Waseda University Doctoral Dissertation

Research on LDPC Decoder Design Methodology for Error-Correcting Performance and Energy-Efficiency

Xiongxin ZHAO

Graduate School of Information, Production and Systems

Waseda University

February 2014

Abstract

Low-Density Parity-Check (LDPC) codes, as linear block error-correcting codes (ECC), which were invented by Gallager in 1963 and rediscovered by MacKay in 1999, nowadays become a hot topic in a variety of application fields. Due to inherently paralleled decoding algorithm and near Shannon limit error-correcting performance, LDPC codes have been widely adopted in modern communication standards, such as Wi-Fi (IEEE 802.11n), WiMAX (IEEE 802.16e), WPAN (IEEE 802.15.3c) and the second-generation of digital video broadcasting (DVB-x2) family of standards. Moreover, storage systems and optical transmission systems also adopt them for improving the lifetime and reliability of devices or communication distances.

LDPC codes can be decoded in time linear to their code-lengths by using iterative belief propagation (BP) or Two-Phase Message-Passing (TPMP) algorithm which was firstly invented by Gallager involving variable-node and check-node message-updating phases. The independence of calculations among variable-nodes or check-nodes in each phase enables arbitrary parallelism for parallel implementation of decoders. Fully-parallel decoders, in which all the variable-node and check-node processing units are implemented, can achieve the highest parallelism but usually suffer from extremely complicated placement and interconnection. A practical solution is the so-called partial-parallel implementation, in which both or either of variable-node and check-node phases are divided into groups and processed separately to trade-off hardware complexity. Structured LDPC codes such as IRA- (irregular repeat-accumulate) LDPC codes and QC- (quasi-cyclic) LDPC codes are invented for partial-parallel implementation and have been adopted in a variety of wireless/wired communication standards.

On the other hand, for most of application cases, supporting multi-mode (multiple code-lengths and multiple code-rates) decoding in one single decoder is appreciated since the communication channel status may change at all times. However, previous decoders designed with TPMP algorithm largely depend on the codes' properties, which makes the multi-mode decoding almost impossible. Due to the introduction of Layered decoding algorithm by Mansour in 2003, the development of LDPC decoders obtains a tremendous progress. It cannot only achieve up to twice convergence speed compared to traditional TPMP algorithm, but also provide practical solution for the multi-mode partial-parallel decoder architectures. The Layered algorithm applies the same check-node functions as TPMP algorithm but different variable-node message-passing schedule, which can increase the flexibility of decoder for multi-mode decoding. Actually, almost all the multi-mode decoders for modern wireless communication systems are designated based on the Layered algorithm. As multi-mode LDPC decoders, many works are developed from the multi-mode prototype designed by Mansour (JSSC 2006). Y. Sun (ISCAS 2007) presents a partial-parallel Layered LDPC decoder for QC-LDPC codes. S. Muller (DATE 2009) presents a LDPC decoder IP for DVB-S2 standard with the partial-parallel Layered architecture. B. Xiang (JSSC 2011) presents a dual-path partial-parallel Layered LDPC decoder for WiMAX to increase implementation parallelism. X. Peng (A-SSCC 2011) proposes bit-serial Layered LDPC decoder for WiMAX to further increase parallelism.

However, there are still many problems to be solved for decoder design technologies. 1) Previously published early termination (ET) schemes, which is applied tremendously increase the decoding throughput or reduce the power consumption, are not well-designed. The termination criteria based on approximation methods lead to significant error-correcting performance degradation and insufficient termination speed. 2) For IRA-LDPC decoder architectures which adopt Layered algorithm for multi-mode decoding, the so-called message updating conflict problem involved in Layered scheduling is not yet solved. Previous solutions such as S. Muller's work (DATE 2009) apply approximation method which degrades the convergence speed. 3) The partial-parallel Layered decoder architecture restricts decoding parallelism which makes it be not applicable for applications with high-throughput or ultra-low power consumption requirements.

This dissertation focuses on design methodology for solving the above problems in multi-mode LDPC decoders. 1) For generic Layered LDPC decoders, an ET scheme named as LSC-ET (last-iteration satisfaction check) scheme is proposed to reduce total decoding time while maintaining the error-correcting performance. 2) For DVB-T2 LDPC decoder, new message updating conflict resolution is proposed to prevent the conflict from occurring during variable-node message updating in Layered decoding. The degradation of error-correcting performance can be avoided. 3) To break the parallelism bottleneck of partial-parallel Layered decoder architecture, 3-state bit-serial fully-parallel Layered decoder architecture is proposed for WiMAX LDPC decoders to increase the decoding parallelism for higher energy-efficiency and enhanced error-correcting performance.

This dissertation consists of the following six chapters.

Chapter 1 [Introduction] introduces the scope and purpose of this dissertation. By a brief discussion on related works, problems and deficiencies of previous LDPC decoders are discussed. Finally, the organization of this dissertation is presented.

Chapter 2 [Background] introduces the fundamentals on LDPC codes, decoding algorithms and decoder architectures. Representations of LDPC codes, including the structured QC-LDPC and IRA-LDPC codes are presented. Evolution of decoding algorithms and their approximation for decoder implementation are also described. Finally, the previous partial-parallel Layered decoder architectures is discussed.

Chapter 3 [A BER Performance-Aware Early Termination Scheme for Generic Layered LDPC Decoder] presents a BER (bit error rate) performance-aware ET scheme for Layered LDPC decoder.

In Layered LDPC decoders, ET scheme without disturbing the decoding schedule is appreciated. Due to the frequently updated a-posteriori probability (APP) messages, hard-decisions (sign of variable-nodes) may be flipped even within each iteration, which means that sequential satisfaction of all check-nodes cannot deduce successful decoding. Previous ET schemes in Layered decoders, such as the HDA-ET scheme (Shao's, IEEE Comm. 1999) and CSC-ET scheme (Ueng's, ISCAS 2009), process the decoding and ET processes concurrently. The former one terminates decoding when the hard-decisions in two successive iterations are the same. The parity-check equations are not checked so that it cannot guarantee successful decoding. The latter one terminates decoding when the parity-checks are satisfied sequentially, which cannot deduce successful decoding.

The last-iteration satisfaction check based early termination (LSC-ET) scheme is proposed in this dissertation. In current iteration the parity-check process is done with the hard-decisions generated from the previous iteration. By applying this scheme, lossless BER performance and fast termination speed can be achieved with slight hardware overhead. Compared with CSC-ET scheme, the decoding time reduction is 5%~10% for WiMAX 1/2 rate codes and 2%~5% for DVB-S2 1/2 rate codes.

Chapter 4 [DVB-T2 LDPC Decoder with Perfect Conflict Resolution] presents a Layered LDPC decoder for DVB-T2 standard with a novel resolution to the message updating conflict problem.

The message updating conflict problem in Layered implementation becomes a big obstacle for multi-mode decoder design. In previous resolution such as in Muller's work (DATE 2009) the Layers with conflict sub-blocks are decoded in TPMP algorithm. In order to achieve this function, arithmetic units are increased by 50% in processing modules and also the convergence speed is slightly reduced compared to pure Layered decoding.

This dissertation focuses on the conflict resolution for DVB-T2 LDPC decoders against the error degradation. Unlike the previous resolutions, the Layered algorithm is directly applied without modification to the parity-check matrices (PCM) or the decoding algorithm. DVB-T2 LDPC decoder architecture is also proposed with two new techniques which can guarantee conflict-free Layered decoding. The PCM Rearrange technique largely reduces the number of conflicts and eliminates all of data dependency problems between Layers to ensure high pipeline efficiency. The Layer Division technique divides the remaining conflict Layers into two sub-layers and processes them concurrently with a well-designed, overlapped pipeline decoding schedule, so that at most 1.2% of decoding time redundancy is required. As a result, the proposed decoder architecture can reduce the extra arithmetic units involved in Muller's solution and reduce around 1/3 error bits compared to Muller's method.

Chapter 5 [High-Parallel Performance-Aware LDPC Decoder for WiMAX] presents a synthesizable LDPC decoder IP for WiMAX system with high parallelism and enhanced error-correcting performance.

For high energy-efficiency LDPC decoder design, the proper parallelism is the most important factor. When the parallel level exceeds the maximum Layer size of PCM, energy-efficiency will decrease because of data dependency problem. Although the partial-parallel Layered architecture proposed by Sun can support multi-mode decoding with more than 1Gbps throughput for WiMAX, the limited decoding parallelism restricts decoders from further improved energy-efficiency. Xiang's dual-path partial-parallel Layered decoder architecture (JSSC 2011) focuses on the dependency problem and achieves almost twice parallelism. Peng's bit-serial Layered decoder (A-SSCC 2011) processes each Layer in fully-parallel way to further increase parallelism. However, the saturation process for bit-serial variable-node message updating requires doubled interconnection circuits. Also, the serial-parallel conversion requires huge decoder area since the decoder functionality cannot be composed of bit-serial components only.

The 3-state fully-parallel Layered decoder is proposed in this dissertation. Two bit-width serial processing is applied instead of 1 bit-width design, which achieves doubled parallelism compared with Peng's work. A new message saturation strategy is proposed to reduce 50% of the interconnection circuits. The improvement in serial-parallel conversion circuits makes the total area increase only 1%. Power synthesis result shows that the proposed decoder achieves 5.83pJ/bit/iteration energy-efficiency which is 46.8% and 72.1% improvement compared to Peng's work and Xiang's work, respectively. Furthermore, an advanced dynamic quantization (ADQ) technique is proposed and implemented in the decoder to enhance BER performance. 6-bit ADQ achieves BER performance close to 7-bit fixed quantization with improved error floor performance.

Chapter 6 [Conclusion] concludes the contribution of this dissertation and discusses about the future work.

Acknowledgement

First of all, I would like to thank my supervisor, Professor Satoshi Goto, for his guidance in my research work, encouragement in my daily life and sufficient financial support. His great foresight and sagacity guide me throughout my research work. His hard working and understanding personality encourage me to follow him all the time. Without his support, I can never get the confidence to defeat every difficulty aboard.

I would like to thank Professor Takahiro Watanabe, Professor Shinji Kimura and Associate Professor Fumiaki Maehara for my research in Waseda University. Thanks for their advices and comments to my dissertation, which is professional and helpful.

Especially, I show my great thanks to Dr. Xiao Peng, Dr. Zhixiang Chen and Dr. Dajiang Zhou for their valuable works in my research and unstopped care in my daily life. I would like to thank Dr. Wen Ji, Dr. Xianghui Wei, Mr. Yichao Lu, Mr. Qian Xie, Ms. Ying Cui, Mr. Nanfan Qiu and Ms. Xin Xiao for working together with me on the LDPC field. I also want to thank all the members of Goto's laboratory together with me during these five years. Thanks to you, I have enjoyed a happy research life in Japan without any homesick and loneliness.

I also extend my heartfelt thanks to my family and my wife. I cannot even live without their support and care in every aspect.

Finally, I should thank the Japanese government scholarship given by MEXT throughout my three-year-long Ph.D. course for my research and life in Japan.

Contents

Index of Figures

Index of Tables

1. Introduction

In recent years, the demand of efficient, reliable transmission and storage systems for digital data increases rapidly. The emergence of large-scale, high-speed data networks for exchanging, processing and storing the digital data in commercial, military and many other areas further accelerate this trend. The major concern of the system designer is the control of transmission errors to make it more reliable. Such technology is called error control coding (ECC) [1].

Error control strategies for one-way communication systems must use the forward error correction (FEC), in which they employ error-correcting codes (ECC) that automatically correct errors detected at the receiver. Although two-way communication systems can use error detection and retransmission strategy, called automatic repeat request (ARQ), FEC is commonly combined with ARQ to maintain transmission throughput.

Low-Density Parity-Check (LDPC) codes [2], as linear block ECC, which were invented by Gallager in 1963 but mostly ignored by coding researchers for more than 30 years until rediscovered by MacKay in 1999 [3], nowadays become a hot topic in a variety of application fields. Due to inherently paralleled decoding algorithm and near Shannon limit error-correcting performance, LDPC codes have been widely adopted in modern communication standards, such as Wi-Fi (IEEE 802.11n) [4], WiMAX (IEEE 802.16e) [5], WPAN (IEEE 802.15.3c) [6] and the second-generation of digital video broadcasting (DVB-x2) family of standards [7]. Moreover, next generation storage systems and optical transmission systems also adopt them for improving the reliability of devices or communication distances. The rediscovery of LDPC codes makes them strong competitors with turbo codes [8] and BCH codes [9] which occupy FEC field for a long period in a variety of applications such as CD, DVD, SSD, wireless communication standards and optical fiber transmission.

1.1 Scope of This Research

A typical transmission system with FEC using LDPC codes can be represented by the block diagram as shown in Fig.1. During LDPC encoding process, redundant information should be added to the encoded source data against the noise of transmission channel. LDPC Decoder attempts to correct the error occurs in the transmission to recover the original source data. Compared to the LDPC encoder and modulator/demodulator, the LDPC decoder is much more complicated and critical module, since the throughput requirement and error-correcting capability becomes more and more critical in recent applications. In this dissertation, contributions are focused on design techniques of LDPC decoders.

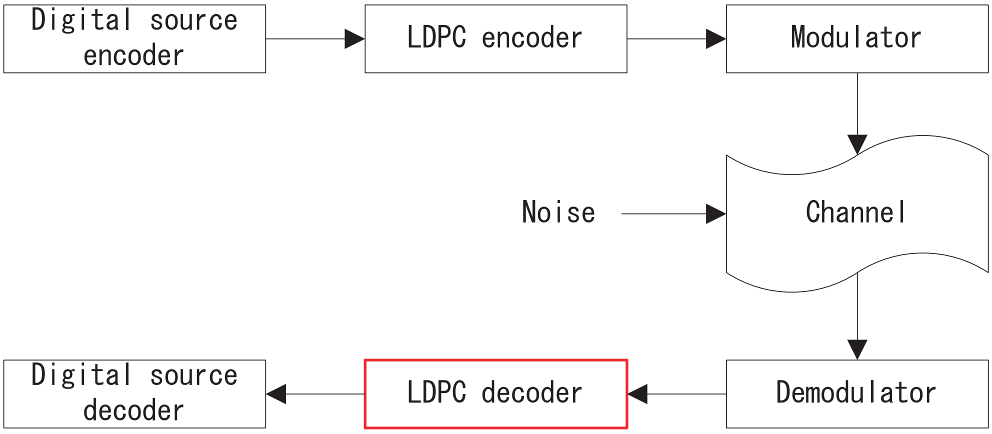


Fig.1. Transmission system with FEC using LDPC codes

Error-correcting capability is the first factor should be taken into consideration for LDPC decoders. Especially in the one-way transmission systems, such as DVB-T2 standard [10], DVB-S2 standard [11], optical fiber transmission system and many kinds of storage media. Even in the FEC-ARQ based systems such as WiMAX system, higher error-correcting capability can effectively enlarge the transmission throughput. The energy-efficiency of decoders, i.e. electrical energy dissipation for per decoded bit, is another important indicator for LDPC decoders. For mobile devices, such as portable TV, smart phone and tablet PC, the communication modules still dissipate a large proportion of power supply.

On the other hand, decoders which can only decode one kind of LDPC code is inconvenient in many applications. Apart from the changeable channel conditions of wireless communications, even for wired communication systems such as Ethernet transmission, the signal attenuation could be changed a lot due to different Ethernet cable lengths. Capability of multi-mode (multi-length and multi-rate) decoding in one single LDPC decoder is appreciated and becomes universal in both industrial products and academic researches.

In this dissertation, we emphasis on both the error-correcting performance and energy-efficiency of multi-mode LDPC decoders. For generic multi-mode LDPC decoders which are decoded with layered algorithm, (i.e. multi-mode oriented architectures) we have proposed an new early termination scheme to reduce power consumption while it can maintain the error-correcting performance. [12] The proposed ET scheme terminates the decoding process immediately when it detects a successful decoding during iterative layered LDPC decoding schedule. Furthermore, the proposed ET scheme already combines the parity-check process so that signals can be generated exactly to indicate successful decoding.

We also design two LDPC decoders for two totally different applications as follows.

1) For DVB-T2 standard which adopts structured irregular repeat-accumulate (IRA) LDPC codes, we have designed full-mode partial-parallel LDPC decoder architecture to enhance error-correcting performance by solving the inherent message updating conflict problem when the decoder applies the layered decoding algorithm. It is the first time DVB-T2 LDPC decoder can apply pure layered scheduling. [13]
2) For WiMAX standard which adopts structured quasi-cyclic (QC) LDPC codes, we have designed full-mode, fully-parallel LDPC decoder architecture to increase the decoding parallelism. Power synthesis result shows that the proposed decoder achieves 5.83pJ/bit/iteration energy-efficiency at over 1Gbps throughput, which is 46.8% improvement compared to state-of-the-art WiMAX LDPC decoder [14]. Furthermore, a newly proposed advanced dynamic quantization (ADQ) scheme

for layered algorithm is combined in the decoder to enhance error-correcting performance so that 6-bit ADQ quantization can achieve performance close to 7-bit fixed quantization with largely improved error floor performance.

1.2 Related Works on LDPC Decoders

Previously published works which relate to the contributions of this dissertation are briefly discussed in this section. Especially, the problems and techniques which need to be improved of the previous works are described in detail.

1.2.1 Early Termination Schemes

Early termination (ET) scheme is a hot topic not limited to LDPC decoder design since it can tremendously increase the decoding throughput or reduce the power consumption. Many works have been done for both traditional TPMP decoder architecture and layered decoder architecture. [15]-[19] ET schemes are usually focused on two different aspects: one is to terminate the decoding process when a codeword is not able to be successfully decoded before reaching the maximum iteration time; [17][18] another one is to terminate the decoding process when a codeword is successfully decoded (or almost finishes decoding) before reaching the maximum iteration time. [15][16][19]

Early termination scheme in layered decoder architecture is different from that in TPMP architecture since the a-posteriori probabilities (APP) are updated more than once in a single iteration of layered decoding process which are only updated once in each iteration of TPMP. For ET schemes in layered decoding schedule, previous works have been done in [15],[16].

In [15], the popular hard-decision aided (HDA) ET scheme, which is traditionally applied in turbo decoding algorithm, stores all the hard-decisions and compares the decoded codeword in two successive iterations. If the decoded codeword of these two iterations are exactly the same, the decoding will be terminated. Otherwise, the decoding process will be terminated when a predefined maximum iteration time is reached. However, this ET scheme does not verify the parity-check equations so that it cannot guarantee lossless BER performance. Sometimes an additional constraint is added to increase the reliability, in which all the absolute values of APP should be larger than a predefined threshold. The threshold value for HDA varies from different channel conditions thus it needs to be determined through simulations for better BER performance. The extra condition reduces termination speed but still cannot ensure lossless BER performance. In [16], the current-iteration satisfaction check based early termination (CSC-ET) which examines the parity-check equations layer by layer during the decoding process. When all the parity-check equations are satisfied sequentially in a single iteration, the decoding process will be terminated immediately or after additional iterations for better BER performance. Both of them can terminate the decoding to reduce average iteration numbers without disturbing the layered pipelined schedule.

However, both of them bring BER performance degradation, which worsens the decoding performance and is not suitable to be implemented in BER performance-aware decoders, such as DVB-S2 standard. Moreover, both of schemes just terminate the decoding process without checking the parity-check equations. For two-way transmission applications which adopt FEC-ARQ error control strategy such as WiMAX, parity-check equations must be checked to generate successful decoding signals. In such case, decoders must be combined with extra parity-check logics.

1.2.2 DVB-x2 LDPC Decoders

The DVB-x2 family of standards, which contains DVB-S2 (satellite transmission system), DVB-T2 (terrestrial transmission system) and DVB-C2 (cable transmission system), adopt code-length as long as 64k bits IRA-LDPC codes as their inner ECC concatenated with outer BCH codes to achieve quasi-error-free (QEF) reception conditions within 0.8dB from the Shannon limit.

11 kinds of code-rates and 2 kinds of code-lengths of LDPC codes have been

adopted in DVB-S2 standard for various transmission channel conditions and applications, while DVB-T2 and DVB-C2 standards pick up 6 and 4 kinds of code-rates out of 11 for their own application fields, respectively. These specialized applications require decoders to support multi-mode decoding. For such extremely long structured IRA-LDPC codes, many researchers attempt to implement multi-mode LDPC decoders with layered algorithm for efficient partial-parallel decoding in similar ways as for multi-mode QC-LDPC decoders. [20]-[24]

To apply the layered algorithm for decoders, PCM should be divided into layers and in each layer the check-node processing should be processed in parallel way. However, the IRA-LDPC codes usually cannot be implemented with layered algorithm as straightforward as QC-LDPC codes because of the so-called message updating conflict problem which appears while divides the PCM into layers. Simply ignoring the conflicts will cause a lot of cutting-edge problems during layered decoding, which leads to significant performance loss and relatively high error floor. The BER performance degradation can be larger than 0.2dB compared with conflict-free layered decoding performance. [24] In order to solve this problem, authors in [20] and [21] decoded the conflict layers with TPMP algorithm while authors in [22] and [23] tried to modify the PCM of conflict layers. The author in [24] proposed a selective recalculation strategy adding to the layered decoding algorithm to achieve conflict-free error-correcting performance, but the proposed algorithm is quite not hardware-friendly due to its instability of complexity.

1.2.3 Multi-Mode QC-LDPC Decoders

The structured QC-LDPC codes have been adopted in many wireless communication standards, not limited to WiMAX, Wi-Fi and WPAN standards. For various transmission channel conditions and applications, multi-mode decoding capability is required in such standards.

For multi-mode QC-LDPC decoders, previous works have been done in [25]-[29]. Some of the decoder works can only support multi-lengths decoding such as [28]. The authors in [25] present a partial-parallel layered LDPC decoder for QC-LDPC codes, which can support variable code-length and code-rate decoding. The presented partial-parallel layered architecture can deal with different PCMs (i.e. different code-rates) by changing the barrel shifter factors of the so-called permutation network. However, Such kind of block by block layered decoding schedule suffers from the data dependency problem in layered decoding which limits the decoding parallelism and throughput. By using this architecture, WiMAX LDPC codes need at least 76~88 clock cycles for decoding each iteration. Authors in [26] present a dual-path partial-parallel layered LDPC decoder for WiMAX in order to increase decoding parallelism with two sets of processing units. By utilizing the sub-matrix reordering and complex bypass controlling techniques to reduce the influence from data dependency problem, they finally reduce the number of clock cycles to 48~54 per iteration. In order to further increase decoding parallelism for better energy-efficiency, we propose a bit-serial layered LDPC decoder for WiMAX in [29]. In this work, the data dependency between adjacent layered does not exist and the block by block decoding schedule is changed to fully-parallel schedule. Bit-serial message processing style is applied to avoid complicated processing units and interconnection. As a result, the number of clock cycles per iteration is reduced to 24~48 while the area of decoder is as large as that of [26].

1.3 Organization

This dissertation contains the following six chapters.

Chapter 1 introduces the scope and purpose of this dissertation. By a brief discussion on related works, problems and deficiencies of previous LDPC decoders are discussed. Finally, the organization of this dissertation is presented.

Chapter 2 introduces the fundamentals on LDPC codes, decoding algorithms and decoder architectures. Representations of LDPC codes, including the structured QC- and IRA-LDPC codes are presented. Evolution of decoding algorithms and their approximation for decoder implementation are also described. Finally, commonly adopted partial-parallel decoder architectures are demonstrated.

Chapter 3 presents a BER performance-aware ET scheme for layered LDPC decoder. Unlike the ET schemes in traditional TPMP decoders, in layered LDPC decoders ET scheme without disturbing the decoding schedule is not easily designed. Due to the frequently updated a-posteriori probability (APP) messages, hard-decisions may be flipped even within each iteration, which means sequential satisfaction of all check-nodes cannot deduce successful decoding. Previous ET schemes in layered decoders, such as the HDA-ET scheme (Shao’s, IEEE Trans. On Comm. 1999) and CSC-ET scheme (Ueng’s, ISCAS 2009), suffer from both significant performance loss and slow termination speed. The most important is that the above schemes do not check the parity-check equations so that extra parity-check logics need to be implemented in the decoder.

The last-iteration satisfaction check based early termination (LSC-ET) scheme is proposed in this dissertation. In current iteration the parity-check process is done with the hard-decisions generated from the previous iteration. By applying this scheme, lossless BER performance and fast termination speed can be achieved with acceptable hardware overhead. Moreover, the proposed ET scheme is not limited to any LDPC codes or check-node message-updating functions.

Chapter 4 presents a layered LDPC decoder for DVB-T2 standard with a novel resolution to the message updating conflict problem. Although the structured IRA-LDPC codes specified in DVB-x2 family achieve excellent coding gain, the message updating conflict problem in layered implementation becomes a big obstacle for multi-mode decoder design. In previous resolution such as in Muller’s work (DATE 2009) the layers with conflict sub-blocks are decoded in TPMP algorithm. In order to achieve this function, arithmetic units are increased by 50% in processing modules and also the convergence speed is slightly reduced compared to pure layered decoding. Simulation results show that compared to pure layered decoding performance this strategy leads to 1.5 times of error bits in the same condition in the worst case.

In this dissertation we focus on the conflict resolution for DVB-T2 LDPC decoders against the error degradation. Unlike the previous resolutions, we can directly apply the layered algorithm without modifying parity-check matrices (PCM) or the decoding algorithm. DVB-T2 LDPC decoder architecture is also proposed with two new techniques which can guarantee conflict-free layered decoding. The PCM Rearrange technique largely 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 the remaining conflicts with a well-designed, overlapped pipeline decoding schedule. At most 1.2% of decoding time redundancy is required. As a result, the proposed decoder architecture can reduce the maximum iteration number required in DVB-T2 decoder and the extra arithmetic units involved in Muller's solution can be avoided.

Chapter 5 presents a synthesizable LDPC decoder IP for WiMAX system with high parallelism and enhanced error-correcting performance. Although the partial-parallel layered architecture can support multi-mode decoding with more than 1Gbps throughput for WiMAX, the limited decoding parallelism restricts decoders from better energy-efficiency. By taking the advantages of both layered scheduling and fully-parallel architecture, the 3-state fully-parallel decoder is proposed in this dissertation, which can fully support multi-mode decoding specified in WiMAX with the parallelism more than 4 times higher than Xiang's dual-path partial-parallel layered decoder architecture (JSSC 2011) and twice than our previous bit-serial decoder (Peng's, A-SSCC 2011).

In the proposed decoder, 6-bit quantized messages are split into bit-serial style and 2 bit-width serial processing lines work concurrently so that only 3 clock cycles are required to decode one layer. By reducing the redundant MUX logics and check message memory of our previous work (Peng's, A-SSCC 2011), it doubles the parallelism and solves the message saturation problem of bit-serial arithmetic with only 1% area increase. Power synthesis result shows that the proposed decoder achieves 5.83pJ/bit/iteration energy-efficiency which is 46.8% and 72.1% improvement compared to Peng's work and Xiang's work, respectively. Furthermore, an advanced dynamic quantization (ADQ) technique is proposed and implemented in the decoder to enhance BER performance in layered decoder architecture. With about 2% area overhead, 6-bit ADQ can achieve BER performance close to 7-bit fixed quantization with improved error floor performance.

Chapter 6 concludes the contribution of this dissertation and discusses about my future work.

2. Background

This chapter introduces the fundamentals on LDPC codes, decoding algorithms and decoder architectures. Representations of LDPC codes, including the structured QC- and IRA-LDPC codes are presented. Evolution of decoding algorithms and their approximation for decoder implementation are also described. Finally, commonly adopted partial-parallel decoder architectures are demonstrated.

2.1 LDPC Codes and Their Representations

An LDPC code is a linear block code for which the parity-check matrix (PCM) has a low density of 1's. Like all kinds of linear block codes, it can be defined via a sparse parity-check matrix H of size $M \times N$.

Regular (n, k) LDPC codes are one kind of LDPC codes whose parity-check matrix H contains exactly Wc ones per column and $Wr = Wc\,(n/m)$ ones per row, where $Wc << m$. The number of ones in common between any two rows is no greater than one.

If the number of ones per column or per row is not constant, the code are called irregular LDPC codes. It has been shown that long random irregular LDPC codes perform arbitrarily close to the Shannon limit.

Gallager did not provide a good code constructing method algebraically and systematically. Good LDPC codes that have been found are largely computer generated, and their encoding is very complex. But later many researchers introduced their ways to construct LDPC codes based on finite geometries. Some long finite-geometry LDPC codes have error-correcting performance only a few tenths of a decibel away from the Shannon limit.

2.1.1 PCM Representation

PCM representation is a common method to represent an LDPC code. The PCM H for a given code can be derived from its generator matrix G (and vice-versa).

When one defines a code, he usually uses PCM or generator matrix to show it while PCM is convenient for decoding and generator matrix for encoding. Of course, both regular and irregular LDPC codes can be represented by PCMs.

A $(10, 5)$ linear block code which is the null space of the following PCM is shown in Fig.2. It is easy to check that this code is a $(4, 2)$ regular LDPC code.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Fig.2. Parity-check matrix of a (10, 5) linear block code

2.1.2 Tanner Graph Representation

A useful graphical representation of linear block codes is the representation by a factor graph or Tanner graph [30], which displays the incidence relationship between the codeword bits and the parity-check sums that check on them. There are two classes of nodes in a Tanner graph, the variable-nodes and check-nodes. The Tanner graph of a code is drawn according to the following rule:

Check-node j is connected to variable-node i whenever element h_{ji} in H is a ‘1’.

Fig.3 shows the corresponding Tanner graph to that its PCM is shown in Fig.2. When there is a ‘1’ in the PCM, there is an edge between the corresponding variable-node and check-node.

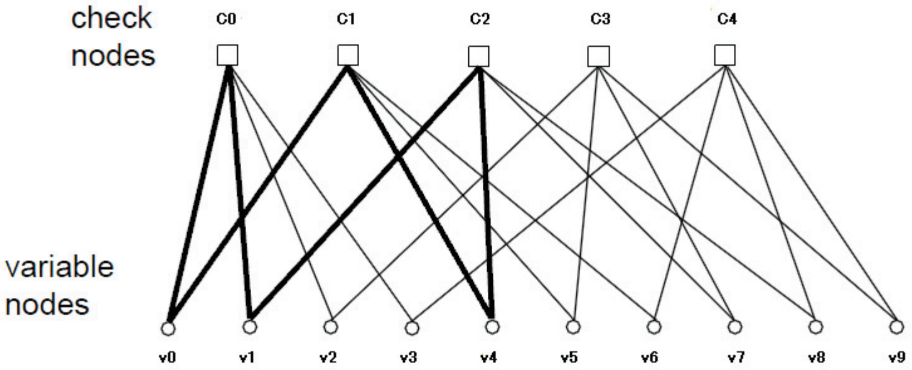


Fig.3. Tanner graph of a (10, 5) linear block code

For a regular LDPC code, the degrees of all the variable-nodes in the Tanner graph are the same and equal to the column weight of the parity-check matrix; and the degrees of all the check-nodes are the same and equal to the row weight of the parity-check matrix. Such a Tanner graph is said to be regular. Furthermore, there are no two variable-nodes of an LDPC code are both checked by two different check-nodes. This implies that the Tanner graph of an LDPC code does not contains cycles of length 4. Short cycles will degrade the iterative decoding performance and even prevent the decoding from convergence.

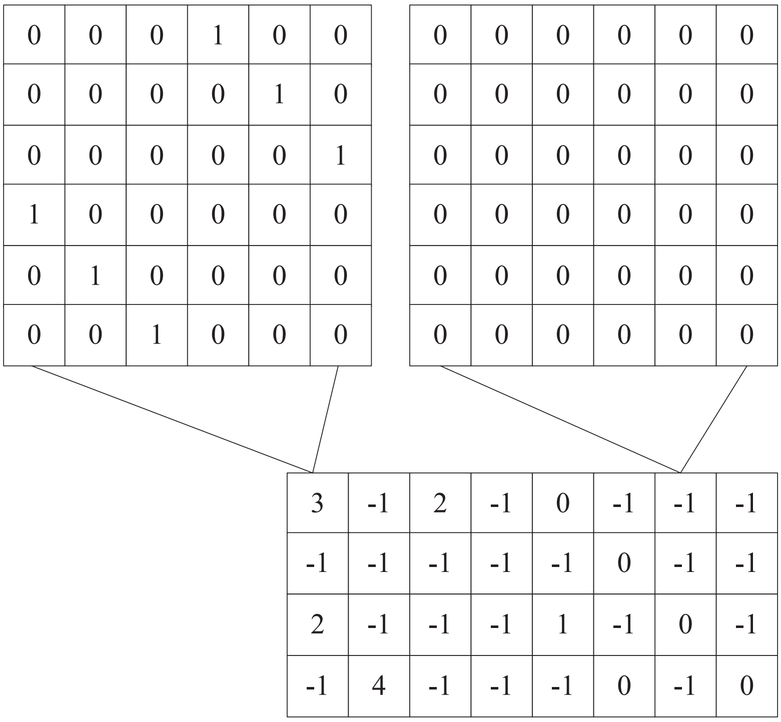
2.2 Structured LDPC Codes

The performance of LDPC codes largely depends on their code structures. A large number of LDPC codes are proposed to achieve near capacity performance, efficiency in encoding and decoding processes, reduced memory requirement, and so on. In this chapter, we put emphasis on two kinds of structured LDPC codes which are the quasi-cyclic (QC) LDPC codes and irregular repeat accumulate (IRA) LDPC codes. For parallel decoder implementation, structured LDPC codes such as IRA-LDPC codes and QC-LDPC codes have been adopted in a variety of wireless/wired communication standards.

2.2.1 QC-LDPC Codes

Originally, a quasi-cyclic code is defined as a linear code for which cyclically shifting a codeword a fixed number $n_0 \neq 1$ of symbol position either to the right or to the left results in another codeword. For $n_0 = 1$, a quasi-cyclic code is a cyclic code. [31]

While in real applications, the definition of QC-LDPC structure is a little bit different from the original QC code definition. LDPC codes such as in WiMAX are defined as QC-LDPC codes. QC-LDPC codes are characterized by the parity-check matrix that consists of small square blocks, which is also called sub-blocks, that are zero matrices or shifted identities. A simple example is shown in Fig.4.



0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1
1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

3	-1	2	-1	0	-1	-1	-1
-1	-1	-1	-1	-1	0	-1	-1
2	-1	-1	-1	1	-1	0	-1
-1	4	-1	-1	-1	0	-1	0

Fig.4. PCM of a QC-LDPC code

Fig.4 shows the 48×24 parity-check matrix of a QC-LDPC code. The 8×4 matrix with numbers is named as the seed matrix. Each position in the seed matrix specifies a zero matrix or a cyclically shifted identity. In this example each block is a 6×6 sub-matrix. '-1' in the seed matrix means a zero matrix as shown above and the non-negative numbers in blocks are the shift values for shifted identities. Due to the cyclic property of QC-LDPC codes, they are effectively suitable for parallel implementation. We can easily combine each row block or column block in the PCM as a partition to process partial-parallel decoding.

The WiMAX LDPC codes belong to such kind of QC-LDPC codes. There are six different classes of codes, which contain four different code rates R from 1/2 to 5/6 [5]. All of the codes have the similar code structure and can be encoded and decoded efficiently. They consist of a fix size of twenty-four columns in their seed PCM, in which the block size is z by z (z is the expansion factor). The first R*24 columns in the seed PCM correspond to the information part and the remained columns correspond to the parity part. The number of rows in the seed PCM depends on the code rate R and equals to the parity part. The parity part has a fixed structure according to the encoding scheme.

The size of z is ranged from 24 to 96 with a step size equal to 4. There are totally 19 kinds of code length, which range from 576 to 2304 bits.

For the variable sizes of sub-matrices, the WiMAX standard only gives six seed matrices for the longest code-length. For other sizes of sub-matrices, they propose equations (2.1) and (2.2) to scale the shift value $\mathrm{p(f,i,j)}$ accordingly.

$$p(f,i,j)=\begin{cases} p(i,j), p(i,j)\le 0 \\ \left\lfloor \dfrac{p(i,j)z_f}{z_0} \right\rfloor, p(i,j)>0 \end{cases} \tag{2.1}$$

$$p(f,i,j)=\begin{cases} p(i,j), p(i,j)\le 0 \\ \mathrm{mod}(p(i,j),z_f), p(i,j)>0 \end{cases} \tag{2.2}$$

Equation (2.1) is for rate 1/2, 3/4A, 3/4B, 2/3B and 5/6 codes while equation (2.2) is only for rate 2/3A codes.

2.2.2 IRA-LDPC Codes

Irregular repeat accumulate (IRA) codes are a generalization of the repeat accumulate (RA) codes. [32] IRA codes can be encoded and decoded in linear time.

Compared to the standard LDPC codes, IRA codes have reduced encoding time [41]. IRA codes can be divided into two types: the non-systematic and systematic ones. In this dissertation, we only focus on the systematic IRA codes.

The Tanner graph for IRA codes is shown in Fig.5.

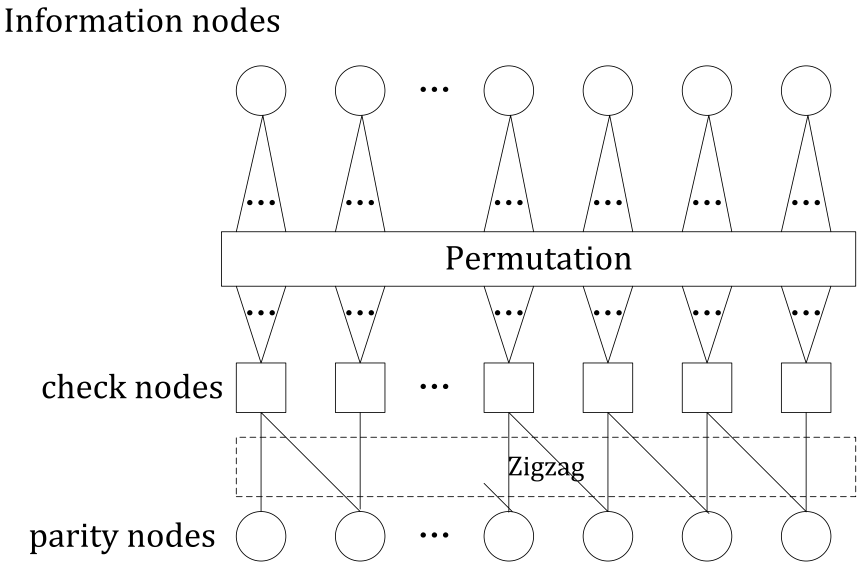


Fig.5. Tanner graph for IRA codes

The variable nodes can be divided into information nodes and parity nodes. The variable nodes connect to the check nodes randomly, and the parity nodes connect to the check nodes in a zigzag pattern as shown in Fig.5. The zigzag pattern enables the encoding process of IRA codes to be realized with accumulators.

The parity-check matrix of the systematic IRA code consists of two parts. One part is deterministic, and the other part is generated randomly. Therefore the PCM H can be described by equation (2.3) as follows.

$$H_{M\times N} = \left[\begin{array}{c|ccccc} & 1 & 0 & 0 & \cdots & 0 \\ & 1 & 1 & 0 & & 0 \\ A_{M\times K} & 0 & 1 & 1 & & 0 \\ & \vdots & & & \ddots & \vdots \\ & 0 & \cdots & 0 & 1 & 1 \end{array}\right] \tag{2.3}$$

We denote the codeword by $c = (i_1, ..., i_K, p_1, ..., p_M)$ where *i* are information bits and *p* are parity bits. Then, using this PCM, the encoding can be carried out by

$$p_1 = \sum_{k=1}^{K} i_k h_{1,k} \tag{2.4}$$

$$p_m = p_{m-1} + \sum_{k=1}^{K} i_k h_{m,k}, 2 \le m \le M \tag{2.5}$$

The most important property is that the parity bits are determined from the information bits and the random part of PCM without any computation of the generator matrix. So that the encoding process can be simplified.

LDPC codes specified in DVB-x2 standards are systematic IRA codes. The code-lengths are 64800 bits for normal frame and 16200 bits for short frame. For different channel conditions and applications, there are totally 11 kinds of code-rates defined in DVB-x2 standards, ranged from 1/4 to 9/10. More detail about the features of LDPC codes in this family of standards will be discussed in chapter 4.

2.3 LDPC Decoding Algorithms

LDPC codes can be decoded in various ways. Among them the Sum-product algorithm (SP), also called Message-passing algorithm or Belief propagation algorithm (BP), provides the best error-correcting performance and yet is practically implementable. In this section, firstly the basic SP decoding algorithm and some approximations for simplifying the decoder implementation are discussed. Then we pay attention to the layered decoding schedule and explain the reason why most of multi-mode decoders adopted such algorithm.

2.3.1 Sum-Product Algorithm

LDPC codes can be decoded iteratively with the SP algorithm as described in [1]. In this algorithm, firstly we define the column index sets $\mathrm{A(m)}$ and the row index sets $\mathrm{B(n)}$ as follows:

$$A(m) = \{n | H_{mn} = 1\}$$
$$B(n) = \{m | H_{mn} = 1\}$$

For the Additive white Gaussian noise (AWGN) channel with noise variance σ^2 and received signal y_i, the conditional probability of being $\mathrm{x}_i = 0$ or $\mathrm{x}_i = 1$ is represented as follows:

$$P(y_i \mid x_i = 0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y_i - 1)^2}{2\sigma^2})$$
$$P(y_i \mid x_i = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y_i + 1)^2}{2\sigma^2})$$

The input to the SP algorithm is the initial message of λ_n. As a log likelihood ratio, the initial message λ_n for the AWGN channel can be represented as follows:

$$\lambda_n = \ln \frac{P(y_n \mid x_n = 0)}{P(y_n \mid x_n = 1)} = \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y_n - 1)^2}{2\sigma^2})}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y_n + 1)^2}{2\sigma^2})}$$

$$= -\frac{(y_n - 1)^2}{2\sigma^2} + \frac{(y_n + 1)^2}{2\sigma^2} = \frac{2y_n}{\sigma^2} \tag{2.6}$$

Following steps shows the detailed process of the logarithm domain SP decoding:

[Initialization]

Calculate the logarithm likelihood ratio (LLR) λ_n for $\mathrm{n} = 1, 2, \ldots, \mathrm{N} - 1$ using equation (2.6), where σ^2 is the variance of noise generated by the AWGN Channel. Set $\beta_{mn} = \lambda_n$ for each (m, n) satisfying $\mathrm{H}_{mn} = 1$. Set the loop counter $l = 1$ and the maximum number of iterations is set to l_{max}.

Step 1 [Check-Node Update]

For all the check-nodes C_m in the order from $m = 1, 2, \dots, M$, compute the intermediate messages α_{mn} according to equation (2.7), where each set (m, n) satisfies $H_{mn} = 1$.

$$\alpha_{mn} = \left(\prod_{n' \in A(m) \backslash n} sign(\beta_{mn'}) \right) \times f\left(\sum_{n' \in A(m) \backslash n} f(|\beta_{mn'}|) \right) \tag{2.7}$$

Where function f(x), which is also called Gallager function, is defined as follows:

$$f(x) = \ln \frac{\exp(x) + 1}{\exp(x) - 1}$$

Step 2 [Variable Node Update]

For all the variable-nodes V_n, in the order from $n = 1, 2, \dots, N$, compute the message β_{mn} with equation (2.8), for each set of (m, n) satisfying $H_{mn} = 1$.

$$\beta_{mn} = \lambda_n + \sum_{m' \in B(n) \backslash m} \alpha_{m'n} \tag{2.8}$$

Step 3 [Hard-Decision]

Calculate all the tentative hard-decision bits $\hat{y}_n$ for $n = 1, 2, \dots, N$.

$$\hat{y}_n = \begin{cases} 0, sign(\lambda_n + \sum_{m' \in B(n)} \alpha_{m'n}) = +1 \\ 1, sign(\lambda_n + \sum_{m' \in B(n)} \alpha_{m'n}) = -1 \end{cases}$$

Step 4 [Parity-Check]

If the tentative hard-decision vector $(\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_N)$ satisfies the parity-check equation (2.9), output the codeword, and terminate the decoding iteration. If not, go to step 1.

$$H \cdot (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_N)^T = 0 \tag{2.9}$$

Step 5 [Maximum Iteration Check]

If $l \le l_{\max}$, set the loop counter $l \leftarrow l + 1$ and go to Step 1. Otherwise output the decoded codeword $(\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_N)$ and stop decoding.

2.3.2 Algorithm Approximation for Hardware Implementation

The original SP algorithm is not hardware-friendly due to the non-linear Gallager function which involves large numbers of exponential, multiplicative, and look-up table (LUT) operations in the check-node (CN) message updating functions. The Min-sum (MS) approximation [33] is invented to achieve lower computation complexity.

Consider the curve of Gallager function f(x) as shown in Fig.6.

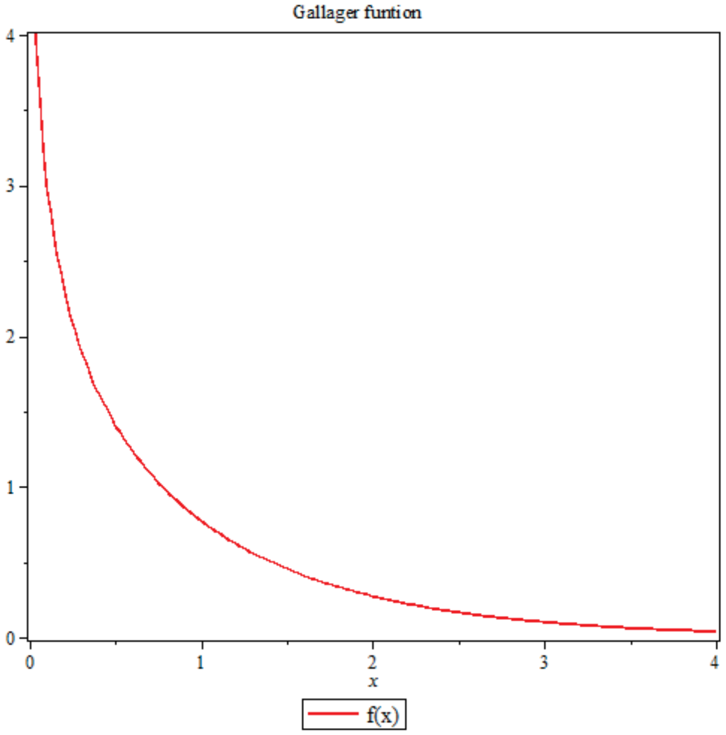


Fig.6. Gallager function curve

We may conclude that the term of Gallager function more depending on the smallest $\beta_{mn'}$ value in $f\left(\sum_{n'\in A(m)\backslash n} f(|\beta_{mn'}|)\right)$ so that we can get the rough approximation as follows:

$$f\left(\sum_{n'\in A(m)\backslash n} f(|\beta_{mn'}|)\right) \approx \min_{n'\in A(m)\backslash n} |\beta_{mn'}|. \tag{2.10}$$

Equation (2.10) shows that the MS algorithm can make the CN message updating function performed only by addition and comparison operations, which is comparably suitable for hardware implementation.

Since the original MS algorithm degrades the error-correcting performance significantly, in real implementation, the modified types MS algorithms are introduced. The most popular and simple ones are the so-called Normalized min-sum (NMS) algorithm [34] and Offset min-sum (OMS) algorithm [35]. The two variations of the MS algorithm usually adopted in recent LDPC decoder implementations to make trade-off between error-correcting performance and implementation complexity. By delicate selecting the corresponding parameters, both can achieve approaching error-correcting performance as SP algorithm.

The NMS approximation algorithm is that on the basis of MS algorithm, CN messages multiply a normalize factor κ as defined in equation (2.11), which can make the calculation of CN messages α_{mn} much accuracy. Parameter κ is usually set to 0.625 $(= 5/8)$, 0.75 $(= 3/4)$ or 0.875 $(= 7/8)$ for hardware implementation consideration.

$$\alpha_{mn} = \left(\prod_{n' \in A(m) \backslash n} sign(\beta_{mn'}) \right) \times \kappa \min_{n' \in A(m) \backslash n} |\beta_{mn'}| \tag{2.11}$$

The OMS approximation algorithm is that on the basis of MS algorithm, CN messages make an offset subtraction of the minimum value as defined in equation (2.12). In order not to flip the sign of CN messages, additional condition is added to the subtraction operation. Experimental parameter β is usually set to 0.125 $(= 1/8)$ for implementation.

$$\alpha_{mn} = \left(\prod_{n' \in A(m) \backslash n} sign(\beta_{mn'}) \right) \times \max(\min_{n' \in A(m) \backslash n} |\beta_{mn'}| - \beta, 0) \tag{2.12}$$

Notice that for different LDPC codes and signal to noise ratio (SNR) status, the above approximated algorithms may quite performance different results. In real applications, the selection should be based on carefully and systematic simulations.

2.3.3 Layered Decoding Schedule

The Layered decoding algorithm, which is also called Turbo-decoding Message-passing (TDMP) algorithm, is invented to make trade-off between error-correcting performance and hardware complexity. The CN function is the same as SP algorithm, but the message updating schedule is totally different. The SP algorithm or TPMP schedule uses one phase to process CN message updating and another for VN message updating. Unlike the TPMP schedule, the layered schedule divides check-nodes into several groups. It seems like that one iteration is divided into several sub-iterations. Inside each sub-iteration, the decoding schedule is the same as TPMP. It means that during a single iteration the APP messages can be updated several times. For each CN (or group of CN), the APP messages are updated after the CN message updating immediately in layered algorithm. The updated APP messages participate in the CN updating of the following CN (or group of CN) within the iteration, which is different from TPMP who cannot use such updated information until the next iteration. It is already proven that the more frequent APP message updating accelerates the convergent speed by at most twice in [36]. Fig.7 and Fig.8 show the difference between layered algorithm and TPMP on message updating schedules. It can be seen clearly that in layered schedule for each sub-iteration the connection become simpler than TPMP algorithm, so that the layered decoding schedule can be considered as one possible solution for partial-parallel implementation.

Nevertheless, the data dependency problem will become serious in layered algorithm due to frequent APP message updating, which will limit the flexibility and parallelism of its hardware implementations.

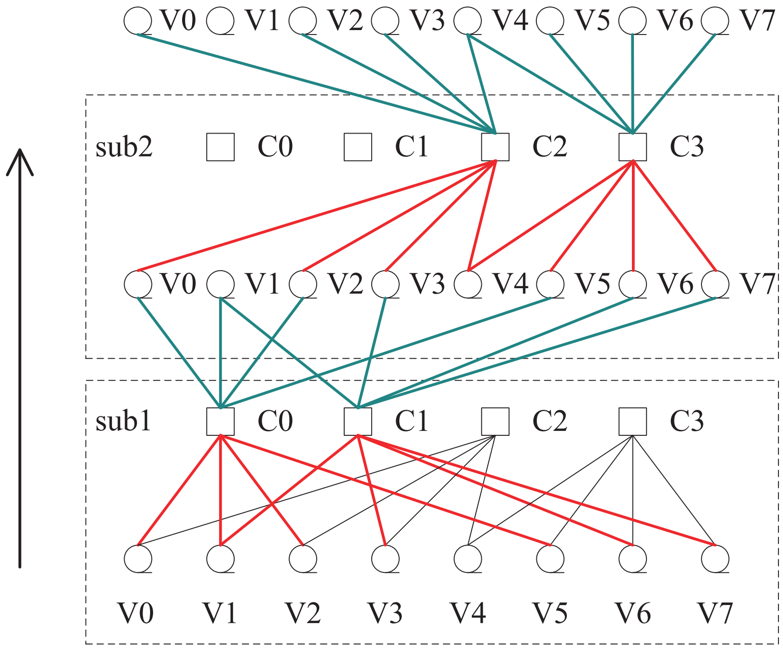


Fig.7. Message updating schedule of layered algorithm

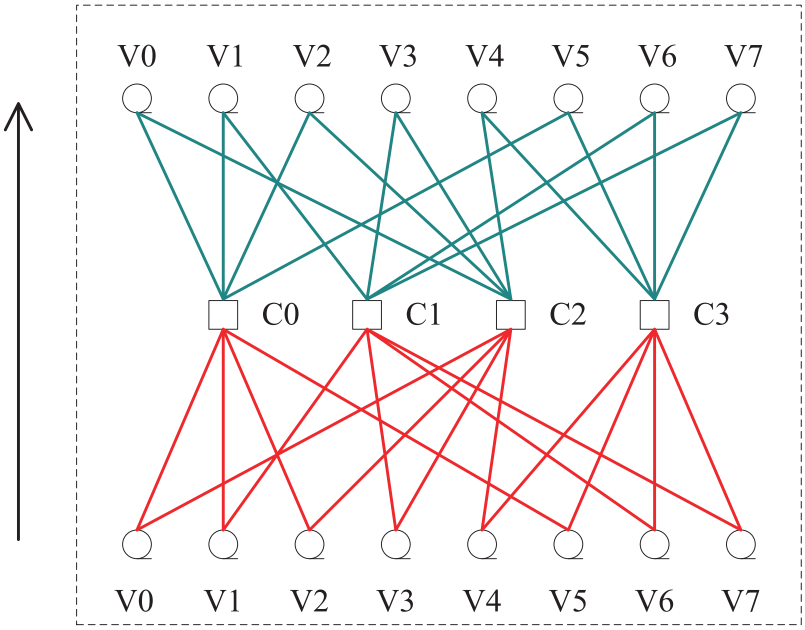


Fig.8. Message updating schedule of TPMP algorithm

Following steps shows the detailed process of the layered decoding algorithm:

[Initialization]

Calculate the log likelihood ratio (LLR) γ_n for $\mathrm{n} = 1, 2, \dots, \mathrm{N} - 1$ using equation (2.6), where σ^2 is the variance of noise generated by the AWGN Channel.

Set $\beta_{mn} = \lambda_n$ for each $(\mathrm{m,n})$ satisfying $\mathrm{H}_{mn} = 1$. Set the loop counter $l = 1$ and the maximum number of iterations is set to l_{max}. Set intrinsic message $\lambda_{mn} = 0$.

Step 1 [Layer Update]

For each layer: a subtraction calculates the difference, then the new intrinsic messages and APP is calculated with equation (2.13), (2.14), (2.15):

$$\rho_{mj} = \gamma_j - \lambda_{mj}[l-1] \tag{2.13}$$

$$\lambda_{mj}[l] = (\prod_{\mathrm{n\in A(m)\backslash j}} sign(\rho_{mn})) \times f(\sum_{n\in \mathrm{A(m)\backslash j}} f(|\rho_{mn}|)) \tag{2.14}$$

$$\gamma_j = \rho_{mj} + \lambda_{mj}[l] \tag{2.15}$$

Step 2 [Hard-Decision]

Compute all the tentative LDPC codeword bits $\overset{\wedge}{\gamma_n}$ for $\mathrm{n} = 1, 2, .., \mathrm{N}$.

$$\overset{\wedge}{\gamma_n} = \begin{cases} 0, sign(\gamma_n) = +1 \\ 1, sign(\gamma_n) = -1 \end{cases} \tag{2.16}$$

Step 3 [Parity-Check]

If the tentative hard-decision vector $(\hat{\gamma}_1, \hat{\gamma}_2, \cdots, \hat{\gamma}_N)$ satisfies the parity-check equation (2.9), output the codeword, and terminate the decoding iteration. If not, go to step 1.

$$H \cdot (\hat{\gamma}_1, \hat{\gamma}_2, \cdots, \hat{\gamma}_N)^T = 0 \tag{2.17}$$

Step 4 [Maximum Iteration Check]

If $l \le l_{\max}$, set the loop counter $l \leftarrow l+1$ and go to Step 1. Otherwise output the decoded codeword $(\hat{\gamma}_1, \hat{\gamma}_2, \cdots, \hat{\gamma}_N)$ and stop decoding.

Since the layered decoding algorithm accelerates the convergence speed for decoding LDPC codes and is especially suitable for multi-mode LDPC decoder implementation, the contributions in this dissertation are all based on layered decoding algorithm.

2.4 Partial-parallel LDPC Decoder Architecture

The general partial-parallel layered LDPC decoder architecture is shown in Fig.9. Not limited to QC-LDPC codes, all the structured LDPC codes which can be divided into layers and for each layer the corresponding CN message updating functions can be independently processed, can utilize this decoder architecture.

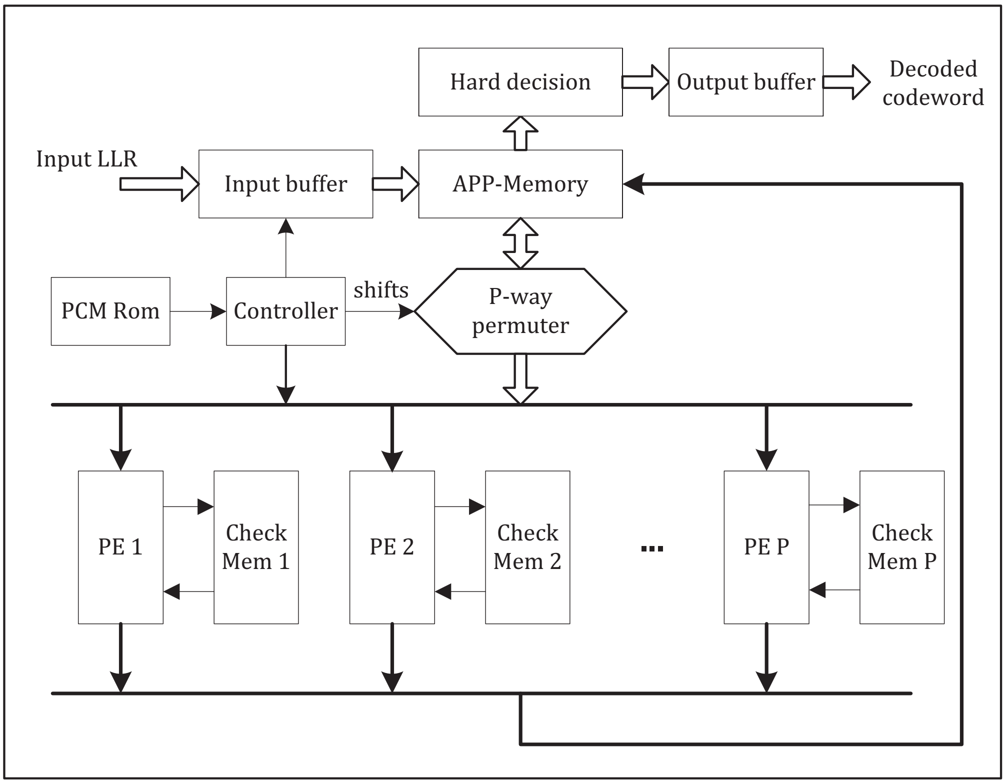


Fig.9. Top-level architecture for partial-parallel layered LDPC decoders

The functions for important modules are described as follows.

1) **APP-Memory**: APP message memory, store APP messages with bit-width of w, where w is the APP message quantization bit size;

2) **PE {1, 2, ..., P}**: Processing elements, paralleled execute the layer message updating functions for each CN of each layer;

3) **Check memory {1, 2, ..., P}**: Check-node message memory, exchange messages directly with corresponding PEs to update CN messages;

4) **PCM Rom**: Store the parity-check matrix information in a

particular way;

5) **Controller**: Generate all the addresses for memories and control the iterative, pipelined decoding process;

6) **P-way Permuter**: Permute a number of P APP messages to corresponding PEs, which allows changeable connections between CNs and VNs, enabling multi-mode decoding. For QC-LDPC codes, P is equal to the expansion factor and it is usually constructed with barrel shifters.

The pipelined iterative decoding schedule is that in each clock cycle a proper APP message block is fetched from APP Memory and goes through the P-way Permuter for a barrel shift operation, and then goes into all the PEs waiting for layer messages updating with proper check messages. After a whole layer inputted and calculated, the updated APP messages and check messages will be written back to the corresponding addresses. The read and write operations will not be stopped until a data hazard occurs.

The partial-parallel layered decoder architecture is widely adopted in such as WiMAX and Wi-Fi decoders because of its moderate parallelism and capability for multi-mode decoding. However, the data dependency problem between adjacent layers limits the parallel level, which restricts this architecture from achieving higher parallelism for high-speed applications.

3. A BER Performance-Aware Early Termination Scheme for Generic Layered LDPC Decoders

Besides the decoding schedules, early termination (ET) scheme is another hot topic in LDPC decoding field since it can tremendously increase the decoding throughput or reduce the power consumption. Many works have been done for both traditional TPMP decoder architecture and layered decoder architecture. Early termination in layered decoder architecture is different to that in TPMP one since the APP messages are updated more than once in a single iteration of layered decoding process which are only updated once in TPMP. For ET schemes in layered decoding schedule, previous works have been done. [15][16] Both of them can terminate the decoding rapidly without disturbing the pipelined layered decoding schedule. However, they bring significant BER performance degradations, which worsen the decoding performance and cannot be applied in BER performance-aware applications, such as DVB-S2 standard.

In this chapter, we propose a BER performance-aware ET scheme for generic layered LDPC decoder. Unlike the previous solutions, it terminates the decoding process based on parity-check functions to ensure successful decoding. By using the proposed scheme, the lossless BER performance and fast termination speed are achieved with acceptable hardware overhead. As a hardware module, the proposed ET scheme is not limited to any standards or check-node message updating functions.

3.1 Problems in Previous ET Schemes for Layered LDPC Decoder Architecture

Due to the difference between TPMP and layered decoding schedule, traditional early termination schemes, which verify the satisfaction of all the parity-check equations, are not precise anymore for the layered decoding architectures since the hard-decisions of APP messages are frequently changed.

The termination criteria of the previous works [15][16] are introduced and also the problems of them are discussed.

3.1.1 HDA-ET Criterion

The most famous ET scheme is the popularly used hard-decision aided (HDA) ET scheme which is originally implemented in turbo decoder. It temporarily stores the hard-decisions and compares the decoded codeword in two successive iterations. If the decoded codeword of two iterations are the same, the decoding will be terminated. Otherwise, the decoding process will be terminated when a predefined maximum iteration time is reached. Since this scheme cannot guarantee acceptable BER performance, sometimes an additional constraint is added to enhance the reliability, in which all the magnitude values of APP are larger than a predefined threshold. The threshold value for HDA-ET varies from different channel conditions and LDPC codes thus it needs to be determined via simulations for better BER performance.

The mechanism for HDA-ET scheme is simply based on observations that in successful decoding the average magnitudes of APP messages are growing faster during iterations and hard-decisions converge to the true codeword. On the opposite, in failed decoding the average magnitudes of APP messages are oscillated and small.

However, two special cases sometimes happen. One is that the hard-decisions converge to a pattern with small number of errors. The HDA-ET criterion terminates the decoding before reaching the maximum iteration time but cannot report a unsuccessful decoding. Another is that the hard-decisions already converge to the true codeword, but several APP cannot get larger due to quantization effect. In such case the decoding cannot be terminated until reaching the maximum iteration time.

Obviously, bigger APP threshold values make the prediction much reliable. They greatly affect the termination speed while cannot get rid of BER performance degradation effectively. Moreover, the HDA-ET scheme makes the error floor occurred quite early. This worsens the error-correcting performance especially in high SNR conditions.

3.1.2 CSC-ET Criterion

Another ET scheme for layered decoding in [16], which we name it as current-iteration satisfied check early termination scheme (CSC-ET), examines the parity-check equations layer by layer with the decoding. When all the parity-check equations are satisfied sequentially in a single iteration, the decoding process will be terminated immediately (CSC-ET_0) or after additional n iterations (CSC-ET_n) for better BER performance.

Since the updating of each APP message is more than once in a single iteration, such kind of serial satisfaction of parity-check equations always utilizes temporary hard-decisions of APP messages. It is not equal to the successful decoding criterion, in which all the parity-check equations should be satisfied at the same time. Fig.10 shows an particular case, in which the CSC-ET constraint is met but it is not correctly decoded.

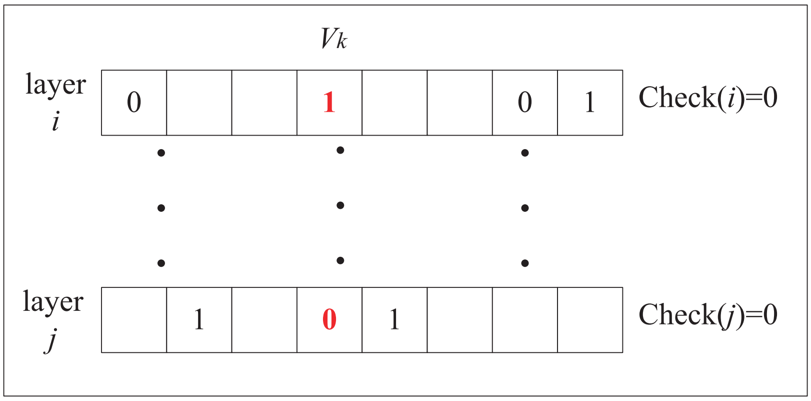


Fig.10. CSC-ET criterion problem

In Fig.10 layers i and j are two check-nodes in PCM. Positions with numbers mean that there are connections between variable-nodes and check-nodes. The number ‘0’ represents positive value of APP, while ‘1’ represents negative one. Variable-node V_k connects to both of them. When the decoder processes on layer i

and layer *j* respectively, the parity-check equations *i* and *j* are both satisfied, but the hard-decision V_k is updated from '1' to '0'. Thus when the parity-check equation *j* is processed, the already satisfied parity-check equation *i* becomes unsatisfied. Through simulations we find that the above phenomenon does happen in the codeword terminated by CSC-ET and it will worsen the BER performance.

Similar situations will occur in CSC-ET criterion as HDA-ET. For successful decoding, there can be flipping hard-decisions which make the intermediate parity-check functions be unsuccessful. In such case, termination speed will be slowed down. Another case is as shown in Fig.10 that CSC-ET criterion may introduce extra errors.

Processing extra iterations when CSC-ET criterion is met can improve BER performance effectively since the sequential satisfaction of parity-check equations is a powerful indicator for codeword which can be successfully decoded. But it is still not equal to successful decoding, thus involves error-correcting performance loss.

3.2 Last-iteration Satisfaction Check Early Termination (LSC-ET) Scheme

By investigating the unsatisfied-constraint problem of CSC-ET, for layered decoder architecture we propose the last-iteration satisfaction check based early termination (LSC-ET) scheme. Unlike the previous schemes, the proposed one can terminate the decoding process without any BER performance loss. Moreover, compared with above two ET schemes, the proposed one achieves smaller average iteration times (AIT) which greatly influences the decoding throughput or power consumption.

The main idea of the proposed LSC-ET scheme is that the decoder executes the parity-check equations during *(i+1)th* iteration with hard-decisions generated from *ith* iteration. The decoding process is terminated only if all of the parity-check equations in a single iteration are satisfied. Otherwise it will be terminated after predefined maximum iteration time. As the CSC-ET scheme, it verifies the parity-check equations layer by layer. The only difference is that LSC-ET utilizes hard-decisions generated from last iteration which CSC-ET uses the ones from current iteration.

Since all of APP messages are updated more than once in a single iteration, we should memorize the last-updated hard-decisions in a single iteration. A simple example shown in Fig.11 explains how our method memorizes the last-updated values. Considering each row as a layer in the PCM, layers 0 to 5 are decoded sequentially. The ‘1’s marked with bold red are last-updated positions for each variable-node.

layer 0	1	0	**1**	0	0	1	0	0	**1**	0	0	0
layer 1	0	0	0	0	0	0	0	1	0	**1**	0	0
layer 2	0	0	0	0	0	**1**	0	0	0	0	1	0
layer 3	1	0	0	**1**	0	0	**1**	0	0	0	0	1
layer 4	0	**1**	0	0	1	0	0	**1**	0	0	0	**1**
layer 5	**1**	0	0	0	**1**	0	0	0	0	0	**1**	0

Fig.11. Last-updated information in PCM

This method can ensure that all the hard-decisions of variable-nodes are only updated once and they are the latest values. The LSC-ET scheme is described as follows:

a) Enlarge one bit (hard-decision bit) for all APP messages and initialize them as zero;

b) When decoding to each layer, only for marked variable-nodes the hard-decision bits are updated according to updated APP messages;

c) Without in the first iteration, process parity-check equations with the hard-decision bits in all layers;

d) When all of parity-check equations are satisfied in a whole iteration, stop the decoding process.

For the satisfaction check in *(i+1)th* iteration, the last-updated hard-decisions obtained from *ith* iteration should be passed to the correct PEs, so they are combined with APP messages together for convenience. In real implementation, we enlarge the data bit-width of both APP memory and permutation network by one bit to achieve this function. Compared to additional parity-check logics, the increased hard-decision bits for APP messages occupy the most part of hardware overhead of LSC-ET scheme. For example in DVB-S2 decoders, the total memory overhead is no larger than 3% (8 bit quantization for APP messages) and it is the same level as HDA-ET, while hardware overhead of CSC-ET scheme can be ignored compared with our proposed one.

It is noteworthy that the proposed LSC-ET scheme processes exactly the same parity-check equations as the traditional parity-check case, so that it can ensure BER lossless performance and exactly report the status of decoding. For the FEC-ARQ based systems such as WiMAX, in which the status of decoding is required, extra parity-check logics should be implemented in the decoder. In such case, the LSC-ET scheme provides the best solution whatever in error-correcting performance, termination speed and hardware complexity.

3.3 Simulation Results and Comparisons

Under the condition of BPSK modulation, AWGN channel and 8-bit fixed point quantization, the BER and 1/AIT simulations for CSC-ET, HDA-ET and LSC-ET schemes have been done. Normalized Min-Sum layered algorithm with factor 0.75 is used to perform iterative decoding. A kind of long LDPC code: (64800 bits code-length, 1/2 code-rate) specified in DVB-S2 and a kind of short LDPC code: (2304 bits code-length, 1/2 code-rate) specified in WiMAX are used throughout the simulations.

The results of BER performance of both codes are shown in Fig.12 and Fig.13. CSC-ET_n represents that after satisfaction of the CSC-ET criterion the decoder will process extra n iterations. Also HDA-ET schemes with two different thresholds are simulated. It is clear that from the BER~SNR curves the proposed LSC-ET scheme achieves the best BER performance.

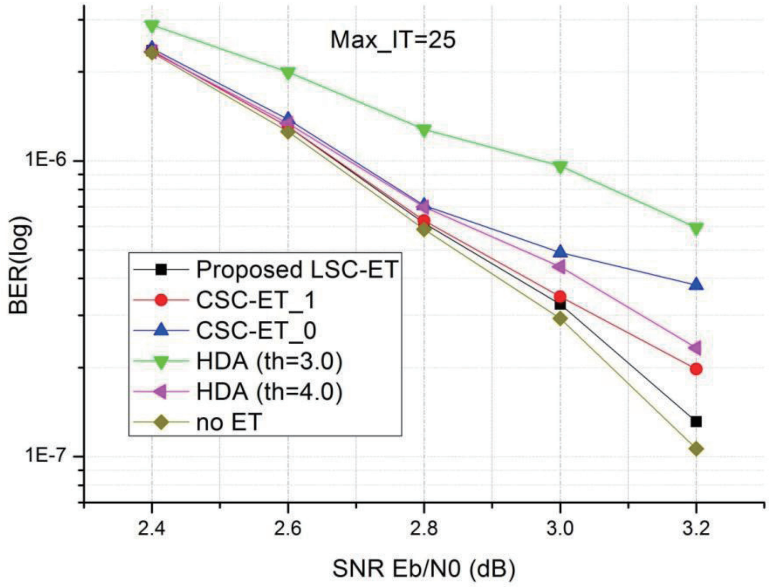


Fig.12. BER performance comparison (64800, 1/2)

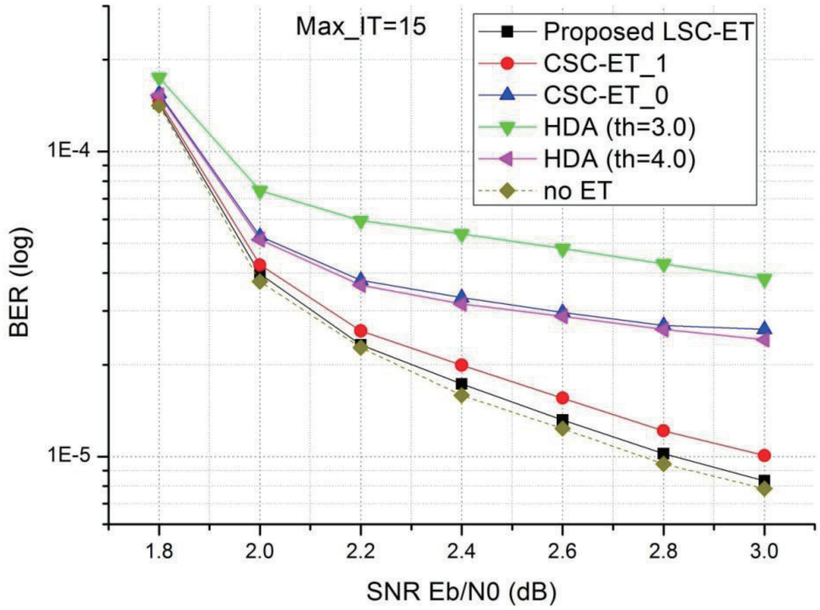


Fig.13. BER performance comparison (2304, 1/2)

Another simulation is for the termination speed. We draw 1/AIT~SNR curves in Fig.14 and Fig.15 where the results of our proposed scheme are unified to one for comparison. From the figures we can find that among all the ET schemes our proposed one achieves very good termination speed. Compared with the second best scheme CSC-ET_1 which achieves slightly worse BER performance, the decoding time reduction is about 5~10% in WiMAX 1/2 rate case, and 2~5% in DVB-S2 1/2 rate case. Although the CSC-ET_0 scheme can terminate faster than LSC-ET scheme, the huge BER performance degradation makes it impractical in real applications.

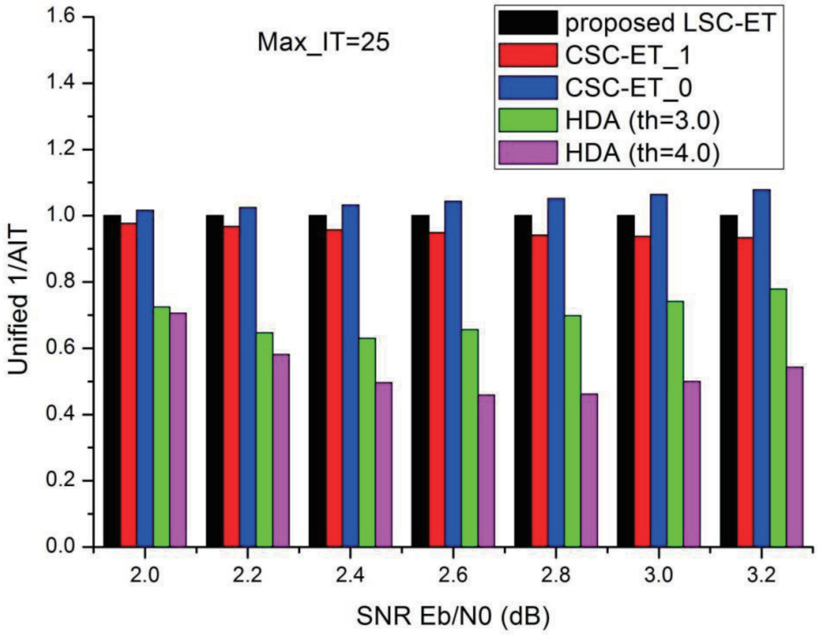


Fig.14. AIT performance comparison (64800, 1/2)

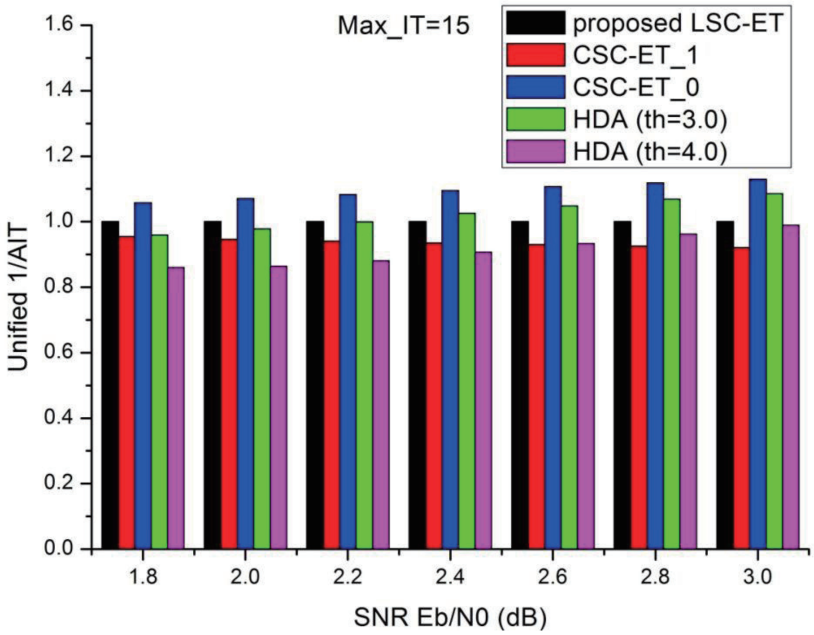


Fig.15. AIT performance comparison (2304, 1/2)

Considering both the termination speed and BER performance, the LSC-ET scheme is the best among the three. CSC-ET_0 scheme can terminate the decoding faster but suffers significant performance loss. The small gap between LSC-ET and no ET performance on the figure is because that LSC-ET does not process parity-check equations for the decoding results of the final iteration. In layered decoding process, an addition iteration processed after a successful decoding may cause unsatisfied parity-check result. The no-et BER performance can be achieved by storing the hard-decisions of the second last iteration into the output buffer registers in the FEC-ARQ systems to get the successful decoding flags.

Table 1 shows the unified 1/AIT for the no-ET scheme of WiMAX 1/2 code based on maximum iteration number set to 15. At the proper SNR range around 2.4dB for rate 1/2, the iteration number reduction for the proposed LSC-ET scheme compared to no-ET scheme is about 60%. It is directly proportional to the decoding throughput or energy-efficiency improvement usually, so that it can largely improve the decoding throughput or energy-efficiency.

Table 1 Unified 1/AIT for WiMAX 1/2 code with no-ET scheme

Rate 1/2 WiMAX code with no-ET	
SNR(dB)	Unified 1/AIT
1.8	0.563
2.0	0.493
2.2	0.439
2.4	0.399
2.6	0.366
2.8	0.341
3.0	0.319

The iteration number reduction is much higher than such as DVB-S2 standard codes which require larger maximum iteration number for better error-correcting performance, and also higher than larger code-rates with the same code length since the average iteration number is always smaller compared to smaller code-rate cases.

3.4 Summary

In this chapter, a BER performance-aware early termination scheme for generic layered LDPC decoder architecture is proposed. Comparison results demonstrate that based on the same experimental conditions the proposed LSC-ET scheme achieves the best BER performance among all of the ET schemes with very good termination speed and acceptable hardware overhead. For applications which require exact decoding failure status, the LSC-ET scheme provides the best solution whatever in error-correcting performance, termination speed and hardware complexity. Moreover, since the proposed LSC-ET scheme does not cause extra error floor, it is the best solution for applications which require strict BER performance and appreciate early termination for lower power consumption or higher decoding throughput.

4. DVB-T2 LDPC Decoder with Perfect Conflict Resolution

4.1 Introduction

As introduced in chapter 2, the IRA-LDPC codes, which benefit from linear encoding complexity and relatively simple structure, have already been adopted in long distance communication systems such as DVB-T2 [10], DVB-S2 [11] and ISDB-S2 [37].

For the structured IRA-LDPC codes, many researchers also attempt to implement with the layered algorithm for efficiently partial-parallel decoding and multi-mode compliant solutions. [20]-[24] However, commonly generated 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 has been already shown in previous works. The error-correcting performance degradation can be larger than 0.2dB in SNR compared with conflict-free layered decoding performance [24]. In order to solve this problem, authors in [20] and [21] decoded the conflict layers with TPMP algorithm while authors in [22] and [23] tried to modify the PCM of conflict layers. The author in [24] proposed a selective recalculation strategy adding to the layered decoding algorithm to achieve conflict-free error-correcting performance, but the proposed algorithm is quite not hardware-friendly due to its instability of complexity.

In this chapter, 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 adjacent layers to ensure high pipeline efficiency. Secondly, the Layer Division technique solves all remaining conflicts with a well-designed, overlapped decoding schedule. A synthesizable DVB-T2 LDPC decoder architecture is also introduced in this dissertation to demonstrate detail implementations.

The remainder of this chapter is organized as follows: Section 4.2 briefly describes the arising message updating conflict problem when applying layered algorithm to DVB-T2 LDPC codes. Previous solution is also discussed in this section. In Section 4.3, the proposed PCM Rearrange and Layer Division techniques are discussed in detail. The proposed decoder architecture is presented in Section 4.4. Section 4.5 demonstrates the comparison results of error-correcting performance and implementation results.

4.2 Message Updating Conflict in Layered DVB-T2 LDPC Decoder

In this section, firstly, the LDPC code structure adopted in DVB-T2 is introduced. Secondly, the arising message updating conflict problem while applying layered algorithm to DVB-T2 LDPC codes is discussed. At last, the involving problems of previous solution is discussed.

4.2.1 LDPC Codes in DVB-T2 Standard

DVB-T2 is the second generation digital terrestrial television broadcasting system. In such kind of channel conditions, the Forward Error Correction (FEC) only technique adopted in the system is designed to provide a "Quasi Error Free" (QEF) quality target, approximately with $\mathrm{BER} < 10^{-9}$. The code -lengths for DVB-T2 LDPC codes are 64800 bits for normal frame and 16200 bits for short frame, the same as DVB-S2 LDPC codes. For different channel conditions, there are 6 and 7 kinds of code rates for normal and short frame out of those from DVB-S2, respectively.

LDPC codes in DVB-T2 standard belong to systematic IRA-LDPC codes. For DVB-T2 LDPC codes specified with code rate $\mathrm{R}=\mathrm{K}/\mathrm{N}$, the codeword c is defined as $c=(i_0,i_1,...,i_{K-1},p_0,p_1,...,p_{N-K-1})$, in which i and p represent information bits and parity bits, respectively. The PCM can be divided into several parts, as shown in Fig.16. 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_{Aj} are specified by row j of the corresponding address tables specified in [10] and the following $\mathrm{M}-1$ columns are generated by cyclically shift a number of q as shown in the figure, which is $\mathrm{q}=(\mathrm{N}-\mathrm{K})/\mathrm{M}$, a code-rate dependent constant. It shows the periodicity of M for each sub-matrix H_{Aj} since the first column can be obtained by cyclically shift value q to the *M-th* column for each H_{Aj}. Matrix H_B, which represents the connections between parity nodes and check-nodes, is a staircase matrix. Values of q according to code-rates specified in DVB-S2 standard are listed in Table 2, in which bold ones are selected in DVB-T2 standard.

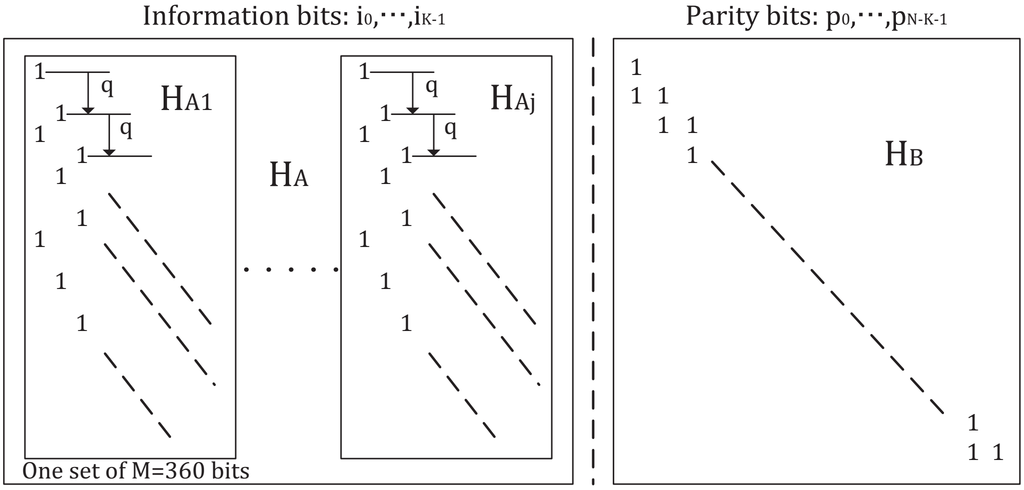


Fig.16. PCM of DVB-T2 LDPC codes

The periodical feature of the PCM makes it possible for partial-parallel decoding. The fixed periodicity $\mathrm{M}=360$ in DVB-T2 LDPC codes enables decoding parallelism up to 360. Through very simple row reorder process based on modulo operations to the row indices, the PCM can be shown as QC-like matrix [21], 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.17. The MSI blocks with several diagonals, which do not exist in normal QC matrices, leads to message updating conflict problem in layered decoding.

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

(a) all-zero block

0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1
1	0	0	0	0	0

(b) shifted identity block

0	1	0	1	0	0
0	0	1	0	1	0
0	0	0	1	0	1
1	0	0	0	1	0
0	1	0	0	0	1
1	0	1	0	0	0

(c) MSI block

Fig.17. Three types of blocks in QC-like PCM

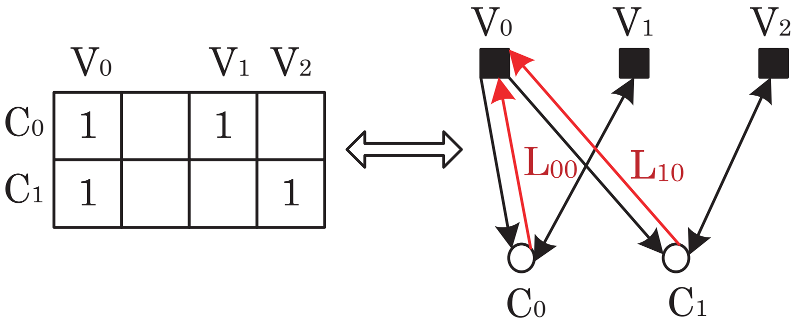
Table 2 Values of q and code-rates of DVB-S2/T2 LDPC codes

Code-rate	q of normal frame	q of short frame
1/4	135	**36**
1/3	120	30
2/5	108	27
1/2	**90**	**25**
3/5	**72**	**18**
2/3	**60**	**15**
3/4	**45**	**12**
4/5	**36**	**10**
5/6	**30**	**8**
8/9	20	5
9/10	18	N/A

4.2.2 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.18, which demonstrates an example of cutting edge with both PCM Fig.18 (a) and bipartite graph Fig.18 (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.18 (b) send the updated extrinsic messages L_{00} and L_{10} calculated with equation (2.14) to variable-node V_0 for APP_0 updating with equation (2.15) concurrently, but only one of them can participate in the equation. As a result, either of the two edges is cut. This problem is called the message updating conflict problem and forces parallel level degradation or performance loss. The error-correcting performance degradation can be larger than 0.2dB in SNR compared with conflict-free layered decoding performance, which is even worse than decoding with TPMP algorithm. To implement the layered algorithm, this problem must be solved first.



(a) PCM

(b) Bipartite graph

Fig.18. Example for message updating conflict

4.2.3 Problems in Previous Conflict Resolutions

Among the previous DVB-S2/DVB-T2 decoder architectures, the most practical conflict resolution is from [21] in which the authors decode the conflict layers with TPMP and layered mixed manner. In their solution, the MSI blocks or conflict blocks is divided into two common cyclic sub-matrices and processed with modified partial-parallel layered decoder architecture for QC-LDPC codes.

From the hardware cost point of view, The architecture proposed in [21] requires 50% extra arithmetic and control logics in the processing units to process the TPMP functions, and extra one clock cycle latency for each APP message updating route.

Since the number of conflict blocks are not so big that this solution does not have significant degradation on convergence speed or error-correcting performance. The detail simulation result to show the degradation from pure layered algorithm will be shown in section 4.5.

4.3 Resolution to Message Updating Conflict for DVB-T2 LDPC Decoders

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.

4.3.1 PCM Rearrange

To achieve the minimum decoding throughput of 90Mbps required by DVB-T2 system, dozens of parallel processors instead of 360 are enough. [22] 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 $\mathrm{M} = 360$ 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 $P = M/\varepsilon$. 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 $\varepsilon \times q$ 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 Sjk as their corresponding left barrel shift values. Each number X of the corresponding row of the address table generates a number of ε shifted identities which can be calculated using equation (4.1) as follows.

$$\begin{cases} for\ k = 0,1,\ldots,\varepsilon - 1 \\ j_k = (X + kq)\operatorname{mod}(\varepsilon q) \\ S_{jk} = \lfloor (X + kq)/(\varepsilon q) \rfloor \end{cases} \tag{4.1}$$

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 3. 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 3. 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 is optimal to help with the resolution of remaining conflicts, which will be discussed in the following section.

Table 3 Number of conflicts in DVB-T2 normal frame

	ε											
R	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

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 equation (4.1) 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. 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 $\varepsilon = 9$, the PCM information size for all six code-rates is about 600k bits, 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 equation (4.1). Considering the ε identities generated by equation (4.1), 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 $1/\varepsilon$ by only storing PCM information for each one of ε layers.

From equation (4.1), by modulo and divide q to the row indices we can derive equation (4.2) as follows:

$$\begin{cases} for\ k = 0,1,\ldots,\varepsilon-1 \\ j_k^* = j_k \bmod q \\ = [X \bmod (\varepsilon q)] \bmod q \\ = X \bmod q \\ \\ r_k^* = \lfloor j_k / q \rfloor \\ = (\lfloor [X \bmod (\varepsilon q)] / q \rfloor + k) \bmod \varepsilon \\ = (\lfloor X / q \rfloor + k) \bmod \varepsilon \end{cases}, \tag{4.2}$$

in which we can find that j_k* do not depend on k, and r_k* are different for each k. Therefore, we treat each ε × ε 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_k* and r_k* through equation (4.2), 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 equation (4.3) as follows:

$$\begin{cases} for\ k = 0,1,\ldots,\varepsilon-1 \\ J = (X \bmod q) \times \varepsilon + (\lfloor X / q \rfloor + k) \bmod \varepsilon \\ K = b \times \varepsilon + k \\ S_{JK} = \lfloor (\lfloor X / q \rfloor + k) / \varepsilon \rfloor \end{cases}. \tag{4.3}$$

Fig.19 (a) shows a non-zero MB example in the rearranged seed matrix H_A* which is generated in the case of ε = 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 feature is also eliminated in the rearranged seed matrix H_B*, which is shown in Fig.19 (b).

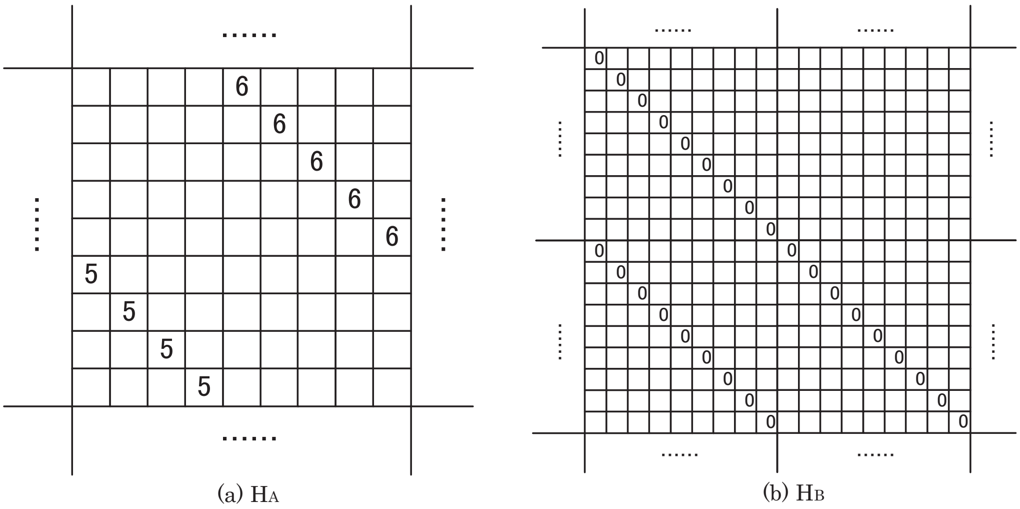


Fig.19. Seed matrix of DVB-T2 after rearrangement

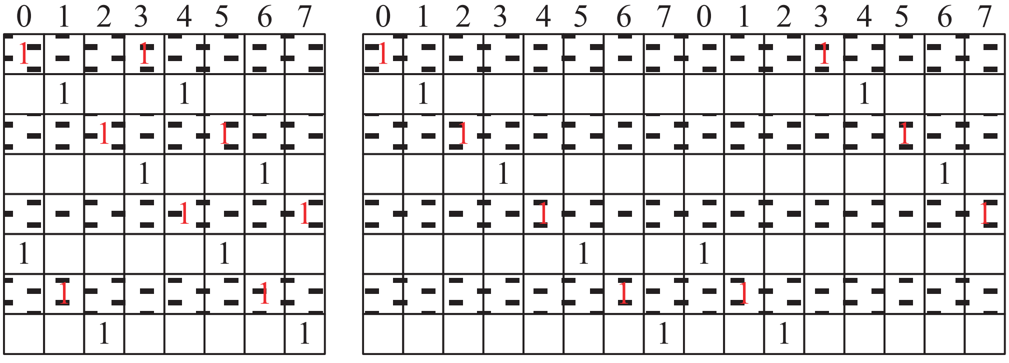
With this proposed PCM rearrange scheme, only information of one non-zero block should be memorized for each ε × ε non-zero MB, and at the same time the data dependency problem between adjacent layers can be solved to ensure the pipeline efficiency in layered decoding.

4.3.2 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.20 (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.20 (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.20, 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 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) (b)

Fig.20. Layer Division for 8x8 conflict block

4.3.2.1 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 equation (4.4) as follows:

$$d=\left|S_1-S_2\right|, \tag{4.4}$$

in which S$_1$ and S$_2$ are shift values for the two diagonals. Obviously, P must be an even number to ensure equally divisions and $d\in D=\{1,2,...,P/2\}$ 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.21, which will not affect the result.

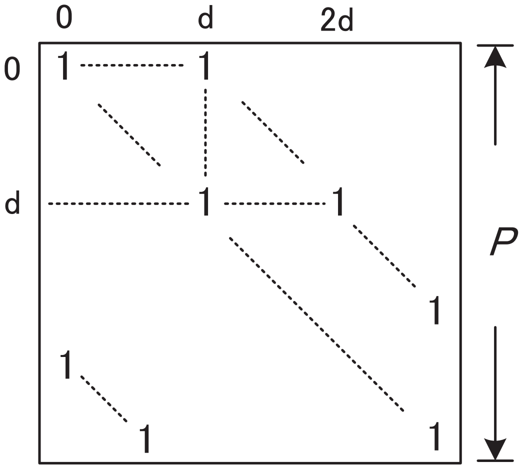


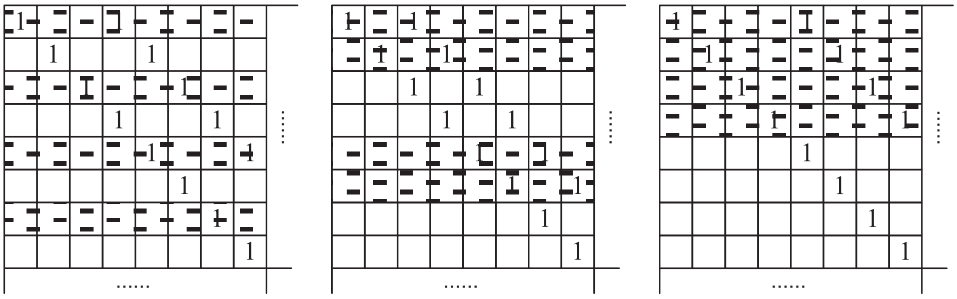
Fig.21. Example of (P, d) conflict pattern

Let us suppose these P rows can be divided into two sets: SET1 and SET2, and initially put Row(0) into SET1. Since both of Row(d) and Row(0) contain elements in Column(d), Row(d) must belong to SET2 to avoid conflicts. Similarly, Row($2id \bmod P$) belong to SET1 while Row($(2i+1)d \bmod P$) belong to SET2 with increasing integer i. This recursion of allocation will be finished when the coefficient $k = 2i$ or k=$2i+1$ can make $kd \bmod P = 0$, 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.22.

```
Initialize $SET_1 = \Phi, SET_2 = \Phi, U_d = \Phi, \delta = 0, P, d$
(i) if $d \bmod 2 = 1$ and $P \bmod 2 = 0$
        $\forall x \begin{cases} Row(x) \Rightarrow SET_1 \mid \left\lfloor \frac{x}{2^\delta} \right\rfloor \bmod 2 = 0 \\ Row(x) \Rightarrow SET_2 \mid \left\lfloor \frac{x}{2^\delta} \right\rfloor \bmod 2 = 1 \end{cases}, Successful$
(ii) else if $d \bmod 2 = 0$ and $P \bmod 2 = 0$
        $d/2 \to d, P/2 \to P, \delta + 1 \to \delta, Goto\ (i)$
(iii) else if $P \bmod 2 = 1$
        $d \times 2^\delta \Rightarrow U_d, Failed$
```

Fig.22. Division algorithm for (P, d) conflict block

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.23 (a), (b) and (c), which are enough to cover all the conflict blocks.



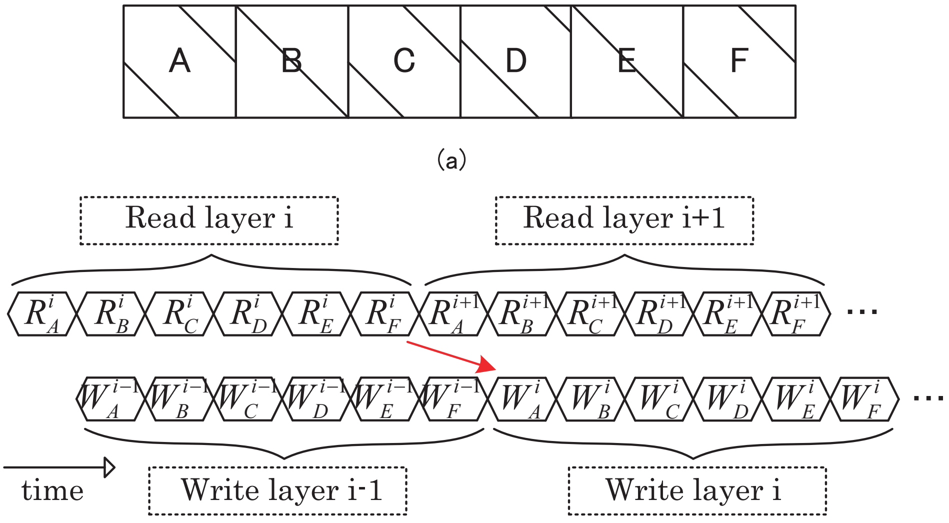
$a)\ d \bmod 2 = 1$ $b)\ d \bmod 4 = 2$ $c)\ d \bmod 8 = 4$

Fig.23. Three division patterns and corresponding d values for $P = 40$

4.3.2.2 Pipelined Conflict Layer Decoding with Maximal Overlapping

After the conflict layers are successfully divided, the remaining 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 convergence speed 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.24 (a). Fig.24 (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".


A
B
C
D
E
F
Read layer i
Read layer i+1
R_A^i R_B^i R_C^i R_D^i R_E^i R_F^i R_A^{i+1} R_B^{i+1} R_C^{i+1} R_D^{i+1} R_E^{i+1} R_F^{i+1} …
W_A^{i-1} W_B^{i-1} W_C^{i-1} W_D^{i-1} W_E^{i-1} W_F^{i-1} W_A^i W_B^i W_C^i W_D^i W_E^i W_F^i …
time
Write layer i-1
Write layer i

(a)

(b)

Fig.24. Timing diagram of fully-overlapped pipeline for normal layers

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 processor 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.25 (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 criterions, the timing diagram for successive conflict layers is shown in Fig.25 (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.25 (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.

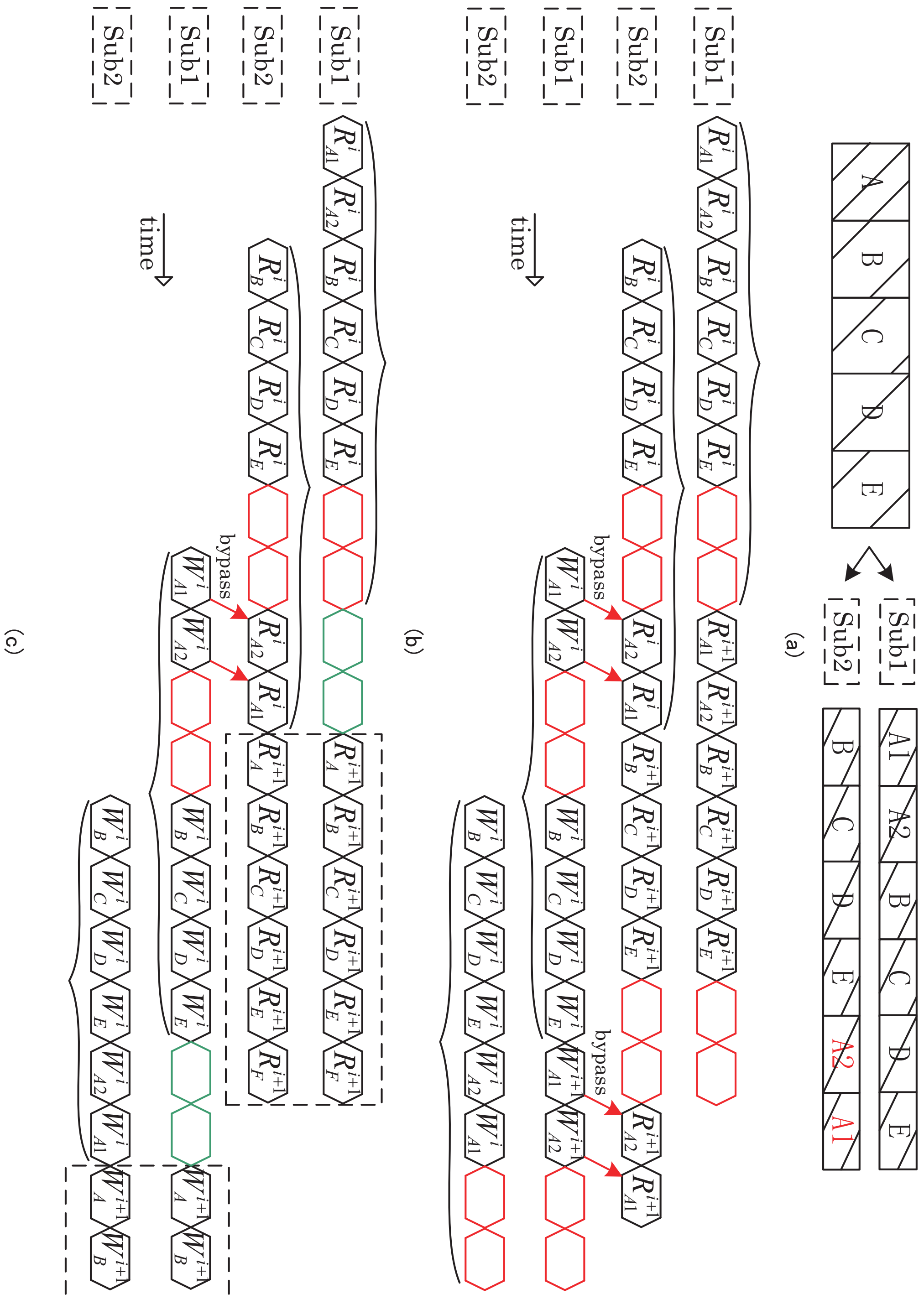


Fig.25. Timing diagram of overlapped pipeline for conflict layers

4.4 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.

4.4.1 Top-level Decoder Architecture

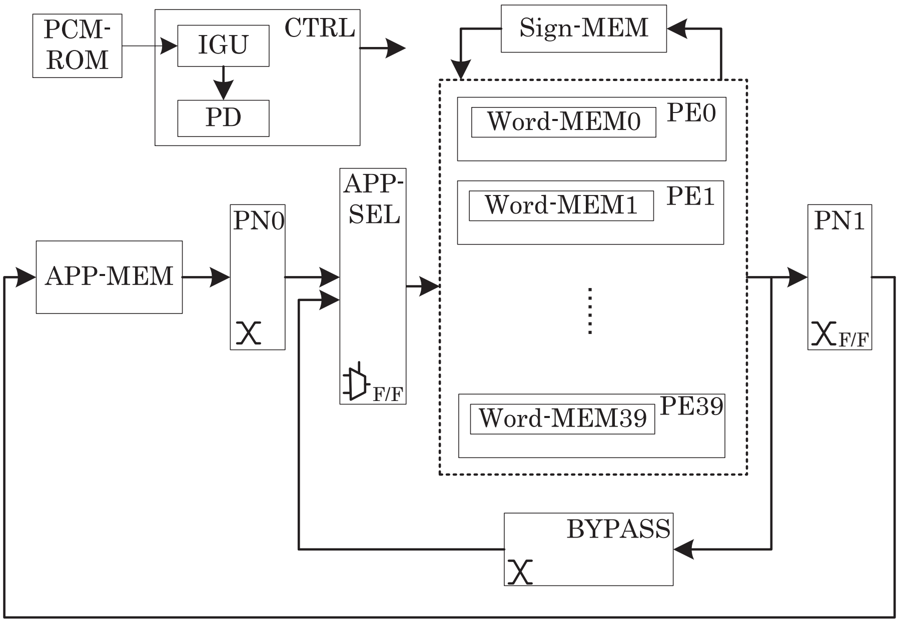


Fig.26. Top-level decoder architecture for DVB-T2

Fig.26 demonstrates the top-level decoder architecture based on the conflict resolution described in Section 4.3. This 40-parallel layered decoder can be divided into several modules which are listed as follows:

a) 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.

b) 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.

c) The APP-MEM dual-port memory module, which is initialized with received channel LLR, stores all the APP messages for updating.

d) The information generating unit (IGU), which is included in the controller module, recovers all PCM information.

e) 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.23.

f) 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.

g) The processor elements (PE) 0~39 are parallel processors for APP and extrinsic message updating.

h) The Word-MEM single-port memory with 40 cuts, which is included in each PE, stores the compressed extrinsic messages without sign bits.

i) The Sign-MEM dual-port memory module stores all the sign bits of extrinsic messages.

j) The bypass unit (BYPASS), which follows the assumption in Section 4.3, 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 $S_2 - S_1$.

k) 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 equation (4.5) with the normalized factor equal to 0.75.

$$L_{ij}^{new} = \gamma \times \prod_{j' \in A(i) \backslash j} sign(E_{ij'}) \times \min_{j' \in A(i) \backslash j} \left| E_{ij'} \right| \tag{4.5}$$

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.

4.4.2 Particular Function Modules

4.4.2.1 Information Generating Unit (IGU)

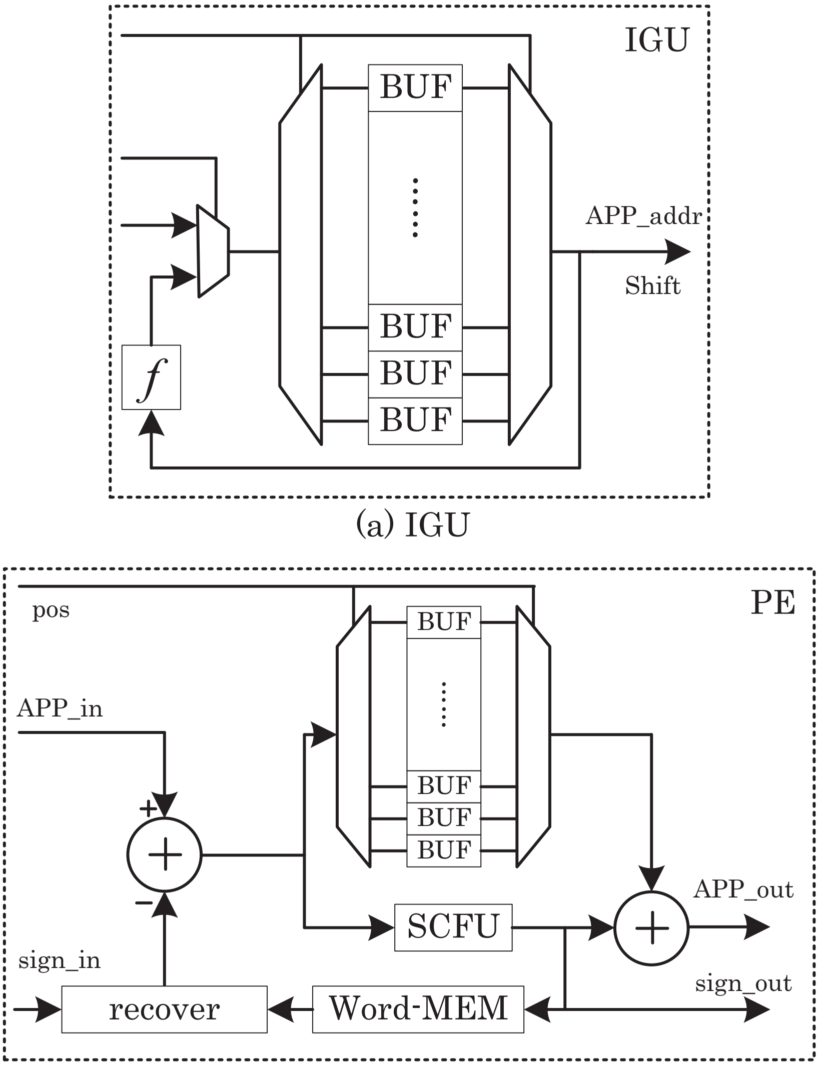
The IGU sub-module, which is designed to recover the PCM information of successive nine layers from those of one layer, is shown in Fig.27 (a). It is composed of 22 buffers for temporarily storing the APP addresses A and shift values S for non-zero blocks of one layer, and also a functional node *f* 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 equation (4.6) as follows.

$$(A,S)^{next} = \begin{cases} (A-8, S-1), \ if \ A \bmod 9 = 8 \\ (A+1, S), \ else \end{cases} \tag{4.6}$$

4.4.2.2 Processor Elements (PE)

Fig.27 (b) shows the structure of proposed PE, which can exactly follow the timing diagrams as shown in Fig.24 (b) and Fig.25 (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 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.



(a) IGU

(b) PE

Fig.27. (a) IGU structure and (b) PE structure

4.5 Results and Comparisons

4.5.1 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 [20]. Several state-of-the-art works on DVB-T2/DVB-S2 LDPC decoding apply the same conflict resolution strategy as in [20], 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. Fig.28 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 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.28 it can be observed that under the same SNR values the proposed strategy achieves a slight BER performance gain compared to that of [20]. At the same channel condition $Eb/N0 = 3.30dB$, for totally 200,000 test frames, the number of error bits are 24 and 16 for [20] 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 [20] still takes the advantage of fast convergence of layered algorithm since the message updating style between layers is still the same as layered algorithm.

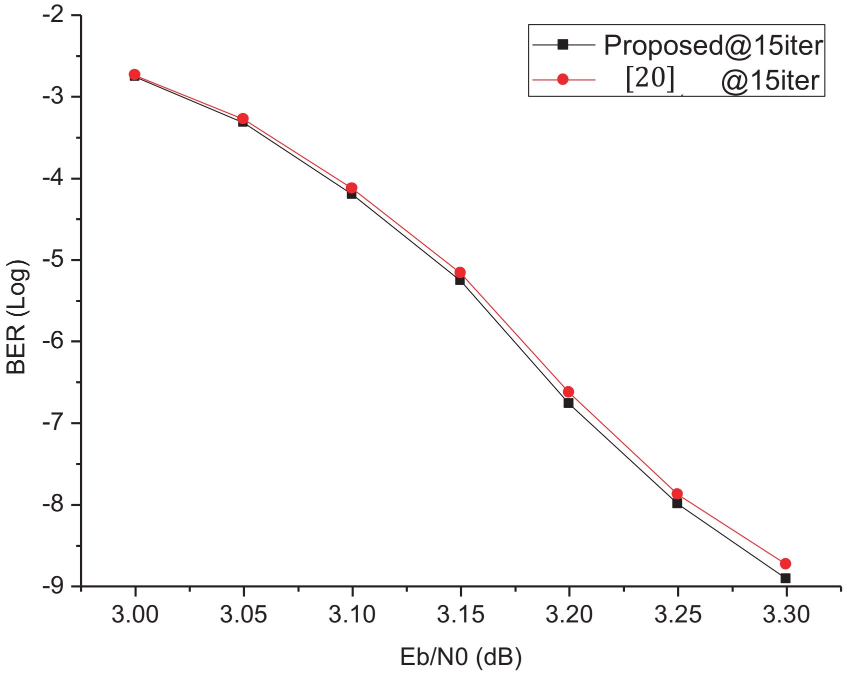


Fig.28. BER performance comparison of different strategies for rate 5/6 of DVB-T2 normal frame

Table 4 Implementation and comparison results

Parameter	Proposed	[21]
Standard	DVB-T2 normal frame	DVB-S2 FEC
Parallelism	40	90
Single-port RAM	553kb	3.59Mb
Dual-port RAM	803kb	
Gate count	193k	650k
Max frequency	319MHz	270MHz
Max iteration number	25	40
Air throughput	116Mbps	90Mbps×2
Check-node function	Normalized Min-sum	3-Min
Technology	90nm	90nm

4.5.2 Implementation Results

The decoder core is synthesized based on SMIC 90nm CMOS technology. Synthesis results obtained with Synopsis Design Compiler are listed in Table 4. For code rate 3/5, which has 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 [21] 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 [21] the memory 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 simple than that of [21]. Moreover, the proposed decoder saves 50% redundant arithmetic units and related circuits for TPMP and layered combined decoding in PEs of decoder [21]. From this consideration, the proposed work is optimized than the reference works in hardware utilization.

4.6 Summary

In this chapter, we propose a partial-parallel layered LDPC decoder architecture with a novel conflict resolution to the message updating conflict problem for decoding DVB-T2 LDPC codes. Two techniques are proposed to guarantee conflict-free layered decoding performance with reasonable parallelism and efficient pipeline. Although not all the IRA-LDPC codes can apply this strategy directly, with sacrifice of a little decoding time, the proposed techniques can be applied to avoid decoding in TPMP scheduling, which can achieve better performance and save decoder area indeed. Simulation result shows that compared to state-of-the-art works, the proposed decoder architecture can achieve pure layered error-correcting performance and convergent speed.

5. High-Parallel Performance-Aware LDPC Decoder for WiMAX

5.1 Introduction

The WiMAX standard adopts QC-LDPC codes as one of its FEC solutions. Multi-mode (multi-rate and multi-length) decoding capability and up to 1Gbps decoding throughput with high reliability are basic requirements for modern WiMAX LDPC decoders. LDPC decoders based on TPMP algorithm are not able to support multi-rate decoding. For example, the fully-parallel LDPC decoders [38][39] have potentials to achieve very high decoding throughput, but suffer from complicated interconnection.; the bit-serial fully-parallel decoder [39] achieves more than 3Gbps throughput with relatively good energy-efficiency and the interconnection can be largely simplified. One typical implementation for multi-mode QC-LDPC decoders is based on the partial-parallel layered decoder architecture [25]. In their work, only one sub-block can be processed per clock cycle. The parallelism is limited by the number of CNs in each layer or the sub-block size, and at least 76~88 clock cycles per iteration are required according to PCMs specified in WiMAX standard. The state-of-the-art work for partial-parallel layered decoder [26] applies two sets of processing units to increase the parallelism and process two sub-blocks in each clock cycle. By utilizing many techniques to avoid data dependency problem, they finally reduce the number of clock cycles per iteration to 48~54 for WiMAX. Our previously proposed bit-serial layered decoder [29] takes the advantages of the flexibility of layered scheduling and high parallelism of fully-parallel architecture. It achieves 24~48 clock cycles per iteration with improved energy-efficiency and area close to [26]. However, the message saturation problem is not solved efficiently, which requires extra bit-width for the interconnection (permutation network). Also the bit-serial arithmetic units occupy relatively small area compared with bit-parallel components while the related multiplexer (MUX) area on the bit-serial message updating datapath largely exceeds them. The above facts lower the efficiency of the fully-parallel architecture.

In this paper, we propose new fully-parallel layered decoder architecture to support multi-mode decoding specified in WiMAX. Similar to the decoding schedule in [29], messages in a whole layer can be updated simultaneously. The 6-bit quantized message updating process and interconnection are split into 3 stages by applying 2 bit-width serial processing lines, so that the complicated interconnection and arithmetic can be avoided. Only 3 clock cycles are required to decode a whole layer instead of 6 as in [29], which results in 12~24 cycles to process one iteration and doubles the parallelism. Simply changing the bit-width of bit-serial processing schedule causes significant increase in decoder area and makes the originally complicated routing problem of the fully-parallel architecture more critical, which will further worsen the efficiency of this architecture. An early-detect-logic based message saturation process is proposed to remove the extra bit-width of permutation network (PN) in previous saturation process. It enables twice data transmission rate with the same complexity of PN as in [29], which effectively contributes to the improvement of energy-efficiency.

On the other hand, in the FEC-ARQ communication systems, the FER (frame error rate) performance is the key factor to the throughput. However, previous LDPC decoders merely paid attention to the error-correcting performance. An advanced dynamic quantization (ADQ) technique is proposed to enhance the error-correcting performance in layered decoder architecture. With 2% area overhead, 6-bit ADQ achieves BER performance close to 7-bit fixed quantization. It is noteworthy that the total gate count has minor increase compared with [29] even we introduce ADQ technique for better error-correcting performance, which is due to the optimized 3-state PU, early-detect-logic based message saturation process, efficient parity-check strategy and improved CN message storage method. The power synthesis report tells that the energy-efficiency is improved by 46.8% compared to the state-of-the-art work [29].

The rest of the chapter is organized as follows. Section 5.2 introduces the preliminaries of the decoder design, including the description of layered decoding algorithm with quantization adopted in this design and the message saturation problem in [29]. Section 5.3 introduces the proposed ADQ technique. Section 5.4 introduces the proposed fully-parallel layered LDPC decoder architecture in detail. The synthesis and comparison results are discussed in section 5.5.

5.2 Preliminaries

In this section, at first the layered decoding algorithm with quantization affect is described for hardware implementation. Secondly, the solution to the message saturation process in the previous bit-serial architecture is demonstrated. At last, the PU architecture of previous bit-serial layered decoder is shown for discussion and comparisons.

5.2.1 Layered Decoding Algorithm with Quantization

Before the introduction of proposed decoder design, the layered decoding algorithm with quantization adopted in the LDPC decoder is described first. Notice that the above quantization specifically refers to APP and VN message quantization instead of intrinsic message quantization, since the quantization of APP and VN messages majorly determines the implementation complexity of layered LDPC decoder architecture.

The layered algorithm applies different message updating schedule than TPMP algorithm. For each CN (or group of CN), the APP messages are updated after the CN message updating immediately in layered algorithm. The updated APP messages participate in the CNU of the following CN (or group of CN) within the iteration, which is different from the TPMP who cannot use such updated information until the next iteration. It is already proven that the more frequent APP message updating accelerates the convergent speed by at most twice. Nevertheless, the data dependency problem becomes serious in layered algorithm which limits the flexibility of its hardware implementations.

The layered algorithm with OMS approximation is adopted in this work and discussed throughout this chapter. As described in [29], the operations of each CN in OMS layered algorithm are focused on three equations as follows.

$$\lambda_{mn} = L_n - \Lambda_{mn}^{t-1} \tag{5.1}$$

$$\Lambda_{mn}^{t} = \max(\min_{N(m)\backslash n} \left|\lambda_{mn}\right| - \beta, 0) \prod_{N(m)\backslash n} \operatorname{sgn}(\lambda_{mn}) \tag{5.2}$$

$$L_n = \lambda_{mn} + \Lambda_{mn}^{t} \tag{5.3}$$

L_n represents the log-likelihood ratio (LLR) of APP message for the *n*th bit of the codeword. λ_{mn} and Λ_{mn} are the LLR of VN and CN messages, respectively. *N(m)\n* denotes the set of the neighboring VNs for CN *m* excluding VN *n*. β is the offset value while *t* is the current iteration number. The sgn() function in Eq. 2 is the sign function defined in mathematics to extract sign of real number. The sgn(x) returns ‘-1’ if x is negative, otherwise it returns ‘1’. We would like to name the process in equation (5.1) as “pre-VNU”, equation (5.2) as “CNU” and equation (5.3) as “post-VNU”, for convenience of explanation in the following scripts.

In most of the decoder implementations, all the messages are quantized. From pre- and post-VNU it can be easily proven that no matter which of the VN and APP messages has larger quantization range, the calculations are possible to get overflowed. VN and APP messages quantized in the same level and equations combined by saturation processes are the practical manner. For quantized messages, the algorithm is slightly changed. The pre- and post-VNU become equation (5.4) and equation (5.5) as follows, while β becomes 1 ulp (unit in the last place).

$$\lambda_{mn} = SAT(L_n - \Lambda_{mn}^{t-1}) \tag{5.4}$$

$$L_n = SAT(\lambda_{mn} + \Lambda_{mn}^{t}) \tag{5.5}$$

SAT() function is the saturation process as shown in equation (5.6) with (*q*, *f*) quantization, in which the *q* and *f* represent the total quantization bit size and fraction bit size, respectively.

$$y = SAT(x) = \begin{cases} 2^{q-f-1} - 2^{-f}, x > 2^{q-f-1} - 2^{-f} \\ -2^{q-f-1}, x < -2^{q-f-1} \\ x, others \end{cases} \tag{5.6}$$

Notice that although the saturation process can be simply applied in various bit-parallel decoders, it is one of the critical parts in the bit-serial architectures due to the differences in message transfer schedules.

5.2.2 Message Saturation Solution in Bit-serial Decoder Architecture

In bit-serial decoder architecture, the APP and VN messages are split into q bits and processed q clock cycles for pre- and post-VNU [29]. Generally consider that the messages are in 2's complement form and processed from least significant bit (LSB) to most significant bit (MSB). Fig.29 shows the diagrams of pre-VNU (a) and post-VNU (b) for one serial processing line in bit-serial architecture. The CNU process cannot get started until all the q bits of VN message are obtained, which forces temporary storage (q-bit DFF registers in Fig.29) for VN message. Since CNU is based on comparison tree and does not involve saturation process, it is constructed in bit-parallel style as that in [29].

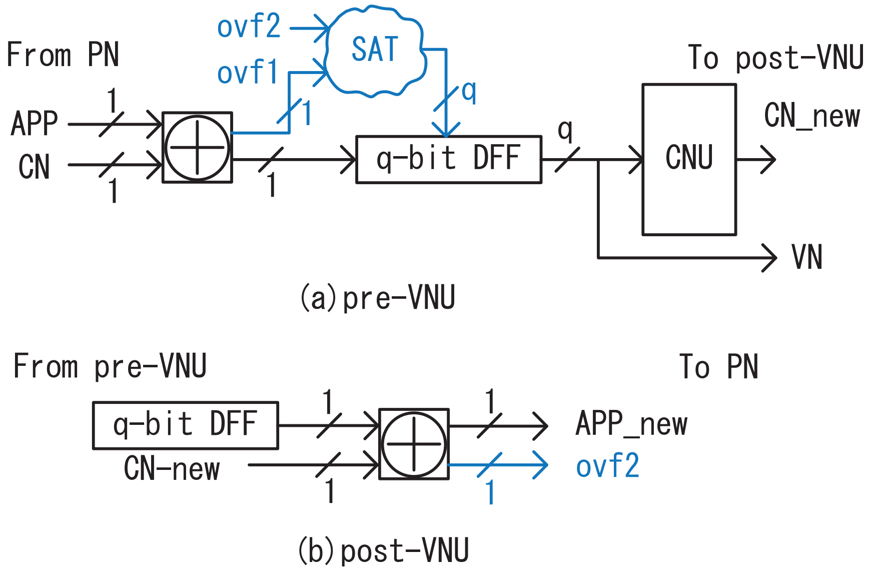


Fig.29. Diagrams of VNU in bit-serial architecture.

The bit-serial arithmetic units in pre- and post-VNU output serial bits of the sum to VN DFF registers and through permutation network (PN) to the input port of next pre-VNU, respectively. Saturation process needs to be implemented to avoid error arithmetic results for current post-VNU and next pre-VNU if the current pre-VNU and post-VNU gets overflowed, respectively. The previous solution in [29] is shown in Fig.29 with extra circuits marked in blue. It is unable to correct previously transmitted bits during the post-VNU process since they are already being processed in the next pre-VNU when the saturation is detected. Therefore, an overflow signal ‘ovf2’ needs to be sent through PN together with the MSB of new APP message to inform pre-VNU that the previously transmitted bits are not exactly correct. For the pre-VNU, the saturation process can be done by checking the overflow signal ‘ovf1’ which is generated in the MSB arithmetic cycle. The saturation process of current post-VNU can be done by checking both ‘ovf1’ and ‘ovf2’ signals in MSB arithmetic cycle of next pre-VNU. When the overflow is detected, maximum or minimum value of VN message will overwrite the message in VN DFF registers. The conditional logic of saturation process implemented in [29] is shown in Fig.30. It can be proven that this logic can get exactly the same saturation results as the bit-parallel solution. The demerit of this solution is that extra 1 bit-width of PN is required for the interconnection, in which the output bits are valid for only one sixth of the processing time.

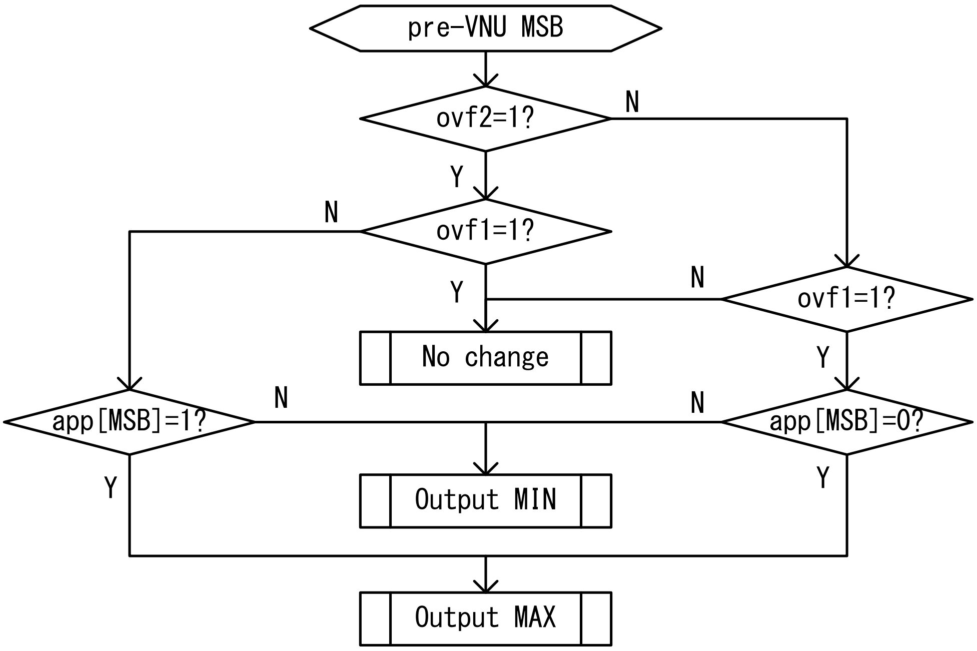


Fig.30. Previous solution to saturation process in bit-serial decoder.

5.2.3 Bit-Serial Layered LDPC Decoder Architecture

Fig.31 shows the structure of the processing unit (PU) and the timing diagram of decoding process of bit-serial decoder architecture in [29]. 24 APP/VN-REG-bit-updaters are responsible for bypassing APP messages or calculating VN messages, one bit per clock cycle, from LSB to MSB. In the similar way, 24 APP-bit-updaters calculate the new APP messages according to VN and CN messages (or the bypassed APP messages), as shown in the detail structure in the middle of Fig.31. PN A and PN B are duplicated networks that PN A is responsible for permuting updated APP message bits at each cycle while PN B is just for permuting the overflow signals which are generated during bit-serial additions at the MSB calculating cycles. The two pipelines for updating the VN and APP messages work without any idle clock cycle, so that they can be fully overlapped to make the PU work efficiently.

Another contribution of the bit-serial decoder architecture is the min-sorter (min-finder) structure. As mentioned in chapter 2, the Min-sum algorithm requires the first and second minimum values of VN magnitudes. Since directly calculating the two values in one clock cycle will lead to longer critical path and larger circuit area, the min-sorter module is designed to calculate the two minimum values in two successive clock cycles. It can guarantee that the CN messages, which are generated using the two minimum values is ready when the updating of LSB in APP message is processing, as shown in the timing diagram of Fig.31.

The min-finder functional module contains two separate 4-stage comparator trees as shown in Fig.31, which find the minimum values of the upper and lower 12 input data independently. Between the comparator trees and the output, a simple selector is implemented to either choose the smaller one for the minimum value in the normal decoding mode or directly pass through the two minimum values in the two-layer concurrent processing mode. [29]

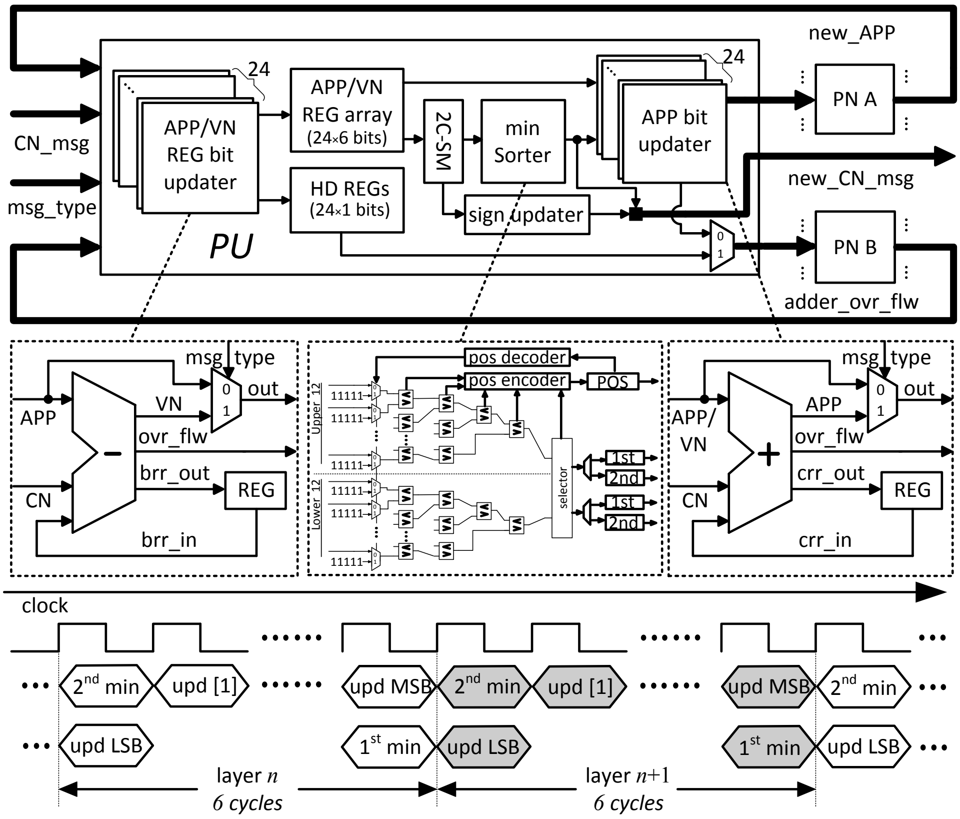


Fig.31. Processing unit structure and timing diagram of bit-serial layered decoder [29].

5.3 Advanced Dynamic Quantization Technique

Along with the iterative decoding, reliabilities of VNs or magnitudes of APP messages increase. Saturation of APP messages during post-VNU process occurs more and more frequently. Fixed quantization scheme for all the messages limits the reliabilities of high reliability VNs to further increasing. As a result, the error-correcting capability of decoder is gradually decreased. To address this issue, the adaptive quantization technique is proposed in [40]. Based on some criterions predefined, the message quantization can be changed to enhance error-correcting performance while maintaining the total quantization bit size. However, the criterions are complicated to be implemented and not suitable for layered algorithm. In this section, a newly proposed advanced dynamic quantization (ADQ) technique for solving the saturation problem to enhance the error-correcting performance in the proposed fully-parallel layered decoder is discussed.

5.3.1 Saturation Effects in Layered LDPC Decoder

In layered decoding algorithm with quantization as discussed in section 5.2.1, for a whole iteration the number of APP message updating is several times larger than in TPMP algorithm. The saturation error of APP message updating in each CN process affects the following CN process immediately. A simple model is demonstrated to explain the bad situation as shown in Fig.32. The variable node v1 connects to four CNs named as cn1, cn2, cn3 and cn4 in the PCM. The signs on the edges represent the increase or decrease of reliability of v1 after the corresponding CN processes. Fig.32 (a) shows the TPMP algorithm case. Suppose that v1 has high, positive reliability (i.e. large APP value) and the accumulation of the four changes is also positive. The result is that the new APP for v1 will become larger and saturation may occur which would limit the VN messages in the next iteration. In the layered algorithm, the iteration is divided into 4 layers as shown in Fig.32 (b). The APP message is updated in every layer. The worst case is that neither of cn1 nor cn2 contributes to the APP of v1 because of the originally saturated APP. In this case, cn3 and cn4 reduce the APP of v1, which results in decreased reliability. If such kind of situation appears in large numbers, the decoding will fail to converge. That is the main reason why layered algorithm performs even worse than TPMP algorithm in small quantization level case.

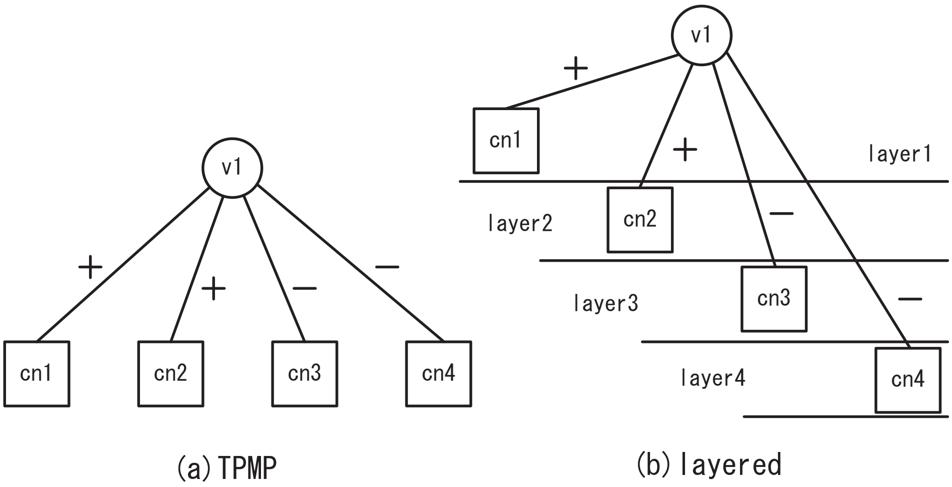


Fig.32. Saturation effects in TPMP (a) and layered algorithm (b).

5.3.2 Advanced Dynamic Quantization

In order to avoid the situation mentioned above, dynamically widening the quantization range while maintain the quantization bit size when a large number of VNs have high reliability is a preferable solution. Taking the hardware implementation into consideration, an efficient criterion to trigger the quantization change is proposed for layered algorithm.

The CNU function in equation (5.2) is the core function in layered algorithm, which contributes all factors (i.e. CN messages) to the convergence. Equation (5.2) contains two parts which are the minimum and the product of signs. The product part is much critical and considered as the indicator for triggering quantization change. Since it is different from the parity-check constraint which is the product of signs of APP messages, we name it as Pseudo-Satisfied (PS) signal. The PS signal of CN m is defined as follows.

$$PS_m = \prod_{N(m)} \operatorname{sgn}(\lambda_{mn})$$

It is simply assumed that there is at most one sign which is different from the correct decision in each CN since such situation majorly occurs after several iterations. The following results can be deduced that if *PS* is positive, the CN tends to increase the reliabilities of all related VNs (i.e. L_n); if *PS* is negative, the CN tends to reduce the reliability or flip the message sign of VN which has the different sign at the expense of decreasing the reliabilities of other VNs.

In order to deal with the decoding failure caused by saturation of APP messages in the layered decoding with dynamic quantization, *PS* signals are considered as the better choice for triggering quantization change than parity-check constraint because they are more related to APP message saturation while the parity-check constraint is more related to decoding convergence. Simulations also show that dynamic quantization with *PS* signals can achieve slightly better error-correcting performance and smaller average iteration number with optimized parameters for both.

Furthermore, since *PS* signals are already calculated in CNU function, the hardware overhead is the smallest. In the case of applying parity-check constraint for the trigger signal, extra 4.9k gates are required for the proposed decoder. The trigger criterion with *PS* for ADQ technique is named as pseudo-unsatisfied criterion and the algorithm description of ADQ is listed as follows.

```
1) Pre-define parameters: $l$, $PS\text{-}TH$
2) Initialize registers: for each $i$: 0 to $l$-1 CNT[$i$] = 0, SUM=0, change_flag=0
3) Iterative layered decoding
for each layer $k$
    Process common layered decoding algorithm
    CNT[$i$+1]=CNT[$i$], (for each $i$:$l$-2 to 0)
    CNT[0]=0
    for each row $m$ in layer $k$
        $PS_m = \prod_{N(m)} \mathrm{sgn}(\lambda_{mn})$
        if $PS_m$=-1
            CNT[0]=CNT[0]+1
    SUM=CNT[0]+CNT[1]+…+CNT[$l$-1]
    if SUM<PS-TH && $k$>$l$ && change_flag=0
        change_flag=1
        Go to step 4)
4) Quantization change
for each column $n$
    $L_n$=$L_n$/2
    for each row $m$
        $\Lambda_{mn}$=$\Lambda_{mn}$/2
Back to step 3)
```

For successive *l* layers, if the total number of CNs with *PS* = -1 is less than *PS-TH*, the quantization change will be triggered. The *l* and *PS-TH* are simulation-based values. The quantization change is processed at most once by dividing all APP and CN messages by 2 from the next layer. In hardware implementation, it is done by right-shifting 1-bit for all the APP and CN messages.

Fig.33 shows the BER vs. SNR simulation results for WiMAX code-rate 1/2 (2304, 1152) and code-rate 5/6 (2304, 1920) codes with different quantization schemes including (6,1) fixed quantization, (7,1) fixed quantization and (6,1) ADQ. OMS layered algorithm with the iteration number set to 10 is used throughout all the simulations. Parameter *l* is set to 8 (4) for code-rate 1/2 (5/6) with *PS-TH* set to 30

(60). These simulation-based parameters are insensitive to channel SNR.

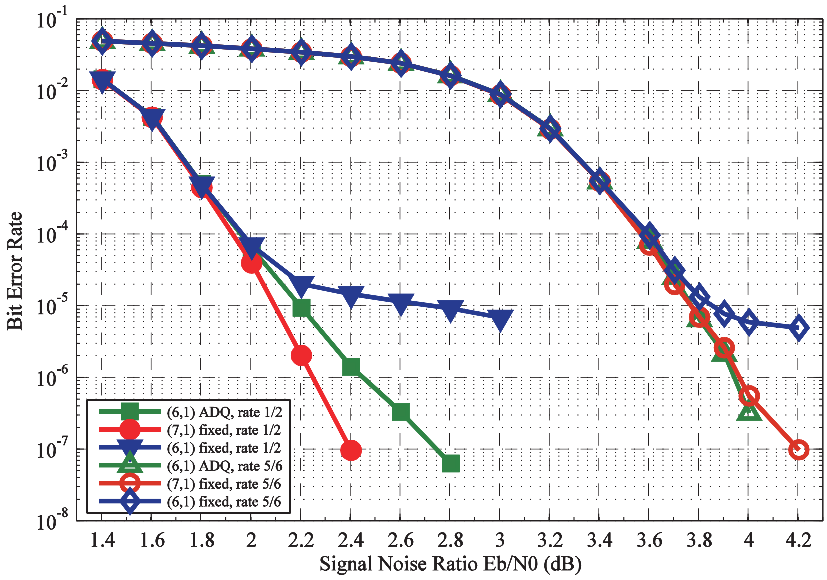


Fig.33. BER performance for WiMAX code-rate 1/2 and 5/6 codes using OMS layered algorithm, max 10 iterations, with (6,1) ADQ, (7,1) fixed and (6,1) fixed quantization schemes.

From Fig.33 it can be observed that simulations with (6,1) ADQ preform much better than (6,1) fixed quantization. It shows less than 0.1dB performance loss in SNR compared to (7,1) fixed quantization at BER = 10^{-5} without error floor detected as low as BER = 10^{-7} for code-rate 1/2 and almost the same performance as (7,1) fixed quantization for code-rate 5/6. In contrast, the simulations with (6,1) fixed quantization perform 0.6dB and 0.1dB worse than (7,1) fixed quantization at BER = 10^{-5} for code-rates 1/2 and 5/6 and suffer from obvious error floor around BER = 10^{-5}.

The proposed ADQ technique takes full advantage of the layered algorithm. During the rapid convergence of layered decoding, it can quickly determine the decoding progress and then process the quantization change immediately, with very limited complexity overhead. Compared to previous method in [40] for TPMP algorithm, the timeliness of determining and processing of quantization change is improved, which enables better error-correcting performance. Hardware implementation of ADQ technique for the proposed decoder is shown in section 4 in detail.

5.4 Fully-Parallel Layered LDPC Decoder

In this section, we first demonstrate the top-level architecture of proposed fully-parallel layered LDPC decoder core in section 5.4.1. The decoder is especially designed for WiMAX standard LDPC codes with messages quantized with 6-bit. Secondly, the 3-state processing unit (PU) module is discussed in section 5.4.2 in detail. Finally, the early-detect-logic based saturation process and hardware implementation of ADQ technique are described in section 5.4.3 and 5.4.4, respectively.

5.4.1 Proposed Decoder Architecture

Fig.34 shows the block diagram of the proposed fully-parallel layered LDPC decoder core architecture. It is made up of a central controller module, 3-state processing unit (PU) array, 96 blocks of register based CN message memory and a permutation network (PN) array with two switch arrays (SA). A 4608 bit-width 2-to-1 MUX is responsible for the transfer of intrinsic messages into the decoding modules. Input and output buffers are not integrated in the decoder core.

The central controller module plays an important role in the decoder core, which not only deals with the input of intrinsic messages into decoding modules and the output of 2304 hard-decision bits, but also controls all of states of decoding including 3-state fully-parallel layered scheduling, hard-decision, and trigger signal generating for ADQ technique. In the proposed fully-parallel layered decoder, a number of 2304 6-bit quantized APP messages are transferred and processed through the decoding modules, which are split into 3 stages. For each stage totally 2304x2 bits are processed within one clock cycle. Since not all the 2304 entries are involved in each CNU processing, the central controller is responsible to tell PUs which entries should be processed and which should be bypassed. When the decoding modules finish all the iterations, the parity-check functions are also processed with the same decoding modules and controlled by the central controller. Compared with [29] which uses extra clock cycles for initialization and extra circuits for processing parity-check function, the initialization stage is merged into decoding stages and parity-check circuits are not necessary, so that it improves the hardware utilization and reduces the latency for continuous decoding.

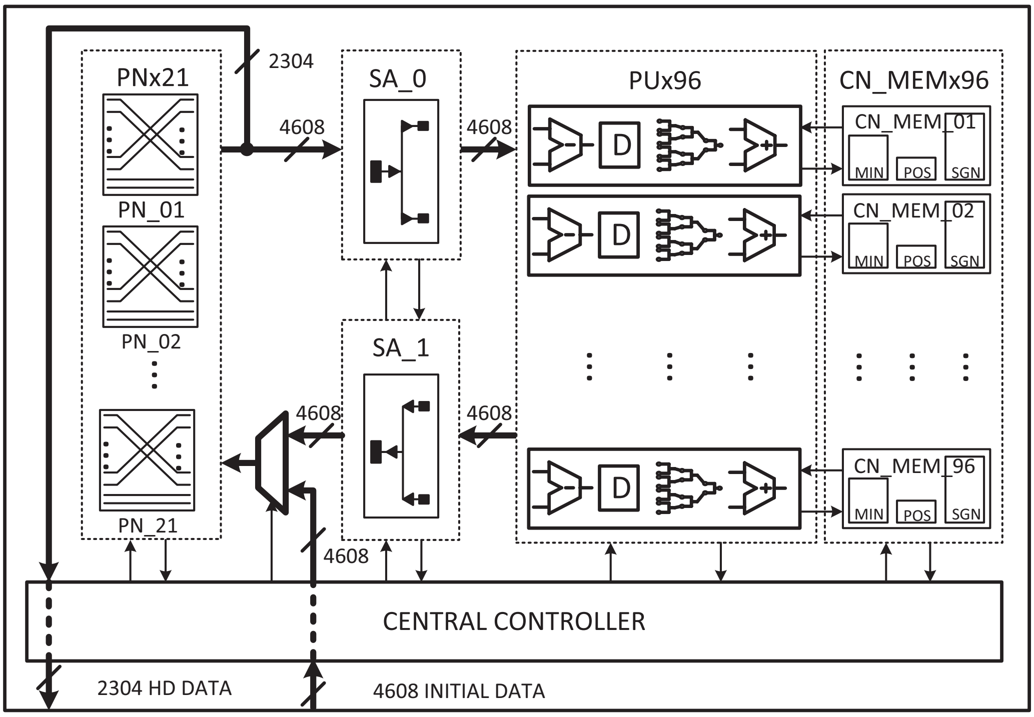


Fig.34. Block diagram of the proposed fully-parallel layered LDPC decoder architecture.

For APP message transfer between successive layers, the PN array is implemented. It contains a number of 21 PN instances which is constructed with multiple stages of MUXs as that in [29] of 2 bit-width. It achieves all the possible combinations for barrel-shift operations with sub-block size not larger than 96, which can fully support the multi-length requirement of WiMAX standard codes. A number of two 2 bit-width cross-bar switch arrays SA_0 and SA_1 are inserted between PN array and PU array to separate 24 blocks of APP messages into two groups for the two-layer concurrent processing proposed in [29], which can reduce clock cycles per iteration and also the sizes of CN message memory for two kinds of code-rates specified in WiMAX.

The CN message memory is separated into 96 blocks connected with 96 PUs, which is composed of registers. The compressed message storage method for CN messages is adopted, in which for one CN only the minimum and second minimum magnitude (MIN), the position of minimum magnitude (POS) and all the 24 sign bits (SGN) are stored instead of memorizing all the CN messages [29]. Furthermore, in this design the storage of CN messages for the two-layer concurrent processing is further split to reduce the depths of MIN and POS parts in CN memory as shown in Fig.35. Finally, the total CN memory size can be reduced from 36.9k bits to 33.4k bits, which leads to 9.4% reduction compared to [29] for code-rate 2/3A which utilizes the most of CN memory and decides the overall CN memory size of decoder.

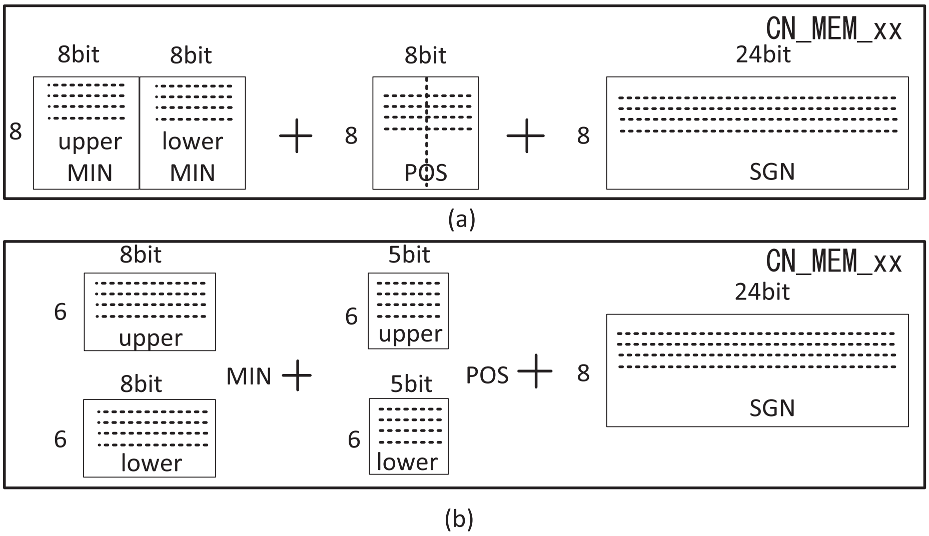


Fig.35. CN message storage method, (a) in [29], (b) in this paper.

5.4.2 3-State Processing Unit (PU)

Fig.36 shows the structure of the 3-state PU including pre-VNU, post-VNU and CNU designs. In each PU, there are 24 entries of 2 bit-width serial processing lines to transfer APP messages, 48 bit-width input for compressed CN messages fetched from CN memory and two 24 bit-width enable signals ‘pre-EN’ and ‘pst-EN’ for instructing the pre-VNU and post-VNU manners. An enable signal set to 1 lets the corresponding VN participate in the message updating of current CN, otherwise PU stores and bypasses the APP message of the corresponding VN for the CN process of next layer.

The CN_RECOVER unit is responsible to recover the compressed CN messages and convert them into serial 2 bit-width form for pre-VNU process. The CNU process is shown in the blue dotted box in Fig.36 (a), in which the 2's complement to signed-magnitude (2C-SM) conversion and comparison trees for finding minimum (i.e. Min_finder) are in bit-parallel styles. The minimum and second minimum magnitudes are found in two clock cycles to reduce critical path. In order to achieve fully overlapped decoding schedule, there must be two bypass routes from pre-VNU to Min_finder and from Min_finder to post-VNU which are not marked in Fig.36 (a), so that two Min_finder units (Min_finder0/1) are implemented to avoid extremely long critical path.

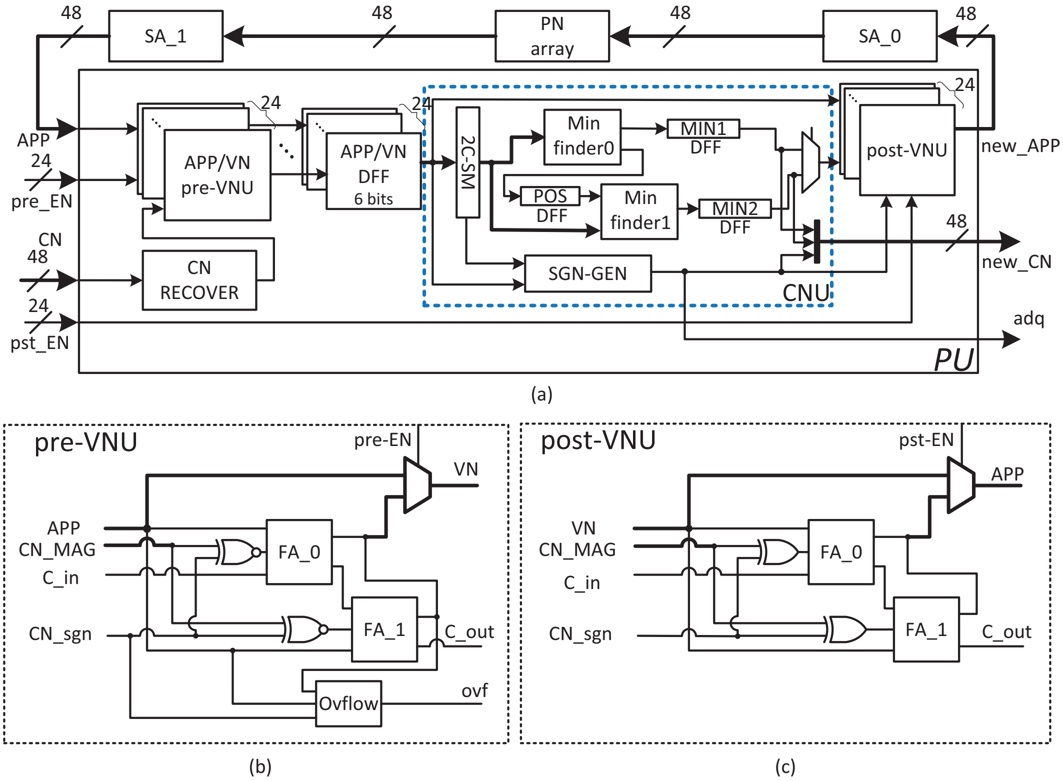


Fig.36. Structures of 3-state processing unit (a), pre-VNU (b), post-VNU (c).

Fig.36 (b) and (c) demonstrate the pre- and post-VNU circuits, respectively. 2 bit-width 2-to-1 MUXs deal with the update or bypass operations for APP or VN messages as described above. The CN messages are inputted with signed-magnitude form. 2 exclusive-NOR or exclusive-OR gates for pre- or post-VNU circuit deal with different CN message signs to process both addition and subtraction with the same full-adders (FA). Such kind of structure avoids the conversion of CN messages from sign-magnitude to 2’s compliment form, thus to reduce circuit area and critical path.

The PU modules only have three operating states: S0, S1 and S2, which are controlled by the central controller. In different states the PUs process different tasks, as shown in Fig.37. The least significant 2 bits, the middle 2 bits and the most significant 2 bits are processed at states S0, S1 and S2, respectively. The blue bits in the VN DFF registers are generated from pre-VNU of current layer and the black bits are from those of previous layer. Two bypass routes marked with red lines pass data which have not yet stored in VN DFF registers and registers for storing the second minimum. By utilizing the bypass routes, pre-VNU of current layer and post-VNU for updating APP messages of previous layer can be fully overlapped.

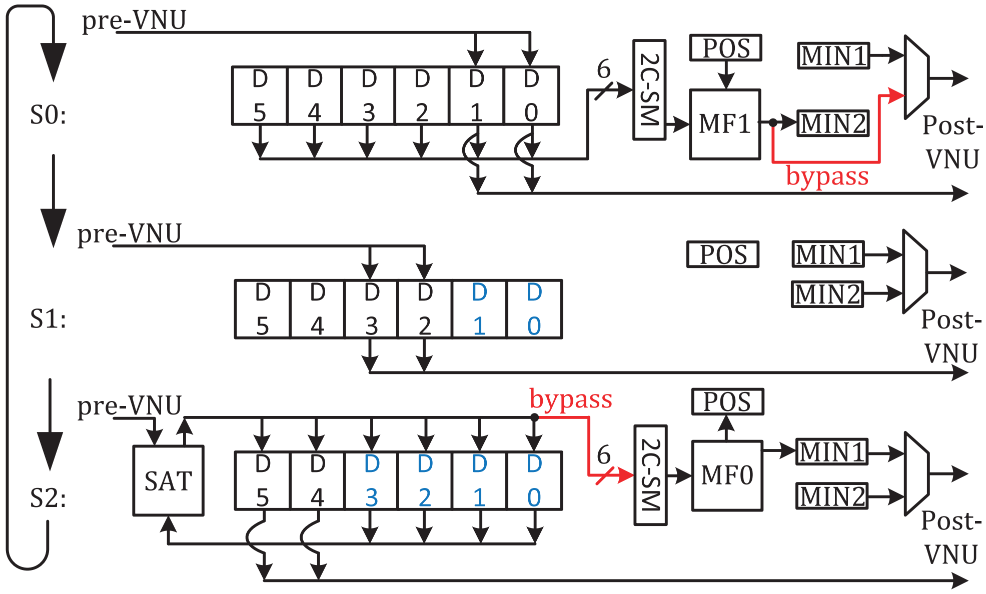


Fig.37. Processing tasks in different states.

Furthermore, the parity-check function can be processed by setting all the CN messages to zero and freezing the state of PU at S2, which is also controlled by the central controller. With this strategy, 4~8 clock cycles are enough to finish all the parity-check functions for different code-rates in WiMAX. The overhead of overall decoding time is only 1/30, which is 1/3 of that in partial-parallel layered decoder [26].

5.4.3 Early-detect-logic Based Saturation Process for Post-VNU

In section 5.2.2 we demonstrate the previous solution for saturation process for pre- and post-VNU. Although it is functionally precise, the extra bit-width requirement for the interconnection worsens the efficiency of the decoder architecture. Adopting the previous solution in this work will lead to 33% parallelism degradation.

By utilizing the properties of 3-state PU and the limited range of CN messages, early-detect-logic based saturation process for post-VNU is proposed and implemented in this work for the fully-parallel decoder architecture. The saturation during post-VNU process can be detected by the early-detect-logic at state S1, instead of S2 for MSB updating, as shown in equation (5.7) and equation (5.8) below.

$$EDP = \overline{VN[5]} \ \& \ VN[4] \ \& \ \overline{Sgn} \ \& \ Carry \tag{5.7}$$

$$EDN = VN[5] \ \& \ \overline{VN[4]} \ \& \ Sgn \ \& \ \overline{Carry} \tag{5.8}$$

EDP and *EDN* are the positive and negative saturation flags, respectively. *VN*[5:0] is the VN message and *Sgn* is the sign bit of CN message generated from the CNU process. *Carry* is the carry bit generated from post-VNU at state S1. Since the magnitude of CN message is limited by 4 bit-width for OMS layered algorithm, it can be guaranteed that if both *EDP* and *EDN* are zero, the final APP message cannot be saturated.

When either of saturations is detected by the early-detect-logic at state S1, the two output bits *APP*[3:2] will be set to ‘11’ and ‘00’ according to the *EDP* and *EDN* flag, respectively. For the next clock cycle (i.e. state S2), the two output bits *APP*[5:4] will be set to ‘01’ or ‘10’, respectively. Since we will not correct the 2-bit APP messages *APP*[1:0] transferred at state S0, when saturation is detected the updated APP message has saturation error within 0~3 ulp. In order to evaluate the error-correcting performance by introducing the early-detect-logic based saturation process, simulations have been done for both (6,1) fixed quantization and (6,1) ADQ schemes. Instead of the BER performance, the FER (frame error rate) performance for decoding WiMAX LDPC codes is observed in the simulations since the error correction strategy of WiMAX system adopts ARQ (automatic repeat-request).

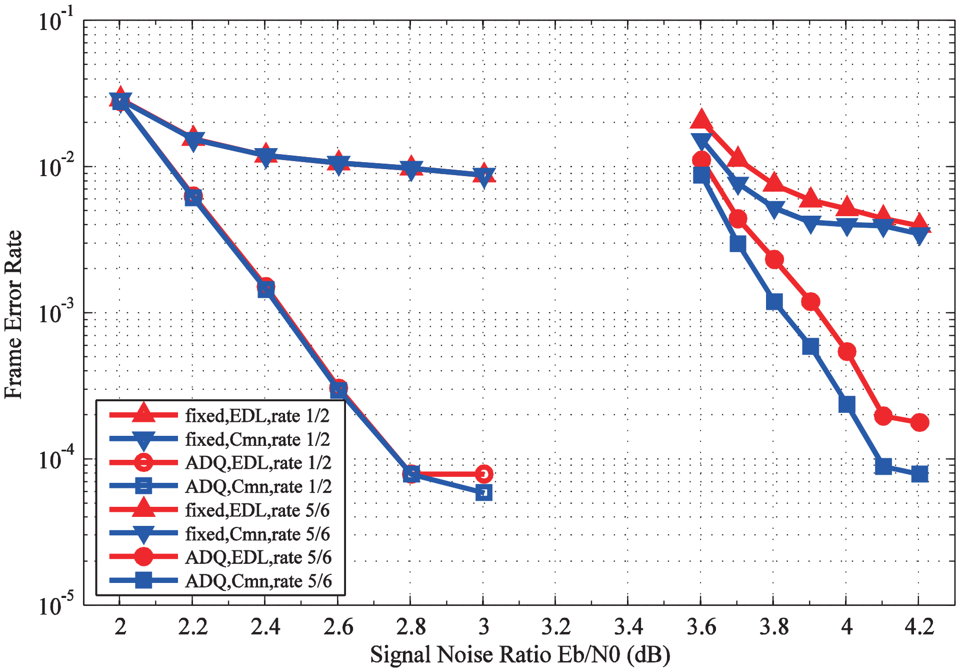


Fig.38. FER performance for early-detect-logic and common saturation processes with fixed quantization and ADQ schemes for WiMAX code-rate 1/2 and 5/6 codes.

The FER vs. SNR simulation results for WiMAX code-rate 1/2 (2304, 1152) and code-rate 5/6 (2304, 1920) codes with different quantization schemes including (6,1) fixed quantization and (6,1) ADQ are shown in Fig.38. “EDL” denotes the early-detect-logic based saturation process and “Cmn” denotes the common saturation process in this figure. OMS layered algorithm with the iteration number set to 10 is used throughout the simulations. Parameter *l* is set to 12 (4) for code-rate 1/2 (5/6) with *PS-TH* set to 60 (60) in ADQ, which can achieve optimized FER performance.

From Fig.38, it can be found that (6,1) ADQ achieves significant improvement in FER performance compared to (6,1) fixed quantization as well as BER performance which is already shown in section 5.3. Simulation results also show that for code-rate 1/2 code, there is almost no difference in FER performance between the two saturation processes for both fixed quantization and ADQ schemes. On the other hand, for code-rate 5/6 code, FER performance with early-detect-logic saturation process is slightly worse than that with the common saturation process for both schemes. The reason is that the larger magnitudes of intrinsic messages at higher SNR for code-rate

5/6 lead to more frequently APP message saturation than code-rate 1/2, so that the saturation errors of early-detect-logic saturation process influence the decoding performance more. Nevertheless, compared with the significant improvement due to ADQ, the small performance loss caused by early-detect-logic saturation process can be neglected.

By applying the early-detect-logic based message saturation process, 2 bit-width PN can support the APP message transmission in 3 clock cycles to meet the processing schedule of the proposed 3-state PU. It enables twice data transmission rate with the same complexity of PN as in [29], which effectively contributes to the improvement of energy efficiency. Compared with the previous saturation process, the proposed one saves 1 bit-width PN which is 61.7k gates, or 1 clock cycle for processing each layer which leads to 25% improvement in parallelism.

5.4.4 Implementation of ADQ Technique

The pseudo-unsatisfied criterion of ADQ is already defined in section 5.3. For hardware implementation, the product function for all the signs is based on a series of exclusive-OR operations, and is already obtained by SGN_GEN unit which is responsible for generating new sign bits of CN messages. No extra logic is required in each PU except 1-bit OR gate for the two-layer concurrent processing mode which is applied to determine at least one pseudo-unsatisfied CN for two-layer concurrent processing mode. The trigger signal is generated by a combinational summation circuit for a number of 96 1 bit-width ‘adq’ signals as the wire pulled in Fig.36, and also four 7 bit-width registers in the central controller module. The total gate count is less than 1k.

On the other hand, the execution mechanism for quantization change of APP messages is processed in PU modules by changing the DFF storage positions after pre-VNU, as shown in Fig.39. Quantization change for CN messages is much simpler that a number of 96 16 bit-width 2-to-1 MUXs are implemented in controller module to selectively right-shift 1-bit for the four minimum values, which requires 3.5k gates.

It is estimated that the overall gate count increase is about 20k, which occupies about 2% of overall decoder area.

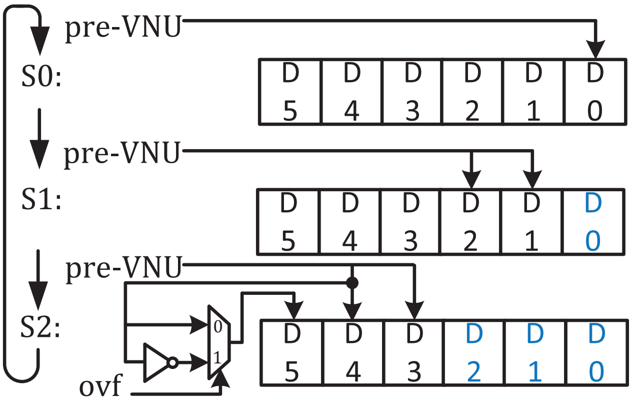


Fig.39. Execution mechanism for quantization change of APP messages.

5.5 Implementation and Comparison Results

This design is synthesized in SMIC 65nm low leakage LVT CMOS technology with Synopsys Design Compiler (DC) version F-2011.09. The maximum clock frequency is reported as 122MHz by the tightest area constraint which results in 978k equivalent gates. The data throughput *Tpt* is calculated by equation (5.9) as shown below.

$$Tpt = \frac{L \times Frequency}{N_{cycle} \times N_{iter} + N_p} \tag{5.9}$$

Where *L* is the code-length, N_{cycle} is the number of clock cycle per iteration, N_{iter} is the iteration number and N_p is the number of clock cycle for parity-check. For the proposed decoder, code (2304, 1920) achieves the highest data throughput, in which case N_p is 4 and N_{cycle} is 12.By using 55MHz clock frequency and 10 iterations for calculation, the decoder achieves data throughput of 1021Mbps, which takes hard-decision and initialization into account. DC power analysis report shows that with 1.2V supply voltage and 55MHz clock frequency the power consumption of decoder is 59.5mW. The energy efficiency *EE* can be calculated by equation (5.10) as shown below.

$$EE = \frac{P}{Tpt \times N_{iter}} \tag{5.10}$$

P denotes the power consumption of decoder. The energy-efficiency of the proposed decoder is calculated to be 5.83pJ/bit/iteration.

Table 5 Comparison with previous WiMAX decoders.

	In this work	[26]	[29]
Technology	SMIC65nm	SMIC0.13μm	SMIC65nm
Supply voltage	1.2V	1.2V	1.2V
Code-length	576~2304		
Code-rate	1/2, 2/3A, 2/3B, 3/4A, 3/4B, 5/6		
Iteration number	10	10	10
Clock cycle# per iteration	12~24	48~54	24~48
Logic gate count	645k	470k	597k
Memory (register)	47.2kb	72.5kb	56.4kb
Equivalent gate count	978k	946k	968k
Frequency	55MHz	214MHz	110MHz
Data throughput	1021Mbps	955Mbps	1056Mbps
Power consumption	**59.5mW	397mW	115mW
*Normalized power	59.5mW	199mW	115mW
Norm. energy-efficiency	5.83pJ/bit/iter	20.8pJ/bit/iter	10.9pJ/bit/iter

*All the powers have been normalized to 65nm process.

**Power in this work is synthesis result from Synopsys Design Compiler.

Table 5 summarizes the implementation results of this design and also the comparisons with the state-of-the-art partial-parallel layered LDPC decoder [26] and the state-of-the-art WiMAX LDPC decoder [29]. The proposed architecture achieves up to 4.3 times parallelism compared to [26] and twice parallelism compared to [29] with 32k and 10k increase in gate count, respectively. The synthesized power consumption is reasonable since the proposed decoder has similar composition with [29] which is fabricated with the same 65nm SMIC process and measured. In order to compare the energy efficiency with [26] which is fabricated in SMIC 0.13μm process, the power consumption of [26] is normalized to 65nm process. The normalizing ratio of power consumption is obtained by considering both synthesis results and theoretical deduction. We have synthesized our designs in both SMIC 65nm and 0.13μm libraries with the tightest area constraint to generate the circuit composition as similar as possible. The power synthesis results show that the normalizing ratio from 0.13μm and 65nm process is around 2. On the other hand, considering the calculation of dynamic power in synthesis tool $P_d \propto CV^2 f_c$, dynamic power P_d is proportional to the load capacitance C which is proportional to the scale of cells. If the circuit structures, supply voltages V and clock frequencies f_c are the same, the dynamic power from 0.13μm to 65nm design becomes half. Therefore, we set the power normalizing ratio from 0.13μm to 65nm as 2 for comparison. Comparison results show that proposed decoder achieves 46.8% and 72.1% improvement of normalized energy efficiency compared to [29] and [26], respectively.

Although the arithmetic units for pre- and post-VNU are enlarged from 1 bit-width to 2 bit-width which also leads to bit-width increase for related MUX circuits, the overall PU area is not increased significantly even the ADQ technique is also implemented. The main reason is due to the reduction of the processing states in PU. The number of processing states in previous PU is designed over 8, which leads to 4-bit state control signal. On each of the 2304 bit-serial message updating datapath, at some ports such as the input of APP/VN register, the inputs of post-VNU and the output of APP message, it requires 1 bit-width 6-to-1 MUX in previous PU or 2 bit-width 3-to-1 MUX in proposed PU in order to select corresponding bits from bit-parallel messages at different states. Synthesis result shows that the gate count reduction is about 30% from 4-bit controlled, 1 bit-width 6-to-1 MUX to 2 bit-width 3-to-1 MUX. Moreover, the gate count increase of arithmetic units of VNU is just 40k which is not large compared with the reduction from MUX, so that the overall PU area is not increased significantly.

5.6 Summary

In this paper, a synthesizable, 3-state fully-parallel layered LDPC decoder IP core for WiMAX is presented. By applying the 3-state fully-parallel processing architecture, extremely high decoding parallelism is achieved, which enables 12~24 clock cycles per iteration and doubles the parallelism of our previously proposed bit-serial decoder [29] with 1% gate count increase. Power synthesis result shows the proposed decoder achieves energy-efficiency as 5.83pJ/bit/iteration, which obtains

46.8% and 72.1% improvement compared to the state-of-the-art WiMAX LDPC decoder [29] and partial-parallel layered LDPC decoder [26], respectively. Moreover, an advanced dynamic quantization (ADQ) technique is proposed to enhance the error-correcting performance in layered LDPC decoder. Simulation results show that by applying the efficient pseudo-unsatisfied criterion, decoding with 6-bit dynamic quantization can perform close to 7-bit fixed quantization with improved error floor performance. Furthermore, for the FEC-ARQ communication systems, the FER performance is improved around 2 orders, which can bring more than 5% increase in transmission throughput for the whole systems.

6. Conclusion

In this dissertation, we focus on the design methodology for multi-mode LDPC decoders. The Layered decoding algorithm is proven to be the better candidate for multi-mode implementation and applied in all the decoder designs in this dissertation. For LDPC codes defined with QC-LDPC structure and IRA-LDPC structure, we provide different decoder architectures to achieve better error-correction performance or energy-efficiency. Contributions are given in both error-correcting performance and energy-efficiency of multi-mode LDPC decoders which are listed as follows.

1) For generic layered LDPC decoders, BER performance-aware early termination scheme is proposed to reduce power consumption while it can maintain the error-correcting performance. For applications which require exact decoding successful status, the proposed LSC-ET scheme is proven to be the best solution whatever in error-correcting performance, termination speed and hardware complexity. Compared to the second best ET scheme CSC-ET_1, the proposed scheme achieves 0.2 dB improvement in coding gain and 5%~10% reduction in the decoding time for WiMAX 1/2 rate codes, and 0.1 dB improvement in coding gain and 2%~5% reduction in the decoding time for DVB-S2 1/2 rate codes.

2) For DVB-T2 LDPC decoder, a novel message updating conflict resolution is proposed to deal with the performance loss caused by cutting-edge problem in layered decoder implementation. Compared to state-of-the-art works, the proposed decoder architecture can guarantee conflict-free, pure layered decoding performance with efficient overlapped pipeline decoding schedule which enables at most 1.2% redundancy in decoding time. Simulation result shows about 1/3 bit error reduction under the same condition and the redundant arithmetic units for TPMP decoding can be avoided.

3) For WiMAX LDPC decoder, a synthesizable, 3-state fully-parallel layered decoder IP is proposed to increase the decoding parallelism. Power synthesis result shows the proposed decoder achieves energy-efficiency as 5.83pJ/bit/iteration under

65nm CMOS process, which obtains 46.8% and 72.1% improvement compared to the state-of-the-art WiMAX LDPC decoder and partial-parallel layered LDPC decoder, respectively. Moreover, an advanced dynamic quantization (ADQ) technique is proposed to enhance the error-correcting performance in layered LDPC decoder. Simulation results show that decoding with 6-bit dynamic quantization can perform close to 7-bit fixed quantization with improved error floor performance. For the FEC-ARQ communication systems, the FER performance is improved around 2 orders, which can bring more than 5% increase in transmission throughput for the whole systems.

For the future work, researches will be deeply continued especially on quantization and termination problems for various LDPC decoder architectures and applications. The LDPC decoder architectures largely depend on the message quantization schemes which decide not only the decoder areas but also the decoding schedules. On the other hand, early termination schemes shows great potentials for power saving and latency reduction. However, the researches about them are not completed yet. For different applications and decoder architectures the probability and convenience of implementation is still unknown. Systematic and theoretic researches are considered to optimize the current and future LDPC decoders on both error-correcting performance and energy-efficiency.

Reference

[1] S. Lin and D. J. Costello, Jr.: Error Control Coding. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2004.

[2] R. Gallager, “Low-density parity-check codes,” Cambridge, MIT Press, MA, 1963.

[3] D.J.C. MacKay, “Good codes based on very sparse matrices,” IEEE Trans. Inf. Theory, vol. 45, no. 3, pp. 399-431, Mar. 1999.

[4] IEEE 802.11n. Wireless LAN medium access control and physical layer specifications, P802.11n/D3.07, 2008.

[5] IEEE 802.16e. Air interface for fixed and mobile broadband wireless access systems, P802.16e/D12, 2005.

[6] IEEE 802.15.3c. Wireless medium access control (MAC) and physical layer specifications for high rate wireless personal area networks (WPANs), 2009.

[7] ETSI, digital video broadcasting (DVB); second generation framing structure for broadband satellite applications. http://www.dvb.org.

[8] C. Berrou, A. Glaviex and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo codes,” Proc. IEEE Intl. Conf. on Communications, (ICC 93), pp. 1064-1070, Geneva, Switzerland, May 1993.

[9] A. Hocquenghem, “Codes corecteurs d’erreurs,” Chiffres, 2, pp. 147-156, 1959.

[10] 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.

[11] 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.

[12] X. Zhao, Z. Chen, X. Peng, D. Zhou and S. Goto, “A BER performance-aware early termination scheme for layered LDPC decoder,” 2010 IEEE Workshop on Signal Processing Systems (SiPS 2010), pp. 416-419, Oct. 6th, 2010.

[13] X. Zhao, Z. Chen, X. Peng, D. Zhou and S. Goto, “DVB-T2 LDPC decoder with perfect conflict resolution,” IPSJ Transaction on System LSI Design Methodology Vol.5, pp. 23-31 Feb. 2012.

[14] X. Peng, Z. Chen, X. Zhao, D. Zhou and S. Goto, “A 115mW 1Gbps QC-LDPC decoder ASIC for WiMAX in 65nm CMOS,” IEEE Asian Solid-State Circuits Conference (A-SSCC 2011), pp. 317-320, Jeju, Korea, Nov. 14-16, 2011.

[15] R.-Y. Shao, S. Lin, and M.P.C. Fossorier, “Two simple stopping criteria for turbo decoding,” in IEEE Trans. on Comm., Vol. 47, pp. 1117-1120, Aug. 1999.

[16] Y. Ueng and Y. Wang, “Modified layered message passing decoding with dynamic scheduling and early termination for QC-LDPC codes,” in Proc. IEEE ISCAS 2009, pp. 121-124, May 2009.

[17] F. Kienle and N. Wehn, “Low complexity stopping criterion for LDPC code decoders,” in Proc. 2005 IEEE Vehicular Technology Conf., VTC2005-Spring, vol. 1, pp. 606-609, May 2005.

[18] J. Li, X. H. You, and J. Li, “Early stopping for LDPC decoding: Convergence of Mean Magnitude (CMM),” IEEE Commun. Lett., vol. 10, no. 9, pp. 667-669, Sep. 2006.

[19] D. Shin, K. Heo, S. Oh and J. Ha, “A stopping criteria for low-density parity-check codes,” in Proc. 2007 IEEE Vehicular Technology Conf., VTC2007-Spring, vol. 1, pp. 1529-1533, June 2007.

[20] A. Segard, F. Verdier, D. Declercq and P. Urard, “A DVB-S2 compliant LDPC decoder integrating the horizontal shuffle scheduling,” ISPACS 2006, pp. 1013-1016, 2006.

[21] S. Muller, M. Schreger, M. Kabutz, M. Alles, F. Kienle and N. Wehn, “A novel LDPC decoder for DVB-S2 IP,” in Proc. 2009 Design, Automation & Test in Europe Conference & Exhibition, (DATE 09), pp. 1308-1313, 2009.

[22] C. Marchand, J.-B. Dore, L. Conde-Canencia and E. Boutillon, “Conflict resolution by matrix reordering for DVB-T2 LDPC decoders,” IEEE GLOBECOM 2009, pp. 1-6, 2009.

[23] C. Marchand, J.-B. Dore, L. Conde-Canencia and E. Boutillon, “Conflict resolution for pipelined layered LDPC decoders,” 2009 IEEE Workshop on Signal Processing Systems (SiPS 2009), pp. 220-225, 2009.

[24] W. Ji, M. Hamaminato, H. Nakayama and S. Goto, “Data conflict resolution for layered LDPC decoding algorithm by selective recalculation,” 2010 3rd International Congress on Image and Signal Processing (CISP 2010), pp. 2985-2989, 2010.

[25] Y. Sun, M. Karkooti and J. R. Cavallaro, “VLSI decoder architecture for high throughput, variable block-size and multi-rate LDPC codes,” Proc. IEEE ISCAS, pp. 2104-2107, 2007.

[26] B. Xiang, D. Bao, S. Huang and X. Zeng, “An 847–955 Mb/s 342–397 mW dual-path fully-overlapped QC-LDPC decoder for WiMAX system in 0.13μm CMOS,” IEEE Journal of Solid-State Circuits, vol. 46, no. 6, pp. 1416-1432, June 2011.

[27] C.-H. Liu, S.-W. Yen, C.-L. Chen, H.-C. Chang, C.-Y. Lee, Y.-S. Hsu and S.-J. Jou, “An LDPC decoder chip based on self-routing network for IEEE 802.16e applications,” IEEE JSSC, vol. 43, no. 3, pp. 684-694, Mar. 2008.

[28] X.-Y. Shih, C.-Z. Zhan, C.-H. Lin and A.-Y. Wu, “An 8.29 mm^2 52 mW

multi-mode LDPC decoder design for mobile WiMAX system in 0.13 μm CMOS process," IEEE JSSC, vol.43, no. 3, pp. 672-683, 2008.

[29] X. Peng, Z. Chen, X. Zhao, D. Zhou, S. Goto, "A 115mW 1Gbps QC-LDPC decoder ASIC for WiMAX in 65nm CMOS," Proc. IEEE A-SSCC, pp. 317-320, Nov. 2011.

[30] R. M. Tanner, "A recursive approach to low complexity codes," IEEE Trans. Inf. Theory, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.

[31] R.L. Townsend and E. Weldon, "Self-orthogonal quasi-cyclic codes," IEEE Trans. Inform. Theory, vol. IT-13, no.2, pp. 183-195, April 1967.

[32] H. Jin, A. Khandekar and R. McEliece, "Irregular repeat-accumulate codes," in Symp. on Turbo codes, pp. 1-8, 2000.

[33] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Dept. Electrical Eng., Univ. Linkoping, Linkoping, Sweden, 1996.

[34] J. Chen and M. C. Fossorier, "Decoding low-density parity-check codes with normalized APP-based algorithm," Proc. IEEE GLOBECOM, vol. 2, pp. 1026–1030, 2001.

[35] J. Chen, "Reduced complexity decoding algorithms for low-density parity-check codes and turbo codes," Ph.D. dissertation, Dept. Electrical Eng., Univ. Hawaii, Honolulu, HI, 2003.

[36] E. Sharon, S. Litsyn and J. Goldberger, "An efficient message-passing scheduling for LDPC decoding," Proc. 23rd IEEE Convention in Tel-Aviv, pp. 223-226, 2004.

[37] A. Hashimoto and Y. Suzuki, "A new transmission system for the advanced satellite broadcast," IEEE Trans. on Consumer Electronics, vol. 54, Issue 2, pp. 353-360, 2008.

[38] A. J. Blanksby and C. J. Howland, “A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder,” IEEE Journal of Solid-State Circuits, vol. 37, no. 3, pp. 404-412, March 2002.

[39] L. Zhou, C. Wakayama, N. Jankrajarng, B. Hu and C.-J. R. Shi, “A high-throughput low-power fully parallel 1024-bit 1/2-rate low density parity check code decoder in 3D integrated circuits,” Proc. ASP-DAC, pp. 92-93, Jan. 2006.

[40] C. Chung, Y. Ueng, M. Lu and M. Lin, “Adaptive quantization for low-density-parity-check decoders,” Proc. ISITA, pp. 13-18, 2010.

[41] T. Ohtsuki, “LDPC codes in communications and broadcasting,” IEICE TRANS. on Communications, vol. E90-B, no. 3, pp. 440-453, Mar. 2007.

Publication List

Journal Paper:

[1] Xiongxin Zhao, Zhixiang Chen, Xiao Peng, Dajiang Zhou and Satoshi Goto, “A 5.83 pJ/bit/iteration High-Parallel Performance-Aware LDPC Decoder IP Core for WiMAX in 65nm CMOS,” IEICE TRANS. on Fundamentals: Special Section on VLSI Design and CAD Algorithms, Vol.E96-A, No.12, pp. 2623-2632, Dec. 2013.

[2] Xiongxin Zhao, Zhixiang Chen, Xiao Peng, Dajiang Zhou and Satoshi Goto, “A 115mW 1Gbps Bit-Serial Layered LDPC Decoder for WiMAX,” IEICE TRANS. on Fundamentals: Special Section on VLSI Design and CAD Algorithms, Vol.E95-A, No.12, pp. 2384-2391, Dec. 2012.

[3] Xiongxin Zhao, Zhixiang Chen, Xiao Peng, Dajiang Zhou and Satoshi Goto, “DVB-T2 LDPC Decoder with Perfect Conflict Resolution,” IPSJ Transaction on System LSI Design Methodology Vol.5, pp. 23-31, Feb. 2012.

[4] Zhixiang Chen, Xiao Peng, Xiongxin Zhao, Leona Okamura, Dajiang Zhou and Satoshi Goto, “A 6.72-Gb/s 8pJ/bit/iteration IEEE 802.15.3c LDPC decoder chip,” IEICE TRANS. on Fundamentals, Vol. E94-A, No.12, pp. 2587-2596, Dec. 2011.

[5] Xiao Peng, Xiongxin Zhao, Zhixiang Chen, Fumiaki Maehara, Satoshi Goto, “Generic Permutation Network for QC-LDPC Decoder,” IEICE Trans. Vol.E93-A, No.12, pp. 2551-2559, Dec. 2010.

[6] Zhixiang Chen, Xiongxin Zhao, Xiao Peng, Dajiang Zhou and Satoshi Goto, “A High Parallelism LDPC Decoder with an Early Stopping Criterion for WiMax and WiFi Application,” IPSJ Transaction on System LSI Design Methodology, Vol. 3, pp. 292-302, Aug. 2010.

[7] Xiao Peng, Zhixiang Chen, Xiongxin Zhao, Fumiaki Maehara and Satoshi Goto, “Permutation Network for Reconfigurable LDPC decoder based on Banyan Network,” IEICE Trans. Electronics, Vol.E93-C, No.3, pp. 270-278, Mar. 2010.

International Conference Paper (with review):

[1] Xiongxin Zhao, Zhixiang Chen, Xiao Peng, Dajiang Zhou and Satoshi Goto, "High-Parallel Performance-Aware LDPC Decoder IP Core Design for WiMAX," IEEE International Midwest Symposium on Circuits and Systems (MWSCAS 2013), pp. 1136-1139, Ohio, USA, Aug. 4-7, 2013.

[2] Xiongxin Zhao, Zhixiang Chen, Xiao Peng, Dajiang Zhou and Satoshi Goto, "DVB-T2 LDPC Decoder with Perfect Conflict Resolution," 2011 IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT 2011), pp. 1-4, Hsinchu, Taiwan, Apr. 23-25, 2012.

[3] Xiao Peng, Zhixiang Chen, Xiongxin Zhao, Dajiang Zhou and Satoshi Goto, "A 115mW 1Gbps QC-LDPC decoder ASIC for WiMAX in 65nm CMOS," IEEE Asian Solid-State Circuits Conference (A-SSCC 2011), pp. 317-320, Jeju, Korea, Nov. 14-16, 2011.

[4] Ying Cui, Xiao Peng, Zhixiang Chen, Xiongxin Zhao, Yichao Lu, Dajiang Zhou and Satoshi Goto, "Ultra Low Power QC-LDPC Decoder with High Parallelism," The 24th IEEE International SOC Conference (SOCC 2011), pp. 142-145, Taipei, Taiwan, Sept. 26-28, 2011.

[5] Zhixiang Chen, Xiao Peng, Xiongxin Zhao, Qian Xie, Leona Okamura, Dajiang Zhou and Satoshi Goto, "A Macro-Layer Level Fully Parallel Layered LDPC Decoder SoC for IEEE 802.15.3c Application," 2011 IEEE International Symposium on VLSI Design, Automation and Test (VSLI-DAT 2011), Hsinchu, Taiwan, Apr. 26, 2011.

[6] Xiongxin Zhao, Zhixiang Chen, Xiao Peng, Dajiang Zhou and Satoshi Goto, "A BER performance-aware early termination scheme for layered LDPC decoder," 2010 IEEE Workshop on Signal Processing Systems (SiPS 2010), pp. 416-419, Oct. 6th, 2010.

[7] Xiao Peng, Zhixiang Chen, Xiongxin Zhao, Fumiaki Maehara, Satoshi Goto, "High Parallel Variation Banyan Network Based Permutation Network for Reconfigurable LDPC Decoder," 2010 21st IEEE International Conference on Application-specific Systems Architectures and Processors (ASAP 2010), pp. 233-238, July 7th, 2010.

[8] Zhixiang Chen, Xiongxin Zhao, Xiao Peng, Dajiang Zhou and Satoshi Goto, "An Early Stopping Criterion for Decoding LDPC Codes in WiMax and WiFi Standards," 2010 IEEE International Symposium on Circuits and Systems (ISCAS 2010), pp. 473-476, Paris, France, Jun. 2010.

[9] Zhixiang Chen, Xiongxin Zhao, Xiao Peng, Dajiang Zhou and Satoshi Goto "A High-Parallelism Reconfigurable Permutation Network for IEEE 802.11n and 802.16e LDPC Decoder," International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2009), pp. 85-88, Kanazawa, Japan, pp. 85-88, Dec.

2009.

[10] Zhixiang Chen, Xiongxin Zhao and Satoshi Goto, "A Memory Efficient Check Message Quantization Scheme for LDPC decoder," 2009 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2009), pp. 1412-1415, Jeju, Korea, Jul. 2009.

[11] Xiongxin Zhao, Zhixiang Chen and Satoshi Goto, "High Efficiency Architecture for DVB-S2 Based LDPC Decoder," 2009 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2009), pp. 1558-1561, Jeju, Korea, Jul. 2009.