006).

- [9] Muller, S., Schreger, M., Kabutz, M., Alles, M., Kienle, F. and Wehn, N.: A novel LDPC decoder for DVB-S2 IP, *DATE '09 Proceedings*, pp.1308–1313 (2009).
- [10] Marchand, C., Dore, J.-B., Conde-Canencia, L. and Boutillon, E.: Conflict Resolution by Matrix Reordering for DVB-T2 LDPC Decoders, *IEEE 2009 GLOBECOM Proceedings*, pp.1–6 (2009).
- [11] Marchand, C., Dore, J.-B., Conde-Canencia, L. and Boutillon, E.: CONFLICT RESOLUTION FOR PIPELINED LAYERED LDPC DECODERS, *IEEE 2009 SiPS Proceedings*, pp.220–225 (2009).
- [12] Ji, W., Hamaminato, M., Nakayama, H. and Goto, S.: Data conflict resolution for layered LDPC decoding algorithm by selective recalculation, *2010 CISP Proceedings*, pp.2985–2989 (2010).
- [13] Sun, Y., Karkooti, M. and Cavallaro, J.R.: VLSI Decoder Architecture for High Throughput, Variable Block-size and Multi-rate LDPC Codes, *IEEE 2007 ISCAS Proceedings*, pp.2104–2107 (2007).



Xiongxin Zhao received his B.S. degree in Electronic Science and Technology from Tsinghua University in 2006. He received his M.E. degree in System LSI, from Waseda University, Japan in 2010. He is currently a doctor candidate in Graduate School of Information, Production and Systems, Waseda University. He becomes a Japanese Government Scholarship (MEXT Scholarship) student from 2010. His research interests include VLSI design and ECC codes.



Zhixiang Chen received his B.S. degree in electronic engineering from Shanghai Jiao Tong University, China, in 2007. He received his M.E. degree in System LSI, from Waseda University, Japan in 2010. He is currently working toward a Ph.D. degree in Graduate School of Information, Production and Systems, Waseda University, Japan. He has been a fellowship recipient of Japan Society of the Promotion of Science from April, 2011. His research interests include algorithm and VLSI design for ECC code.



Xiao Peng received his B.S. and M.S. degrees in Electronic Science and Technology from Tsinghua University in 2005 and 2008, respectively. He received his Ph.D. degree in September 2011 from Waseda University. He is currently a post doctor in Waseda University. His research interests include wireless communication and error correcting code, especially the LDPC codes.



Dajiang Zhou received his B.E. and M.E. degrees from Shanghai Jiao Tong University, China. He received his Ph.D. degree in engineering from Waseda University, Japan, in 2010, where he is currently a researcher. His interests are in algorithms and VLSI architectures for multimedia and communication signal processing.

cessing.



Satoshi Goto received his B.E. and M.E. degrees in electronics and communication engineering from Waseda University in 1968 and 1970, respectively. He received his Doctor of Engineering degree from Waseda University in 1981. He joined NEC Laboratories in 1970, where he worked on LSI design, multimedia system and software as GM and VP. Since 2003, he has been a professor at Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan. His main interests are in VLSI design methodologies for multimedia and mobile applications. He has published 7 books and more than 200 technical papers in international journals and conferences. Dr. Goto served as GC of ICCAD and ASPDAC and was a board member of IEEE CAS society. He is a Life Fellow of IEEE, Fellow of IEICE, and member of the Academy Engineering Society of Japan.

tem and software as GM and VP. Since 2003, he has been a professor at Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan. His main interests are in VLSI design methodologies for multimedia and mobile applications. He has published 7 books and more than 200 technical papers in international journals and conferences. Dr. Goto served as GC of ICCAD and ASPDAC and was a board member of IEEE CAS society. He is a Life Fellow of IEEE, Fellow of IEICE, and member of the Academy Engineering Society of Japan.

(Recommended by Associate Editor: *Masanori Hariyama*)