行政院國家科學委員會補助專題研究計畫 ■成果報告
□期中進度報告

（計畫名稱）應用於無線通訊之低功耗基頻處理器

計畫類別：■個別型計畫　□整合型計畫
計畫編號：NSC97－2220－E－009－166－MY3
執行期間：　97 年 8 月 1 日至 100 年 7 月 31 日

執行機構及系所：國立交通大學 電子工程學系

計畫主持人：李鎮宜 教授
計畫參與人員：陳志龍、林義閔、李欣儒、林佳龍、許智翔

成果報告類型(依經費核定清單規定繳交)：□精簡報告　■完整報告

本計畫除繳交成果報告外，另須繳交以下出國心得報告：
□赴國外出差或研習心得報告
□赴大陸地區出差或研習心得報告
■出席國際學術會議心得報告
□國際合作研究計畫國外研究報告

處理方式：除列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

中　華　民　國　100　年　10　月　31　日

目錄

應用於無線通訊之低功耗基頻處理器

A Low-Power Baseband Processor for Wireless Communication System

計畫編號: NSC97-2220-E-009-166

執行期間: 97 年 8 月 1 日 至 100 年 7 月 31 日

主持人：李鎮宜 交通大學電子工程系教授

參與學生：陳志龍、林義閔、賴義澤、李欣儒、林佳龍、許智翔

I. 摘要

中文摘要

基頻訊號處理在無線通訊系統上扮演關鍵性的角色，不僅可有效提升傳輸的效能，更能提供多模式和多標準的系統實現方案。然而要達成低成本和低功率設計方法，不僅對於個別模組的演算法需深入瞭解外，也必須融入系統層級的行為，方能提供一個具有技術競爭力的解決方案。因此在這三年的研究計畫，我們針對 OFDM 主流無線通訊系統所需求的關鍵模組 Viterbi decoder、LDPC decoder、RS decoder 及 BCH decoder 進行相關議題的研究，探討低功率 Viterbi decoder，高速、低功率 LDPC decoder 以及低複雜度的軟性 BCH、RS decoder 的設計方式，並研究在不同設計規範下達成多模和多標準的作業模式，之後將會把此關鍵模組設計整合，並配合基頻訊號的同步模組電路，完成一個符合多模、多標準的低功率基頻處理器。

關鍵字

基頻處理器；多模式；多標準；低成本；低功率；Viterbi Decoder；LDPC Decoder；BCH Decoder；RS Decoder

英文摘要

Signal processing in baseband processor designs plays a key role in wireless communication system designs—in not only improving overall system transmission performance, but also providing the capability of multi-mode and multi-standard for cost-effective system realization. To reach better performance indices in terms of low-cost and low-power, it is necessary to investigate system design methodologies, covering in-depth exploration of algorithms of key modules and exploitation of unique features/behaviors of a complete system. As a result, a more competitive solution can be delivered. In three year (2008/8~2011/7), we have concentrated on the key modules (Viterbi decoder, LDPC decoder, BCH decoder and RS decoder) of the main-stream OFDM wireless communication systems. The first issue is low power solution for Viterbi decoder. The second issue is the high-speed solution for LDPC decoder. The last issue is the low-cost solution for soft BCH and RS decoder. In the end, these design techniques and key modules will be integrated on a design platform, together with synchronization modules, to come up with a multi-mode, multi-standard, and low-power baseband processor.

Keywords

Baseband Processor, Multi-mode, Multi-Standard, Low-Cost, Low-Power, Viterbi Decoder, LDPC Decoder, BCH Decoder, RS Decoder

II. 計畫的緣由與目的

A. LDPC Decoder

1. A 11.5-Gbps LDPC Decoder Based on CP-PEG Code Construction

Low-density parity-check (LDPC) code is a famous error control code with near Shannon limit performance [A-1] and can be described by its parity-check matrix H. The message exchanging order between nodes is called scheduling, which will influence the convergence speed of the decoding algorithm. In standard BP algorithm, simultaneous update of all check node messages or variable node messages is named as flooding scheduling. Alternatively, the layered BP algorithm [A-3] performing message update along different check node groups is a method of check-node-centric sequential scheduling (CSS). Researchers have revealed that CSS could reduce maximum iteration to approximate half of the standard BP with similar performance.

Recently, LDPC codes adopted in high-throughput systems have high code-rate property to increase channel efficiency. However, the introduced large check node degree dc will cause implementation full of difficulties. Even though the CSS could reduce the iteration number, the throughput is still degraded due to long critical path of check node unit (CNU).

In this project, the proposed decoder aims at providing a high-throughput and hardware-efficient solution to the high code-rate LDPC with large check node degrees. In order to reduce the iteration number, the decoding scheduling is based on the variable-node-centric sequential scheduling (VSS; also known as shuffled decoding [A-6]), where the messages are updated along different variable node groups. Since the inputs of CNU operation are also divided into several subgroups, the complexity and critical path delay of CNU are reduced. Furthermore, single pipelined approach and modified CNU are proposed to minimize the message storage memory. Using a (2048, 1920) LDPC code constructed by circulant permutation progressive edge-growth (CP-PEG) algorithm [A-7] as a design example, the overall decoder chip

implemented in 90nmtechnology will show its advantages in terms of throughput, energy efficiency, and hardware efficiency.

2. A 2.37-Gb/s 284.8mW Rate-Compatible (491,3,6) LDPC-CC Decoder

Near the rediscover of LDPC-BCs, LDPC-CCs were proposed in 1999 [A-10]. LDPC-CCs have the characteristics of convolutional code not found in LDPC-BCs. Continuous encoding supports any length of input data stream, which is especially suitable for streaming video and packet-switching network. The puncture scheme applied in LDPC-CCs provides flexible code-rates by abandoning certain positions of encoded bits according to the puncture table. Simple encoder circuitry composed by registers, multiplexers and a few XOR gates has lower hardware cost and power consumption, and can be used in distributed sensor network. Furthermore, the correlation between codeword symbols of LDPC-CCs is limited to a specific interval (constraint length m_s+1, m_s is the memory size of encoder). This locality property lowers the overall routing complexity of the decoder. Although possessing many advantages, LDPC-CCs were few chosen by standards. The main reason lies in its bottlenecks of the long decoding latency, high power consumption, and low-to-moderate decoding throughput.

The throughput of LDPC-CC decoders reported in literatures was only several hundred Mbps, which is difficult to compete with LDPC-BC decoder with several tens of Gbps throughput. Cause of lower throughput can be explained by the decoder structure. LDPC-CC decoder consists of serially concatenated processors, and each processor decoding a sliding window on the trellis diagram can be taken as one iteration in LDPC-BCs. Increasing processor number can enhance error-correcting capability but cannot increase throughput. Therefore many works put efforts on realizing the parallel message passing: analysis of parallelization concepts in [A-11], single-instruction-multiple-data (SIMD) architecture in [A-12], and joint code-decoder design in [A-13]. Recently, a high throughput LDPC-CC decoder design was proposed by adding regularity during code construction [A-14]. However, achieving high throughput is still challenging for some time-varying LDPC-CC code without regularity.

B. BCH and RS Decoder

1. Soft BCH Decoder Chip for DVB-S2 System

The Bose-Chaudhuri-Hocquenghen (BCH) [B-1] codes are popular in storage and communication systems recently. From DMB-T [B-2] and DVB-S2 [B-3] applications shown in Fig. 1, the BCH codes with long block length are specified to suppress the error floor due to iterative LDPC decoding. Since BCH codes perform as outer codes in those communication systems, the soft information from the inner decoder can be employed to further improve the error-correcting performance.

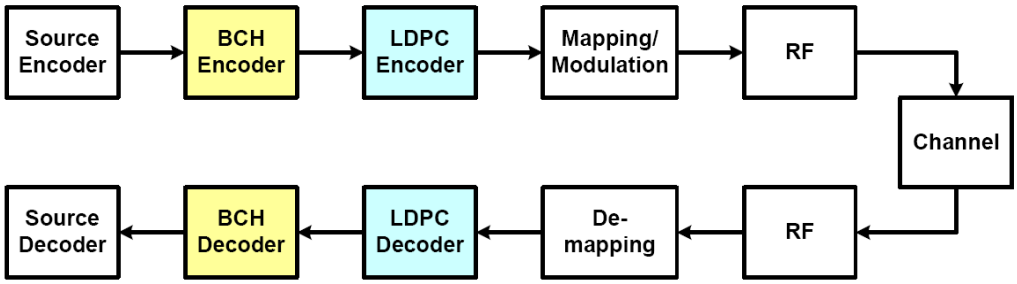


Fig. 1. Block diagram of DMB-T and DVB-S2 systems

Soft decision decoding of BCH codes has aroused many research interests. Forney developed the generalized-minimum-distance (GMD) [B-4], which uses algebraic algorithms to generate a list of candidate codewords and chooses a most likely codeword from the list. Other algorithms with the same concept of candidate list, such as Chase [B-5] and SEW [B-6], are also widely used in many applications. This report illustrates a soft BCH decoding method using error magnitudes [B-7] to deal with the least reliable bits. For example, Fig. 2 shows the results of a concatenated code with 16-state BCJR [B-8] and BCH (255,239) over GF(2^8). Based on the soft information from previous decoder, the performance gain of the BCH decoder is about 0.73 db at BER = 10^{-6} when 2t+1 candidate bits within a codeword are chosen to correct errors.

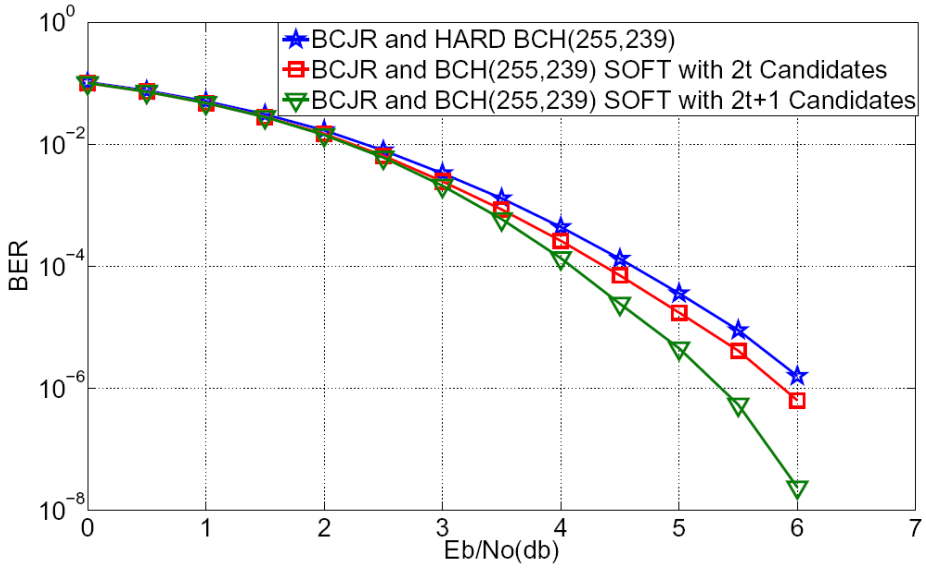


Fig. 2. Simulation results for BCH (255,239) concatenating with a 16-state BCJR under BPSK modulation and AWGN channel

2. An Improved Soft BCH Decoder with One Extra Error Compensation

The Bose-Chaudhuri-Hocquenghen (BCH) [B-1] codes are popular in storage and communication systems, such as flash device, DMB-T [B-2] and DVB-S2 [B-3] broadcasting systems. Recently, soft decoding of BCH codes has aroused many research interests. Forney developed the generalized-minimum-distance (GMD) [B-4] to generate a list of candidate codewords and choose a most likely codeword from the list. Other algorithms with similar concept, such as Chase [B-5] and SEW [B-6], are also widely used in many applications. Moreover, Therattil and Thangaraj provided a sub-optimum MAP BCH decoding method with Hamming SISO decoder in 2005 [B-12].

In general, the complexity of a soft BCH decoder is much higher than a hard BCH decoder for decoding an entire codeword. Nevertheless, soft BCH decoders with lower complexity can be revealed by focusing on the least reliable bits instead of the whole codeword. A soft BCH decoding algorithm using error magnitudes to deal with the least reliable bits was developed in 1997 [B-7]. However, Fig. 3 shows that there is about 0.25 dB performance loss at BER = 10−5 in AWGN channel as compared to hard decision BCH decoder for BCH (255,239) code. For the existing soft decision algorithms, the soft BCH decoder provides either better error correcting

performance or lower hardware complexity than a traditional hard BCH decoder. In this project, a soft BCH decoder which has similar concept as [B-7] and enhances the correcting performance by compensating one extra error while maintaining the low hardware complexity is presented.

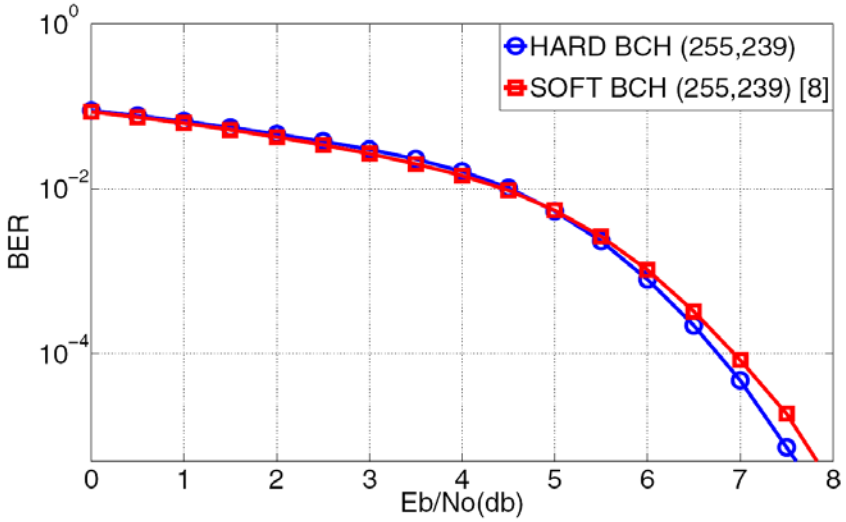


Fig. 3. Simulation Result for BCH (255,239)

3. Soft RS Decoder Chip for Optical Communication System

Reed-Solomon (RS) codes are widely used in various communications and digital data storage systems due to the advantage of overcoming the burst errors. According to International Telecommunication Union (ITU-T) recommendation, RS (255,239) is standardized in high speed optical fiber systems and Gigabit Passive Optical Network (GPON) applications, which demand 2.5 Gb/s throughput for achieving 10-40 Gb/s with 16 RS decoders or satisfying the maximum up and down link requirement. To resist the increasing noise induced by higher transmission rate required in optical communication systems, soft RS decoding algorithms are exploited for achieving considerable performance gain. These algorithms modify the received sequences to form a list of candidate codewords and choose the most probable one. Nevertheless, because of high computational complexity, these soft decoding algorithms are still unsuitable for practical implementation. In this project, a decision-confined soft decoding algorithm is proposed to enhance performance while maintaining area efficiency. Instead of decoding all the candidate codewords like other soft decoding algorithms, our design only decodes the candidate codeword with the degree of

error-locator polynomial $\Lambda(x)$ less than error correction capability t. The Gray code based bit-flipping method is also exploited leading to only one suit of hardware requirement.

C. Viterbi Decoder

1. A Low-Power Viterbi Decoder Based on Scarce State Transition and Variable Truncation Length

The Viterbi decoder implementing the Viterbi algorithm [C-1] for decoding convolutional codes is composed of three main blocks: the branch metric (BM) unit, the ACS unit, and the survivor memory. The BM unit generates branch metrics from the input data. The ACS unit recursively accumulates branch metrics as path metrics (PM) and makes decisions to select the most likely state transition sequences, or the survivors. Survivor memory stores the survivors for retrieving the data sequence.

There are two well-known survivor memory management approaches: the register-exchange (RE) and the trace-back (TB) [C-2]. The register-exchange is conceptually the simplest technique that eliminates repeatedly memory access operations. Therefore, this approach has shorter latency and is suitable for high speed decoder implementations. However, due to the data movement among registers, the approach is considered to be power inefficient.

Fig. 4 shows the conventional 2^{υ}-state Viterbi decoder with the register-exchange architecture [C-3]. The decisions from ACS units will be shifted within the survivor memory from left to right. Applying the scarce state transition (SST) technique and the variable truncation length, we illustrate the proposed low-power Viterbi decoder for the MB-OFDM UWB system [C-4] in Fig. 5. The SST unit is integrated to reduce state transition activities, leading to less dynamic power consumption [C-8]. Furthermore, the path merging detector monitors the merged point for all survivors and adjusts the truncation length to avoid unnecessary data movement in registers. Many redundant operations in the survivor memory can be reduced to save power dissipation. Additionally, considering the high throughout

requirement, the radix-4 ACS structure is exploited because of a better compromise between cost and throughput [C-5].

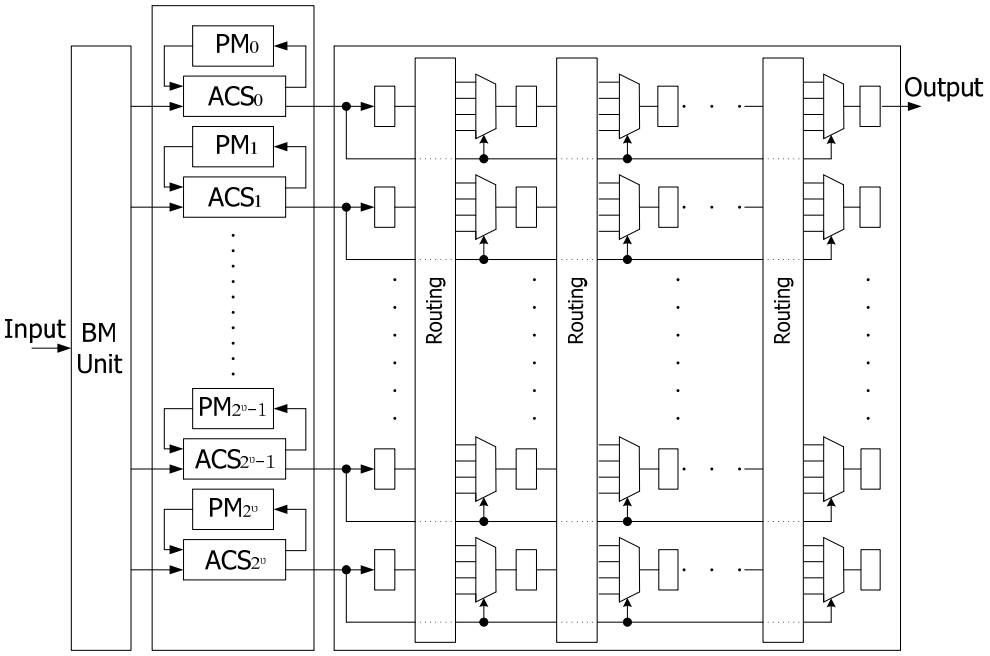


Fig. 4. The conventional register-exchange architecture

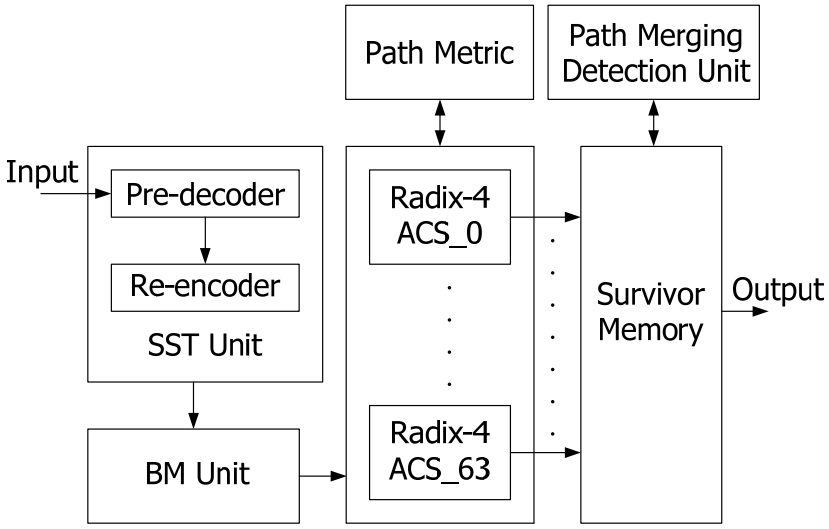


Fig. 5. The proposed Viterbi Decoder Architecture

For the Viterbi decoder, the scarce state transition (SST) algorithm is a low power technique to reduce the state transition activity significantly under high SNR conditions [C-6]-[C-8]. In the conventional Viterbi decoding model (see Fig. 6 (A)), u(D) denotes the information sequence, C(D)=u(D)G(D) is the codeword sequence deriving from the generator polynomial G(D). From the received sequence r(D), the Viterbi decoder estimates the decoded

information o(D).

The SST Viterbi decoding architecture in Fig. 6 (B) includes two additional blocks: pre-decoder and re-encoder. Assume

$$r(D) = u(D) \cdot G(D) + e(D) = C(D) + e(D) \quad (1)$$

and e(D) is the error sequence from a noisy channel, the pre-decoder directly decode the information sequence from r(D):

$$i(D) = r(D) \cdot G^{-1}(D) = \hat{u}(D) \quad (2)$$

The re-encoder then encodes i(D) to a new codeword z(D).

$$z(D) = i(D) \cdot G(D) = \hat{C}(D) \quad (3)$$

The Viterbi decoder performs maximum likelihood decoding on y(D), which is defined as follows:

$$y(D) = r(D) + z(D) = C(D) + e(D) + \hat{C}(D) \quad (4)$$

In high SNR conditions, e(D) is nearly zero, and the decoded information sequence becomes

$$o(D) = i(D) + n(D) = \hat{u}(D) + n(D) \quad (5)$$

If the channel condition is good enough, the decoder estimates an approximately zero sequence; as a result, the dynamic power is reduced as the channel becomes better.

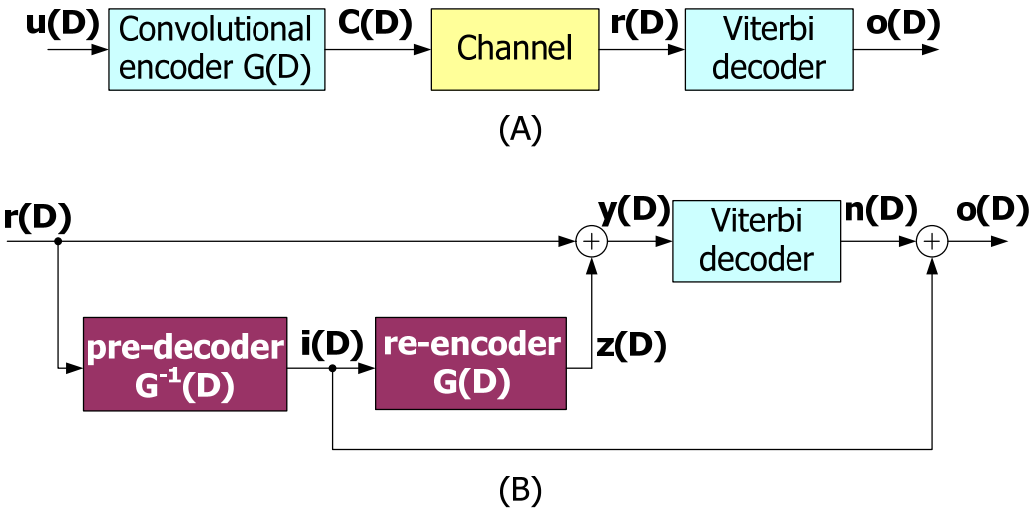


Fig. 6. (A) Conventional model (B) SST decoding model

2. A Low Power Differential Cascode Voltage Switch with Pass Gate Pulsed Latch for Viterbi Decoder

In mobile communication systems, and especially in wireless local area networks (WLAN), information must be transmitted at high data rates. Furthermore, an efficient error-control code is commonly adopted to enhance system performance. Accordingly, convolution codes have been exploited extensively in communication systems, as they provide a superior error correction capacity while maintaining reasonable coding complexity. The Viterbi algorithm is one of the best algorithms for decoding convolution codes with modest computing resources. However, as data rates increase, the power dissipation and system complexity also increase. Moreover, as required transmission rates of wireless systems increase, the error-control mechanism has come to dominate power dissipation.

Fig. 7 presents the power distribution along a Viterbi Decoder we have proposed in [C-10] using UMC 90 nm CMOS technology. The survivor memory unit (SMU) is constricted by a register-array and dissipates most power in a Viterbi Decoder. Therefore, high performance, low power consumption, and robustness are the basic requirements of the design of clocked storage elements in a Viterbi decoder. This project presents a low power differential cascade voltage switch with pass-gate (DCVSPG) pulsed latch for the Viterbi decoder.

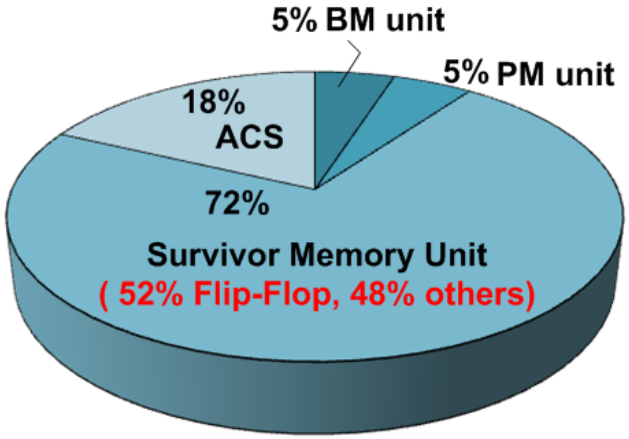


Fig. 7. Power distributions of a Viterbi Decoder

III. 研究方法及成果

A. LDPC Decoder

1. A 11.5-Gbps LDPC Decoder Based on CP-PEG Code Construction

a) CODE STRUCTURE AND DECODING ALGORITHM

(1) CP-PEG LDPC Code Construction

The (2048, 1920) irregular LDPC code, rate-15/6, used in this project was constructed by CP-PEG algorithm and shown in Fig. 8(a). The constructed parity-check matrix H consists of p*p circulant permutation (CP) and all-zero matrices. A CP matrix is a cyclic square matrix with constant row and column weight of one. The number of each CP matrix indicates the cyclic shift amount and -1 means all zero matrixes. By setting p=32, there are 4*p check nodes and 64*p variable nodes in bipartite graph, where each check node has uniform degree 46, and 16*p, 24*p, 24*p variable nodes have degrees of 4, 3, 2 respectively. The performance of this code was proven to have better performance than other PEG-based LDPC codes [A-7]; nevertheless, the high check node degree required suitable decoder architecture to overcome implementation difficulties.

(2) Variable-node centric Sequential Scheduling

In VSS approach, the initialization, stopping criterion test, and output steps remain the same as the standard BP algorithm. The only difference between two algorithms lies in the updating procedure. The normalized min-sum (NMS) algorithm which compensates the approximation error in check node In VSS approach, the initialization, stopping criterion test, and output steps remain the same as the standard BP algorithm. The only difference between two algorithms lies in the updating procedure. The normalized min-sum (NMS) algorithm which compensates the approximation error in check node is shown and described in the next page.

In this work, the codeword is divided into G=4 groups, therefore the parity-check matrix H is divided into 4 sub-matrices (H1 to H4). As shown in Fig. 8(b), each sub-matrix consists of equal

number of variable nodes with the same degree to reduce the hardware cost of variable node unit (VNU). Moreover, the sub-matrices with the same shift amounts (shaded blue CP matrices) are arranged in the same position thus the routing and control could be further simplified.

1) **Initialization:** $z_{mn}^{(0)} = P_n$,
2) **Iterative Decoding:** For $0 \le g \le G-1$, perform the following two steps.
   a) Check node to variable node update step, for $g \cdot N_G \le n \le (g+1) \cdot N_G - 1$ and each $m \in M(n)$, process

$$\varepsilon_{mn}^{(i)} \approx \prod_{\substack{n' \in N(m) \backslash n \\ n' \le g \cdot N_G - 1}} sign(z_{mn'}^{(i)}) \times \prod_{\substack{n' \in N(m) \backslash n \\ n' \ge g \cdot N_G}} sign(z_{mn'}^{(i-1)})$$

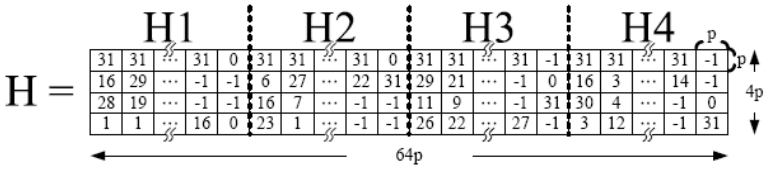
$$\times \min \left\{ \min_{\substack{n' \in N(m) \backslash n \\ n' \le g \cdot N_G - 1}} \left\{ \left| z_{mn'}^{(i)} \right| \right\}, \min_{\substack{n' \in N(m) \backslash n \\ n' \ge g \cdot N_G}} \left\{ \left| z_{mn'}^{(i-1)} \right| \right\} \right\} \times \beta \tag{1}$$

   b) variable node to check node update step, for $g \cdot N_G \le n \le (g+1) \cdot N_G - 1$, process

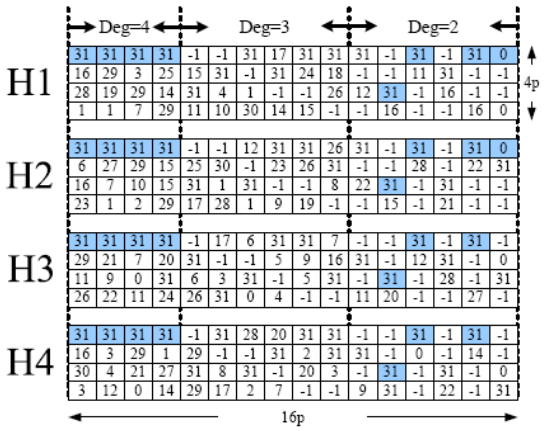
$$z_{mn}^{(i)} = P_n + \sum_{m' \in M(n) \backslash m} \varepsilon_{m'n}^{(i-1)} \tag{2}$$

$$z_n^{(i)} = P_n + \sum_{m \in M(n)} \varepsilon_{mn}^{(i-1)} \tag{3}$$

3) **Hard Decision:** Let X_n be the n-th bit of decoded codeword. If $z_n^{(i)} \ge 0$, $X_n = 0$, else if $z_n^{(i)} < 0$, $X_n = 1$.



(a) Original matrix



(b) Permuted and divided into four groups

Fig. 8. Parity-Check Matrix of(2048,1920) LDPC Code

b) PROPOSED DECODER ARCHITECTURE

In this section, the complete decoder architecture will be presented, including data path, scheduling, and VLSI structure of CNU and modified CNU.

(1) Single Pipelined Architecture

The entire decoder depicted in Fig. 9(a) is composed of fully-parallel CNUs and partial-parallel VNUs, where the VNU2, VNU3, and VNU4 will handle variable node operations with degree 2, 3, and 4 respectively. Let $\alpha_g^{(i)}$ denote the sorted messages sent from variable nodes in the g-th group to one specific check node at i-th iteration, which is:

$$\alpha_g^{(i)} = \min_{\substack{n' \in N(m) \setminus n \\ g \cdot N_G \le n' \le (g+1) \cdot N_G - 1}} \left\{ \left| z_{mn'}^{(i)} \right| \right\} \qquad (4)$$

Then the magnitude part of check node to variable node message in (1) could be computed by the following equation:

$$\left| \varepsilon_{mn}^{(i)} \right| = \min \left\{ \left\{ \alpha_j^{(i)} \right\}_{j<g}, \alpha_g^{(i)}, \left\{ \alpha_k^{(i-1)} \right\}_{k>g} \right\} \qquad (5)$$

Fig. 9(b) demonstrates the timing diagram of proposed decoder. There are G initialization cycles required to calculate $\alpha_g^{(i)}$ for $0 \le g \le G - 1$. Since only one subgroup of the message $z_{mn}^{(i)}$ is updated in g-th cycle of one iteration, the main operation of CNU could be simplified to calculate $\alpha_g^{(i)}$ (local sorting) in each cycle and then perform global sorting like equation (5). From the proposed single pipelined architecture, only messages $\alpha_g^{(i)}$ and $\varepsilon_{mn}^{(i)}$ are stored. The sorted results could be represented by min value, second min value, and the index of min value in NMS algorithm. Therefore, the proposed decoder only latches two values, one index, and sign part of messages in each subgroup, while the variable node to check node message $z_{mn}^{(i)}$ is on-the-fly calculated. The single pipelined architecture is feasible because the CNU could be updated immediately after VNU's operations in VSS approach.

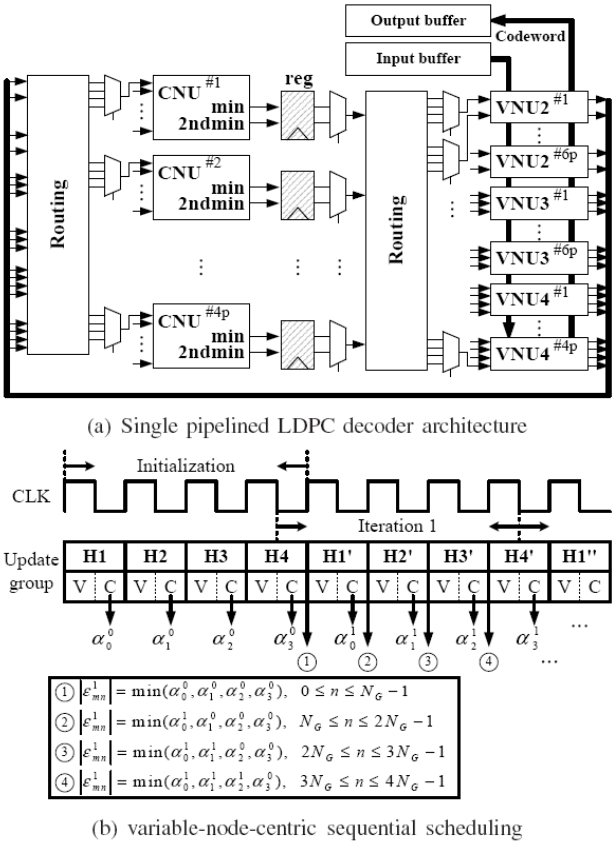


Fig. 9. Proposed Architecture and Scheduling

(2) Modified CUN

The operation of check node to variable node update could be divided into magnitude part and sign part. Fig. 10(a) illustrates the magnitude part of CNU, which is an accumulative sorter composed of a local sorter and a global sorter. The local sorter is used to find the local min and second min values in each subgroups, and global min and second min values of a check node will be found by a global sorter. Similarly, the sign operation can be computed in an accumulative way like the accumulative sorter.

For our proposed CP-PEG LDPC codes with $d_c = 46$, The VSS approach with G = 4 could divide 46 inputs of the sorter into only 12 inputs. More subgroup number G will result in fewer inputs of sorter, but increase the storage for min, second min, and index values of each subgroup. In order to further reduce the storage overhead of each subgroup, we propose a reduced storage accumulative sorter as shown in Fig. 10(b). The basic idea is to simplify the local min and local

second min from G − 1 subgroups into one group. Some extra control circuits are needed to open or close the feedback loop in Fig. 10(b). This sorter architecture is beneficial since the complexity reduction of storage registers and global sorters is higher than the overhead of control circuits. Section IV will show the performance of this modified CNU is similar to original CNU.

(3) Summary

In traditional two-stage pipelined architecture, both $z_{mn}^{(i)}$ and $\varepsilon_{mn}^{(i)}$ messages are kept in registers or memory. Assume the bit-width of messages is w (= 6) and variable node degree is dv, then the required memory size (or registers) is as follows:

$$\begin{aligned} & MEM_{VNU} + MEM_{CNU} \\ = \; & (\sum dv) \cdot w + \\ & (N-K) \cdot (Min + 2ndMin + Index + Sign) \\ = \; & (\sum dv) \cdot w + (N-K) \cdot (2 \cdot (w-1) + log_2\lceil dc \rceil + dc) \\ = \; & 5888 \cdot 6 + 128 \cdot (2 \cdot 5 + log_2\lceil 46 \rceil + 46) = 43264 \end{aligned} \quad (6)$$

For the proposed single pipelined decoder and modified CNU in Fig.4 (b), the memory size is reduced to

$$\begin{aligned} & MEM_{CNU} \\ = \; & (N-K) \cdot \\ & (2 \cdot (Min + 2ndMin + Index + 2ndIndex) + Sign) \\ = \; & (N-K) \cdot (4 \cdot (w-1) + 4 \cdot log_2\lceil dc \rceil + dc) \\ = \; & 128 \cdot (4 \cdot 5 + 4 \cdot log_2\lceil 46 \rceil + 46) = 11520 \end{aligned} \quad (7)$$

Therefore the overall register reduction of proposed architecture is 73%, leading to the following advantages: fewer registers, higher utilization of functional units, and reduced complexity. Since high-rate LDPC codes usually have more VNUs than CNUs (in our case: 512 VNUs and 128 CNUs), the elimination of registers from VNU to CNU not only reduces hardware cost but also lowers power consumption of clock tree.

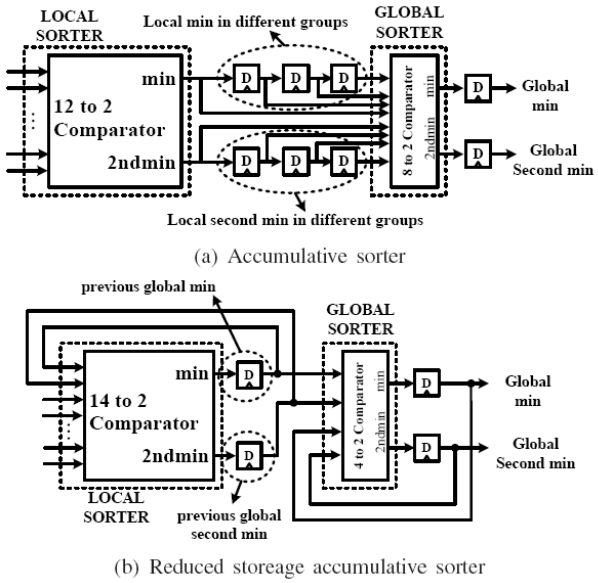


Fig. 10. CNU Architecture

c) PERFORMANCE AND IMPLEMENTATION RESULTS

Under AWGN channel with BPSK modulation, the performance curves are simulated to determine the required bit-width and maximum iteration number. The simulation parameters of proposed algorithm are 6-bit input quantization (5-bit integer and 1-bit decimal fraction), scaling factor 0.75 for NMS algorithm, and 4 or 5 iterations. In Fig. 11, the bit-error rate (BER) curves indicate that 4 iterations for the proposed algorithm are sufficient to achieve similar performance of standard BP algorithm with 7 iterations. Furthermore, in the aspect of almost the same code-rate and better error-correcting capability, our CP-PEG LDPC codes outperforms 1.2 dB better than the (255, 239) RS code at BER=10− 5, which reveals the potential of CP-PEG LDPC codes for storage applications and fiber optical communication systems. The overall SNR loss between this work and Shannon limit is only 1.6dB. The proposed LDPC decoder is implemented by standard-cell design flow and fabricated in 90-nm 1P9M CMOS technology. The core occupied 3.84 mm2 of area with 68% utilization. The die photo is shown in Fig. 12, where the distribution of CNUs and VNUs is auto-determined by APR tool. Since required decoding cycles of one LDPC codeword are 4 initialization cycles plus 4 iterations, the throughput is (1920bit/20cycles)×frequency. Fig. 13 shows the measured maximum throughput and power dissipation under different SNR conditions and supply voltages. The measurement result

indicates that the test chip with FF corner can achieve 11.5 Gbps throughput under 1.4V supply voltage. The throughput could be scaled down to 5.77Gbps with 0.8V supply voltage to meet the throughput requirement of IEEE 802.15.3c standard and the energy efficiency will be 0.033 nJ/bit. Compared with the state-of-the-art in

Table 1, the proposed LDPC decoder outperforms others in the aspects of throughput, hardware efficiency, and power efficiency. Since the LDPC code specifications of these designs are different, the SNR loss between each work to their Shannon limit is also listed for reference.

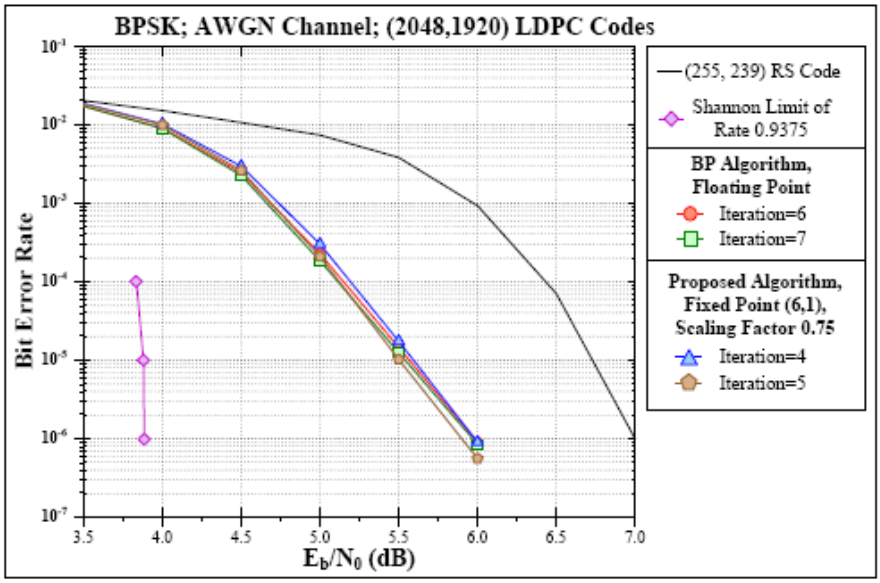


Fig. 11. Performance

d) CONCLUSION

A high-throughput and power-efficient LDPC decoder is presented. Utilizing the characteristic of variable-node-centric sequential scheduling, the proposed decoding algorithm could reduce the maximum iteration number without performance loss. In addition, the single pipelined architecture and modified CNU can save 73% message storage memory and decrease the sorter size, resulting in a low-complexity design. After implementation in 90nm technology, the test chip occupies 3.84 mm^2 of area and supports maximum 11.5 Gbps data rate under 1.4V supply voltage.

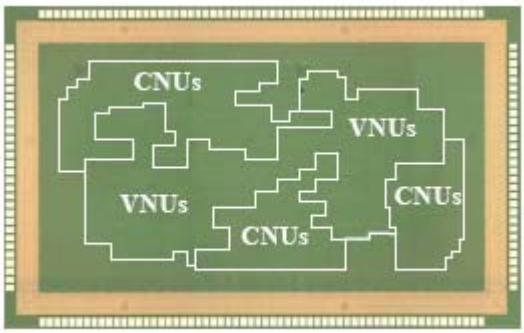
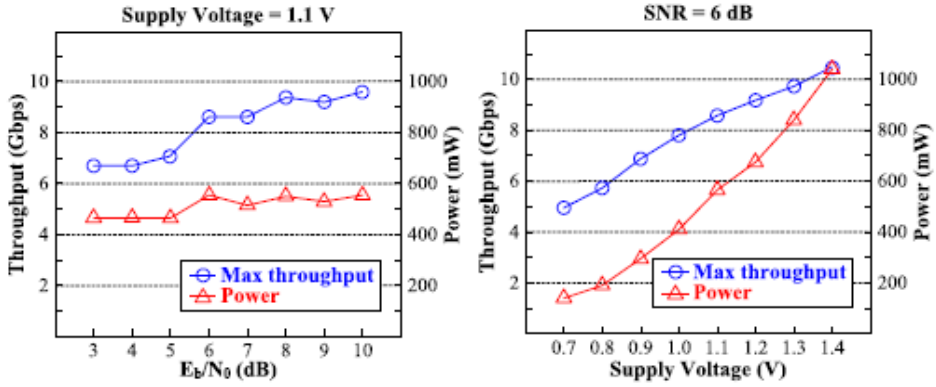


Fig. 12. Chip Micrograph



(a) at different SNR conditions (b) under different supply voltages

Fig. 13. Measured Maximum Throughput and Power Consumption

Table 1

Comparison with the State-of-The-Art

	This work	CICC'07 [8]	SOVC'07 [9]
Process	90-nm	0.13-μm	0.13-μm
Code Spec	(2048, 1920)	(660,480)	Wimax
Code Rate	0.9375	0.73	0.5
Core Area	3.84 mm^2	7.3 mm^2	4.45 mm^2
Gate Count	708k	690k	420k
Iteration	4	15	2 - 8
Input Quantization	6 bits	4 bits	8 bits
Clock Frequency	120 MHz	300 MHz	83.3 MHz
Max. Throughput	11.5 Gbps	2.44 Gbps	222 Mbps
Power	191.2 mW [1]	1383 mW [2]	52 mW
Energy Efficiency	0.033 nJ/bit [1]	0.566 nJ/bit [2]	0.23 nJ/bit
Hardware Efficiency [3]	12.1	3.5	0.53
SNR loss to Shannon limit [4]	1.6 dB	2.8 dB	2.9 dB

[1] when frequency is scaled to 5.77 Gbps throughput with 0.8V supply voltage where BER=10^{-6}

[2] when E_b/N_0=5.5dB indicating BER$< 10^{-8}$

[3] Throughput/Gate count (Mbps/K-gate)

[4] when BER=10^{-5}

2. A 2.37Gb/s 284.8mW Rate-Compatible (491,3,6) LDPC-CC Decoder

a) PROPOSED ALGORITHM AND ARCHITECTURE

Fig. 15 demonstrates the algorithm-level optimization to accelerate the decoding convergence speed by using the on-demand variable node activation (OVA) scheduling technique [A-15]. The main idea is to change the variable activation location leaving from the processor to the position right before each check node input. The OVA scheduling is similar to the layered decoding in LDPC-BCs that check nodes could access the most recent messages. The original VNU can be disassembled into several sub-VNU (SVNUs) and distributed within a processor. Since the equation of VN-to-CN messages (e.g. n_1 and n_2 in Fig. 15) has two common terms, we may calculate n_2 from n_1 by deducting m_2 (done by pre-SVNU) and adding m_1 (done by post-SVNU). Therefore, the channel values (i.e. u and v) are concealed in VN-to-CN messages and the storage space of channel values can be removed from processors to save 17% memory.

When the channel values are concealed within the summation values, the bit-width of each message should be adjusted to avoid truncation error. In the situation of w-bit channel value, the summation values needs (w + 2)-bit. Since the operations of pre-SVNU and post-SVNU are independent, they can be retimed such that the messages between them only need (w+1)-bits. The original critical path from CNU to post-SVNU is also diminished by one adder delay.

Fig. 16 is the bit-error-rate (BER) performance of the rate-compatible (491, 3, 6) time-varying LDPC-CC proposed in [A-16] under AWGN channel. In contrast to log-BP algorithm with 10 processors, the proposed algorithm with 5 processors can achieve similar or even better performance in all code-rates. Therefore, only half processors are required under the same performance, leading to half decoding latency reduction as well.

In the original structure of Fig. 14, the LDPC-CC decoder can only decode one bit in one cycle, so the information throughput will be f_{clk} Mb/s at f_{clk} MHz clock frequency. To increase throughput, the node level optimization duplicates both CNUs and VNUs to ρ (folding factor)

units and the decoder throughput becomes ($\rho \times f_{clk}$) Mb/s. The proposed folding technique primarily duplicates the combinational logic while the sequential circuits are only slightly increased. In the meantime, the FIFO structure is also modified accordingly to provide sufficient input data for operation units. Each FIFO in the conventional processor is segmented by ρ factor to support required bandwidth.

For irregular time-varying LDPC-CC with large folding factor, neither register-based FIFO with high power consumption nor memory-based FIFO with serious memory conflict is suitable. In order to making trade-off between bandwidth and power, the hybrid-partitioned FIFO structure is presented. The first step is calculating the length of the longest continuous sectors of every folded row. Then sectors are to be merged into one memory bank together, where the depth of the memory bank is the minimum value of the sector lengths. If the original sector is larger than the memory depth, the excess part is still stored in registers. This procedure continues to merge sectors until the memory depth is less than a pre-defined parameter. In this work, 50% of messages in each processor are partitioned into three two-port memories (10.5 Kbits) and the clock buffers are also reduced by 54%.

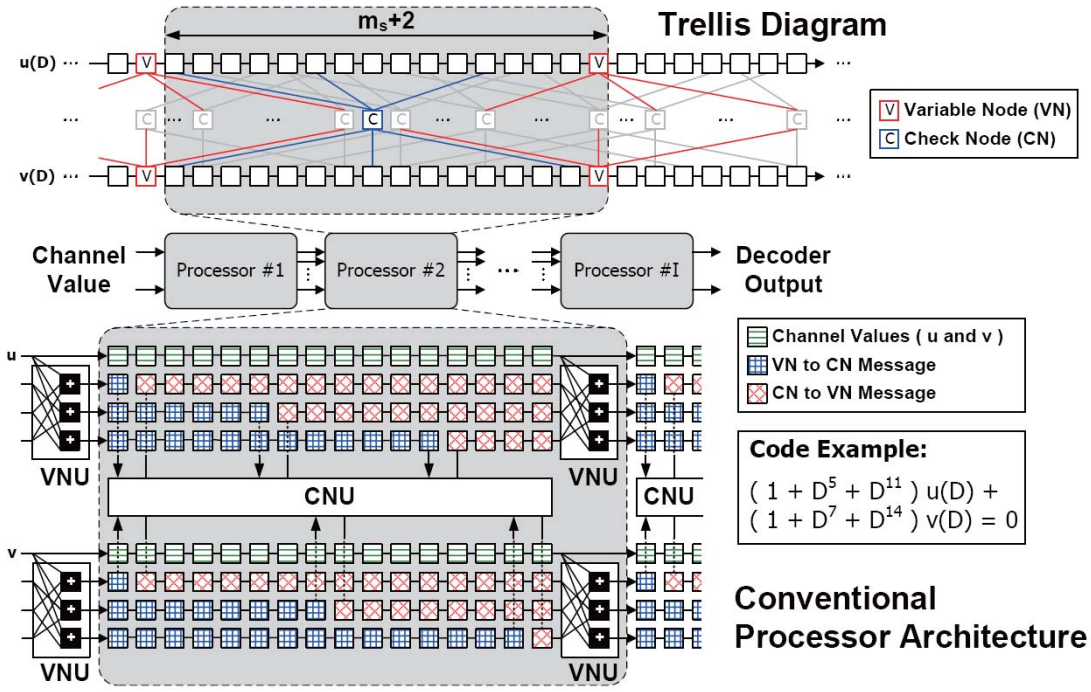


Fig. 14. The (14, 3, 6) LDPC-CC decoder and conventional processor architecture. Note that the constrain length m_s=14, VN degree d_v=3, CN degree d_c=6 in this example.

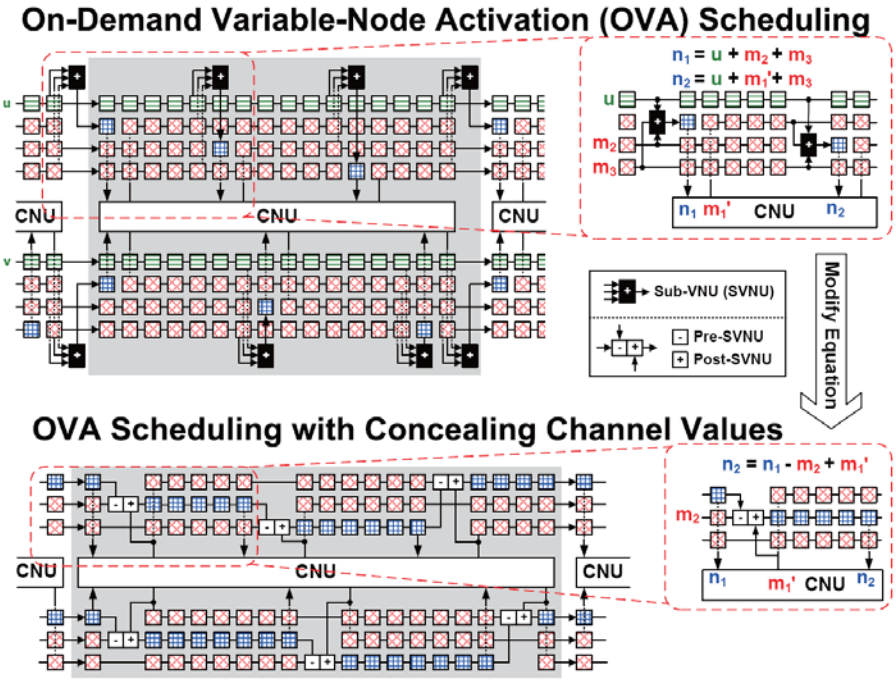


Fig. 15. Algorithm level optimization (OVA scheduling with concealing channel values).

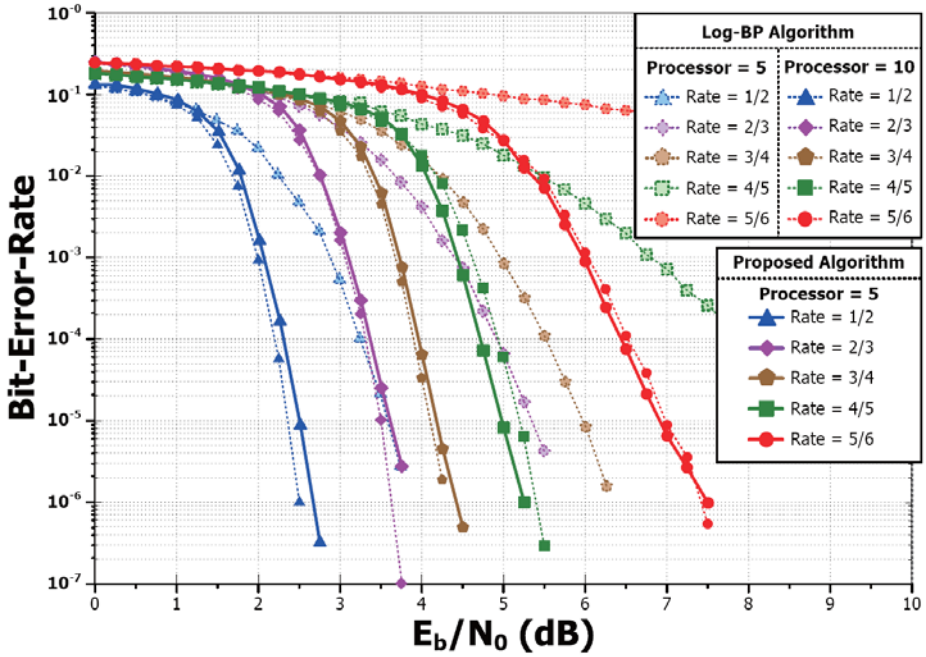


Fig. 16. BER performance of Log-BP algorithm (floating-point) and our proposed scheduling in Normalized Min-Sum algorithm with scaling factor 0.875 (fixed-point (6,2)) under AWGN channel.

b) IMPLEMENTATION RESULTS

Fabricated in 90nm 1P9M CMOS process, our test chip integrates the OVA scheduling with concealed channel values, folding architecture, re-timed SVNU, and hybrid-partitioned FIFOs. Key features and performance comparison are given in Table 2. The decoder chip occupies

2.24mm2 area with 479K gates and 52.5Kb SRAM. Measurement results show that the decoder draws 284mW under 1.2V supply voltage while running at 198MHz. Since the folding factor equals 12, the information throughput of the LDPC-CC decoder achieves 2.37 Gb/s. When supply voltage is scaled down to 0.8V as shown in Fig. 17, the power is reduced to 90.2 mW with an energy efficiency of 0.0114 nJ/bit/proc. Compared with other LDPC-CC decoders [A-14], [A-17], this work provides higher throughput, less area, and better energy efficiency. Compared with the Turbo decoder [A-18], this work achieves much higher throughput with lower power and less die area. In conclusion, our proposed LDPC-CC decoder outperforms state-of-the-art designs and has the potential to be one candidate for next-generation communication systems. The chip micrograph is shown in Fig. 18.

Table 2

Chip summary and comparison with state-of-the-art

	This work	[B-8]	[B-5]	[B-9]
FEC Type	LDPC-CC	LDPC-CC	LDPC-CC	Turbo Code
Constraint Length [(a)]	984	258	960	3200
Code-Rate	1/2, 2/3, 3/4, 4/5, 5/6	1/2	1/2	1/3
CMOS Technology	90-nm	90-nm	90-nm	0.13-μm
Input Quantization	6 bits	8 bits	6 bits	-
Processor [(a)]	5	3	1	5.5
Memory	52.5 kb	-	23.04 kb	129 kb
Area (mm^2)	2.24	1.5	0.924	3.57
Frequency (MHz)	198 [(b)]	600	250	302
Data Rate (Gb/s)	2.37 [(b)]	0.6	2.0	0.39
Power (mW)	284 [(b)]	368.7	-	788.9
Energy Efficiency (nJ/bit/proc)[(a)]	0.024 [(b)]	0.2048	0.064	0.37
	Measured	Measured	Synthesis	Measured

[(a)] these terms represent block size, iteration, and nJ/bit/iter in Turbo code
[(b)] measured at BER=10^{-5} without early-termination at 1.2V supply for rate 1/2

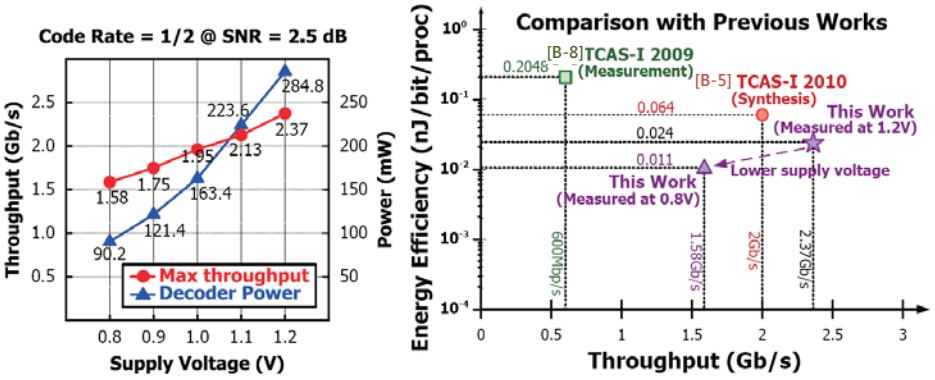


Fig. 17. Measurement Results and the comparison with previous works.

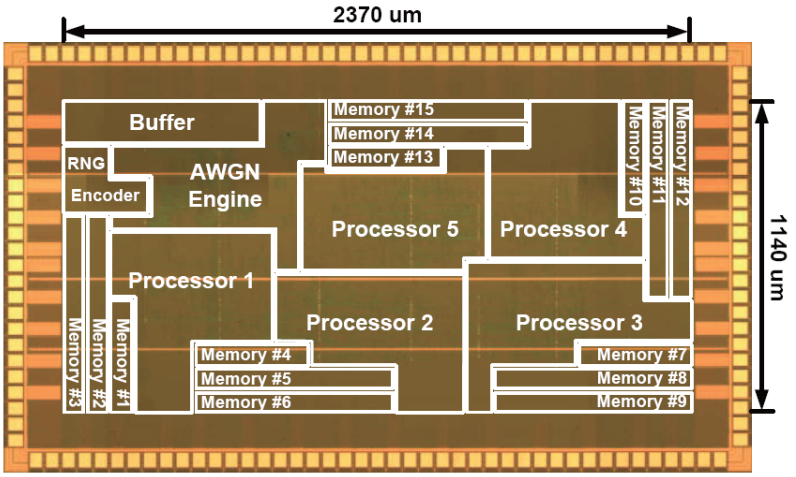


Fig. 18. Chip micrograph.

B. BCH and RS Decoder

1. Soft BCH Decoder Chip for DVB-S2 System

a) Soft Decision BCH Decoding

Conventional BCH decoding contains syndromes calculation, key equation solver, and Chien search [B-1]. In general, the complexity of a soft BCH decoder is much higher than a hard BCH decoder for decoding an entire codeword. Nevertheless, soft BCH decoders with lower complexity can be revealed by focusing on the least reliable bits instead of the whole codeword.

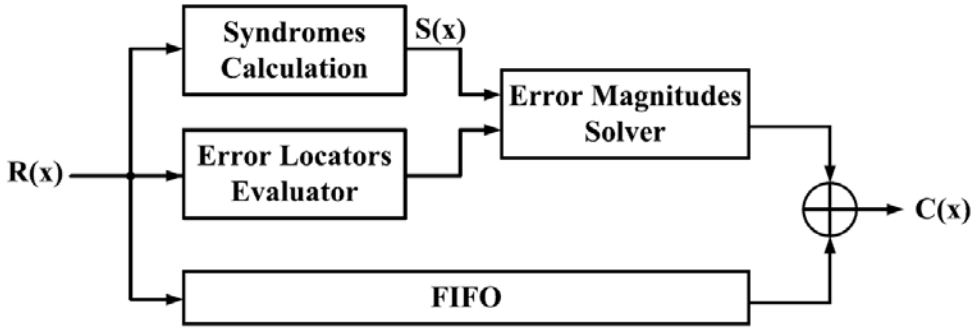


Fig. 19. Soft Decision BCH Decoding Block Diagram

As shown in Fig. 19, the soft BCH decoding using error magnitudes [B-1] includes three major steps: syndromes calculation, error locators evaluator, and error magnitudes solver. From the received polynomial R(x), the syndromes polynomial $S(x) = S_1 + S_2x^1 + \cdots + S_{2t}x^{2t-1}$ are expressed as

$$S_j = R(\alpha^j) = \sum_{i=1}^{v}(\alpha^j)^{l_i} = \sum_{i=1}^{v}(\beta_{l_i})^j \qquad (1)$$

For $j = 1, 2, \cdots, 2t$, where α is the primitive element over $GF(2^m)$. Notice that li is the i-th actual error location and $\beta_{li} = \alpha^{li}$ indicates the corresponding error locator. With soft inputs, error locators evaluator can choose 2t least reliable inputs and evaluates their corresponding error locator values to form a β-vector, $[\beta_{c1}, \beta_{c2}, \ldots, \beta_{c2t}]$. Also, the error location set, $\{L_{c1}, L_{c2}, \ldots, L_{c2t}\}$, can be calculated with β- vector because the β value of the L_{ci} -th location is $\beta_{ci} = \alpha L_{ci}$. The relation between β_{ci} and the syndromes can be formulated as

$$\begin{bmatrix} \beta_{c_1} & \beta_{c_2} & \cdots & \beta_{c_{2t}} \\ \beta_{c_1}^2 & \beta_{c_2}^2 & \cdots & \beta_{c_{2t}}^2 \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{c_1}^{2t} & \beta_{c_2}^{2t} & \cdots & \beta_{c_{2t}}^{2t} \end{bmatrix} \begin{bmatrix} \gamma_{c_1} \\ \gamma_{c_2} \\ \vdots \\ \gamma_{c_{2t}} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{2t} \end{bmatrix} \quad (2)$$

where γ_{ci} is the error magnitude corresponding to β_{ci} for i = 1, 2, . . . , 2t. The left 2t × 2t matrix in (2) is defined as β - matrix. From (1) and (2), it is evident that if all the errors are in the location set, the exact γ_{ci} value can be determinated; otherwise, this decoding approach fails to correct errors. The error magnitudes solver shown in Fig. 19 is used to solve (2) to get γ_{ci}. For those γ_{ci} equal to 1, the corresponding L_{ci} are the exact error locations. The codeword polynomial C(x) can be obtained by inversing the L_{ci} -th values in the received polynomial R(x).

b) Proposed Algorithm and Architecture

(1) **Error Locators Evaluator**

As shown in Fig. 20, error locators evaluator architecture includes the reliability part, the error locator part and the error location part. The upper part is the reliability part which stores the reliabilities of 2t least reliable candidates R_{c1} ,R_{c2} , . . . ,R_{c2t} . The medium part is the error locator part to construct the β-vector. Because the β value of the i-th location is α^i, the β value of (i+1)-th locations is α times the β values of i-th location. The β value can be computed by multiplying α^{-1} with register REG if the input is serial in from the highest degree coefficient of R(x). Thus, the error locator part can use a constant multiplier to calculate the error locator of each input. Notice that register REG initially contains the β value of the first input. The bottom part is the error location part. The decoding method focuses on the least reliable bits instead of the whole codeword, so the error location part uses a counter to compute the error location L_{ci} corresponding to each R_{ci} for serial input. Hence, the Chien search procedure is no longer required and a lot of redundant decoding latencies can be eliminated.

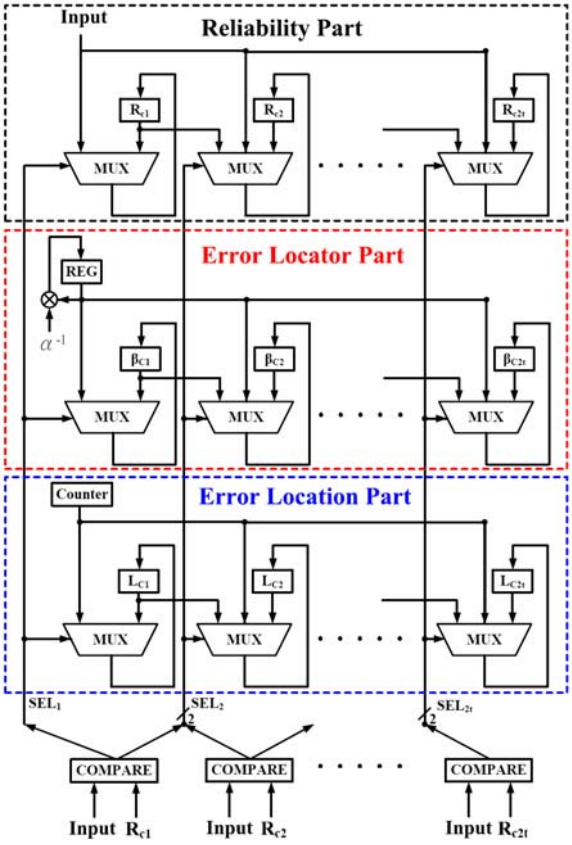


Fig. 20. Error Locators Evaluator Architecture for Serial Input

Error locators evaluator classifies the soft inputs to choose 2t least reliable inputs as the candidate reliabilities R_{c1} ,R_{c2} , . . . ,R_{c2t} . Their corresponding error locators β_{ci} and error locations L_{ci} are also calculated and stored in registers. Error locators evaluator compares the soft inputs with R_{ci} , and then generate the select signals SEL_i to control the multiplexers. In the i-th stage, if the input is smaller than R_{ci-1} , the i-th stage is updated with (i-1)-th stage value. If the input is greater than R_{ci-1} and smaller than R_{ci} , the i-th stage is updated with the input value. Otherwise, the i-th stage holds its current value.

(2) Error Magnitudes Solver (EMS)

To obtain the valid γ_{ci} value in (2), the *Gauss Elimination* method is the most intuitive way but the complexity is O(n3). Two alternative algorithms for improving decoding efficiency under different error correcting capabilities t are revealed. One uses the characteristic that the valid error magnitude in BCH codes is either 0 or 1, and the other employs the quick Vandermonde matrix solution.

(a) Heuristic Error Magnitudes Solver(H-EMS):

In BCH codes, the valid error magnitude in (2) is either 0 or 1, so the problem can be formulated into checking all combinations of γ_{ci} over GF(2) instead of calculating real error magnitudes. A 2t-bit counter is used to do a heuristic search for all binary combinations. Since $S_1^2 = S_2$, $S_2^2 = S_4$, . . . , $S_t^2 = S_{2t}$ in BCH codes, the even part of syndromes check can be eliminated to simplify (2) as :

$$\begin{bmatrix} \beta_{c_1} & \beta_{c_2} & \cdots & \beta_{c_{2t}} \\ \beta_{c_1}^3 & \beta_{c_2}^3 & \cdots & \beta_{c_{2t}}^3 \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{c_1}^{2t-1} & \beta_{c_2}^{2t-1} & \cdots & \beta_{c_{2t}}^{2t-1} \end{bmatrix} \begin{bmatrix} \gamma_{c_1} \\ \gamma_{c_2} \\ \vdots \\ \gamma_{c_{2t}} \end{bmatrix} \begin{bmatrix} S_1 \\ S_3 \\ \vdots \\ S_{2t-1} \end{bmatrix} \quad (3)$$

Table 3

The Proposed Heuristic EMS Algorithm

Input : *a* $2t-bit\ CNT : \{CNT_1, CNT_2, \ldots, CNT_{2t}\} = 0$
syndromes vector $[S_1, S_3, \ldots, S_{2t-1}]$, *and* β_{c_i} *matrix*
for $(k = 1; k < 2^{2t}; k{+}{+})$
  for $(j = 1; j < 2t;$ j=j+2)
    $\delta_j = \sum_{i=1}^{2t} CNT_i \times \beta_{c_i}^j$
  if $(all\ \delta_j == S_j)$
    Decoding Successes !!!
  else
    $CNT = CNT + 1$

Table 3 illustrates details of the proposed H-EMS algorithm. The i-th bit of CNT, CNT_i, performs as the i-th error magnitude, γ_{ci}. Thus, by iteratively flipping each CNT_i value, a heuristic search for all binary combinations can be completed. At each iteration, the solver can verify where the equation (3) stands or not. As shown in Fig. 21, H-EMS uses 2t(t-1) multipliers to construct the β-matrix. Each β_{ci}^j value will be calculated with CNT_i, and the solver checks the results equal to the syndromes or not.

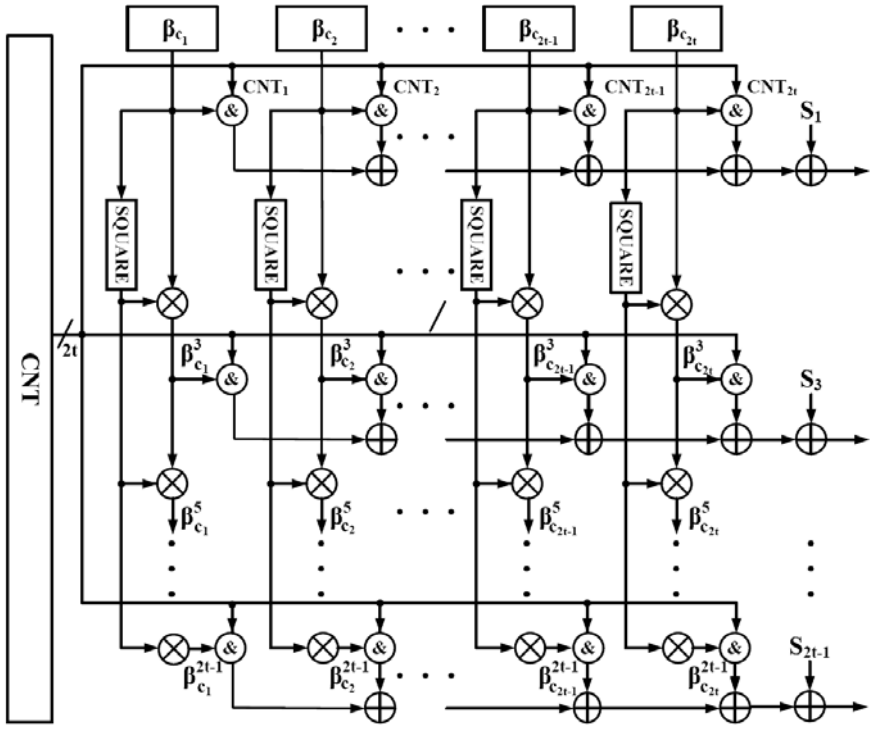


Fig. 21. Heuristic Error Magnitudes Solver Architecture

(b) Borck-Pereyra Error Magnitudes Solver(BP-EMS)

Table 4

Borck-Pereyra Algorithm

$$
\begin{array}{l}
Input\ :\ syndromes\ vector\ \ [S_1, S_2, \ldots, S_{2t}] \\
\qquad \beta_{c_i}\ vector\ [\beta_{c_1}, \beta_{c_2}, \ldots, \beta_{c_{2t}}] \\
\text{for } (k = 1; k < 2t, k{+}{+}) \\
\qquad \text{for } (i = 2t; i > k, i\text{- -}) \\
\qquad\qquad S_i = S_i - \beta_{c_k} S_{i-1} \\
\text{for } (k = 2t-1; k > 0, k\text{- -}) \\
\qquad \text{for } (i = k+1; i \le 2t, i{+}{+}) \\
\qquad\qquad S_i = S_i / (\beta_{c_i} - \beta_{c_{i-k}}) \\
\qquad \text{for } (i = k; i < 2t, i{+}{+}) \\
\qquad\qquad S_i = S_i - S_{i+1} \\
\text{for } (k = 1; k \le 2t, k{+}{+}) \\
\qquad S_i = S_k / \beta_{c_k} \\
Output : S_i\ is\ error\ magnitude
\end{array}
$$

Since the β_{ci} matrix is a Vandermonde matrix, Borck-Pereyra algorithm [B-9][B-10] shown in Table 4 can calculate the error magnitudes efficiently for large matrix. In Borck-Pereyra algorithm, the variable S_i which initially contains the ith syndrome value is updated iteratively. In stead of using β - matrix to compute (2), Borck-Pereyra algorithm uses β - vector to reduce the

implementation complexity. After all computations, S_i indicates the i-th error magnitude. From Table 4, Borck-Pereyra algorithm has division, multiplication and addition operations. Notice that the multiplier can be shared if the divider can be decomposed into an inversion and a multiplier. Thus, as shown in Fig. 22, BPEMS only contains 1 multiplier, 1 inversion, 3 adders and a control logic. The control logic determines the computation order of the syndromes and β_{ci}, and the computation results will be used to update each S_i value. The inversion in the proposed architecture is carried out in composite field because the finite field inversion over GF(2^m) is costly and infeasible with table-lookup implementation for large m.

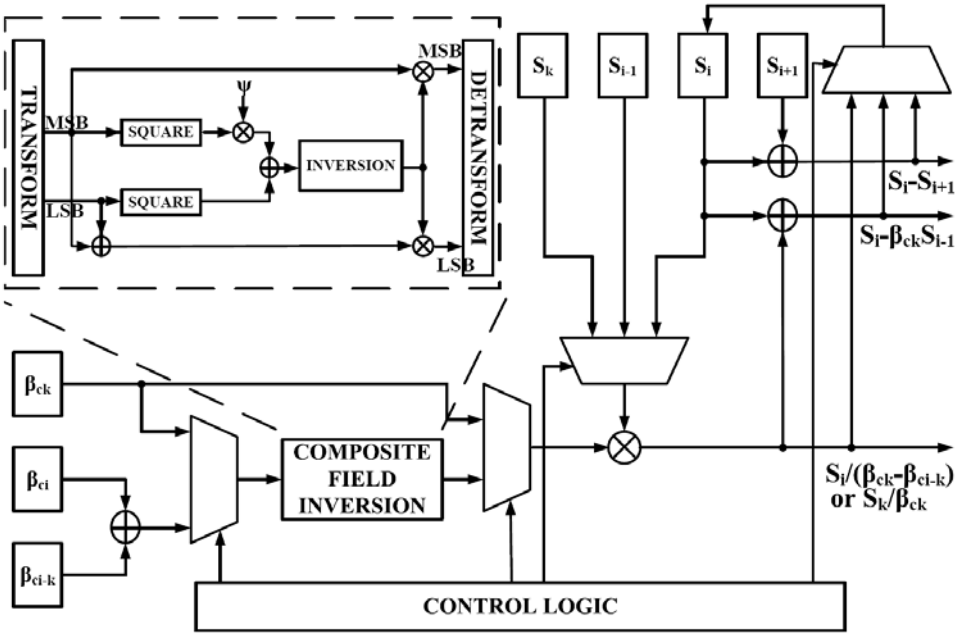


Fig. 22. Borck-Pereyra Error Magnitudes Solver Architecture

Composite field [B-11] is viewed as an extension field of GF(2^k) while given m = kr. The finite field GF(2^m) can be constructed by coefficients from the subfield GF(2^k). Operating in subfield leads to lower implementation complexity and better computation efficiency. For example, every element in GF(2^{16}) can be represented by bx+c and inversion of bx+c can be derived as (4) with the polynomial $x^2 + x + \phi$ [B-11], where b and c are over GF(2^8).

$$\frac{1}{bx+c} = (b^2\psi + bc + c^2)^{-1}(bx + b + c) \qquad (4)$$

The composite field inversion over GF(2^{16}) is only 2.1K gate count in CMOS 90nm technology while the inversion using Look Up Table method is about 186K gate count.

c) *Simulation and Implementation Result in DMB-T and DVB-S2 Systems*

In DMB-T and DVB-S2 systems, BCH (762,752) over GF(2^{10}) and BCH (32400, 32208) over GF(2^{16}) are defined to be concatenated with LDPC codes respectively. Fig. 23 shows the simulation results for DMB-T system with LDPC (7493, 3048) at 20 iterations. Similarly, the simulation results for DVB-S2 system with LDPC (64800, 32400) at 50 iterations is shown in Fig. 24. The proposed soft BCH decoders have 0.5db gain in DMB-T and similar performance in DVB-S2 at BER = 10^{-5}. These two BCH codes are implemented and demonstrated in

Table 5. Each BCH code is implemented in both hard decision and soft decision methods.

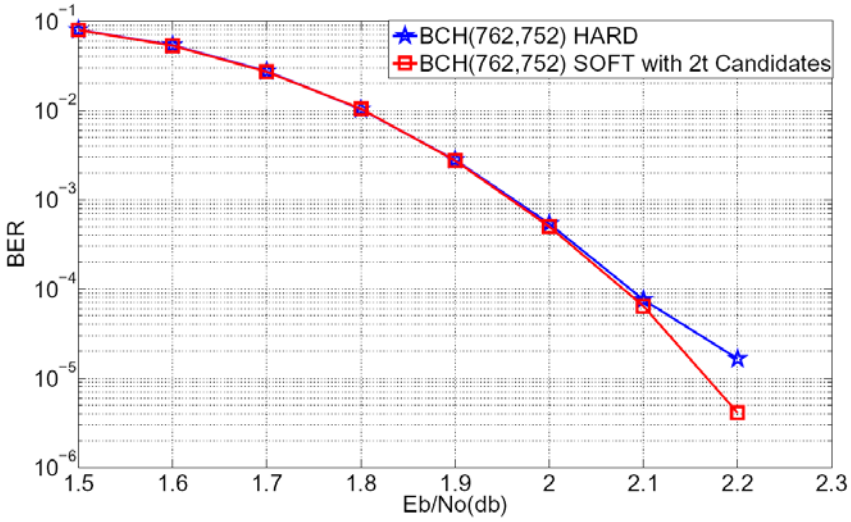


Fig. 23. Simulation results for BCH (762, 752, 1) in DMB-T system under BPSK modulation and AWGN channel

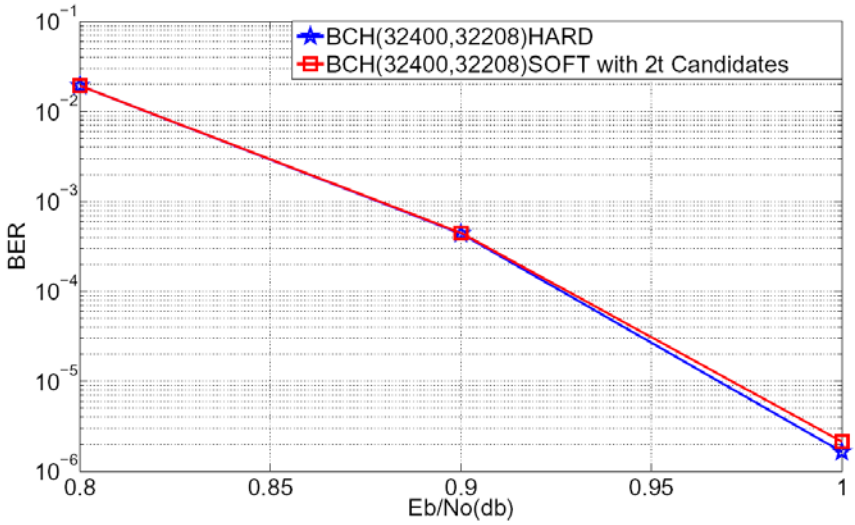


Fig. 24. Simulation results for BCH (32400, 32208, 12) in DVB-S2 system under QPSK modulation and AWGN channel

Table 5

Summary of Implementation Results

	Hard BCH (762,752), t = 1	Soft BCH (762,752), t = 1	Hard BCH (32400,32208), t = 12	Soft BCH (32400,32208), t = 12
Technology	90nm	90nm	90nm	90nm
Architecture	Look Up Table	H-EMS	iBM + Chien Search	BP-EMS
Operation Frequency	500MHz (Post Layout)	500MHz (Post Layout)	200MHz (Post Layout)	333MHz (Measurement)
Core Area	$14400\mu m^2$	$4225\mu m^2$	$190497\mu m^2$	$102400\mu m^2$
Gate Count	3.1K	1.2K	54.0K	26.9K
Latency	763	763	64824	34104
Throughput	492.8Mbps	492.8Mbps	99.3Mbps	314.5Mbps

For BCH (762,752), key equation procedure is not needed due to t = 1. To eliminate Chien search, the hard BCH decoder uses look up table method to solve the error location, and the soft BCH decoder uses the H-EMS architecture. Calculating all the combination values at one cycle, the gate-count of the soft BCH decoder is only 38.8% of the hard BCH decoder under the same latency and operation frequency. For BCH (32400, 32208) with t = 12, the hard BCH decoder uses iBM algorithm to solve key equation and needs Chien search to get error locations. By inserting registers in composite field inversion, the operation frequency of the soft BCH decoder with BP-EMS is enhanced from 166MHz to 333MHz with only 2.5% latency increment in overall decoding procedure. Computing error locations without Chien search, the soft BCH decoder has almost half latencies of the hard BCH decoder. Hence, the soft BCH decoder has much better throughputs than the hard BCH decoder. The measurement result reveals that the soft BCH decoder saves 50.0% gate-count and 47.4% clock cycle latency as compared with the hard BCH decoder. Fig. 25 is the chip microphoto of soft BCH (32400, 32208).

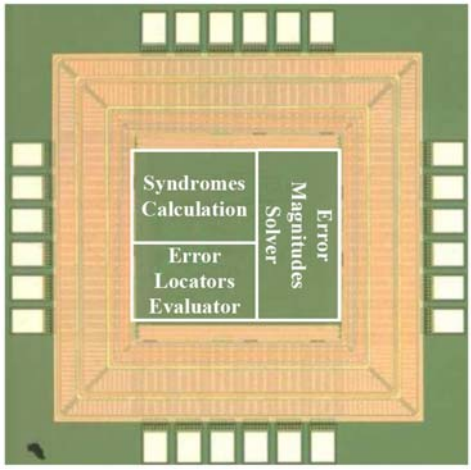


Fig. 25. Microphoto of Soft BCH(32400,32208) Chip

2. An Improved Soft BCH Decoder with One Extra Error Compensation

a) PROPOSED COMPENSATION SOFT BCH DECODING

The proposed soft BCH decoder shown in Fig. 26 includes three major steps: syndrome calculator, error locator evaluator, and compensation error magnitude solver. From the received polynomial R(x), the syndrome polynomial $S(x) = S_1 + S_2x^1 + \cdots + S_{2t}x^{2t-1}$ is expressed as

$$S_j = R(\alpha^j) = \sum_{i=1}^{v}(\alpha^j)^{e_i} = \sum_{i=1}^{v}(\beta_{e_i})^j \tag{8}$$

for $j = 1, 2, \cdots, 2t$, where α is the primitive element over $GF(2^m)$. Notice that ei is the i-th actual error location and $\beta_{ei} = \alpha_{ei}$ indicates the corresponding error locator.

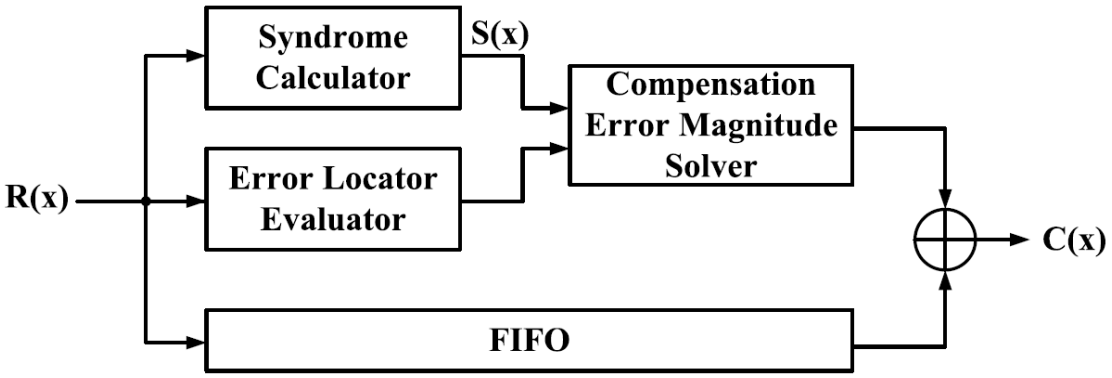


Fig. 26. Soft Decision BCH Decoding Block Diagram

With soft inputs, error locator evaluator can choose 2t least reliable inputs and evaluate their corresponding error locators to form the error locator set $B = [\beta_{l1}, \beta_{l2}, \ldots, \beta_{l2t}]T$. Also, the error location set, $L = [l1, l2, \ldots, l2t]T$, can be calculated with B because the error locator of the li-th location is $\beta_{li} = \alpha_{li}$. The relation between B and the syndrome vector, $S = [S_1, S_2, \ldots, S_{2t}]T$, can be formulated as

$$\begin{bmatrix} \beta_{l_1} & \beta_{l_2} & \cdots & \beta_{l_{2t}} \\ \beta_{l_1}^2 & \beta_{l_2}^2 & \cdots & \beta_{l_{2t}}^2 \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{l_1}^{2t} & \beta_{l_2}^{2t} & \cdots & \beta_{l_{2t}}^{2t} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{2t} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{2t} \end{bmatrix} \tag{9}$$

where $\Gamma = [\gamma_1, \gamma_2, \ldots, \gamma_{2t}]T$ is the error magnitude set corresponding to B, and the $2t \times 2t$ matrix in (9) is defined as β-matrix B. Let $\Delta = [\delta_1, \delta_2, \ldots, \delta_{2t}]T$ be defined as

$$\underline{\Delta} = \mathbf{B} \times \underline{\Gamma} + \underline{S} \tag{10}$$

From (8) and (9), it is evident that if all the errors are in the error location set, the exact γi value can be determinated and Δ will be all zero; otherwise, this decoding approach fails to correct errors. There are at most 2t error locations can be determined. However, it is very likely that only one error outside L but the decoder can't solve any error. To improve the error correcting ability, we additionally check whether Δ is a geometrical sequence or not to make a compensation for an error location outside L. A geometrical sequence $\Delta = [\beta_{lloss}, \beta_{lloss2}, \ldots, \beta_{lloss2t}]$ means an error location loss can be found, where $\beta_{lloss} = \alpha_{lloss}$. For example, if there are four errors in 1st, 3rd, 5th and 9th locations for a BCH (255,239) decoder which can correct 2 errors, S is expressed as

$$\underline{S} = \begin{bmatrix} \beta_1 + \beta_3 + \beta_5 + \beta_9 \\ \beta_1^2 + \beta_3^2 + \beta_5^2 + \beta_9^2 \\ \beta_1^3 + \beta_3^3 + \beta_5^3 + \beta_9{}^3 \\ \beta_1^4 + \beta_3^4 + \beta_5^4 + \beta_9^4 \end{bmatrix}^T \tag{11}$$

In the case that the decoder collects B = [β1, β3, β6, β9], and Γ = [1, 1, 0, 1], Δ becomes

$$\begin{aligned} \underline{\Delta} &= \begin{bmatrix} \beta_1 & \beta_3 & \beta_6 & \beta_9 \\ \beta_1^2 & \beta_3^2 & \beta_6^2 & \beta_9^2 \\ \beta_1^3 & \beta_3^3 & \beta_6^3 & \beta_9^3 \\ \beta_1^4 & \beta_3^4 & \beta_6^4 & \beta_9^4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \beta_1 + \beta_3 + \beta_5 + \beta_9 \\ \beta_1^2 + \beta_3^2 + \beta_5^2 + \beta_9^2 \\ \beta_1^3 + \beta_3^3 + \beta_5^3 + \beta_9^3 \\ \beta_1^4 + \beta_3^4 + \beta_5^4 + \beta_9^4 \end{bmatrix} \\ &= [\beta_5, \beta_5^2, \beta_5^3, \beta_5^4] \end{aligned} \tag{12}$$

Then not only errors at 1-st, 3-rd and 9-th locations but also an error at 5-th location can be corrected. Therefore, the proposed compensation soft BCH decoder can correct at most 2t+1 error. The compensation error magnitude solver (CEMS) shown in Fig. 26 is used to solve (9) and (10) to get Γ and Δ. For those γ_i equal to 1, the corresponding li and lloss are the exact error locations. The codeword polynomial C(x) can be obtained by inversing values at error locations in the received polynomial R(x).

To obtain the γ_i value in (9), the Gauss Elimination method is the most intuitive way but the complexity is O(n3). In BCH codes, the valid error magnitude in (9) is either 0 or 1, so the problem can be formulated into checking all combinations of γ_i over GF(2) instead of calculating real error magnitudes. A 2t-bit counter is used to do a heuristic search for all binary combinations. Since S12 = S2, S22 = S4, . . . , St2 = S2t in BCH codes, the even part of syndromes check can be eliminated to simplify (9) as :

$$\begin{bmatrix} \beta_{l_1} & \beta_{l_2} & \cdots & \beta_{l_{2t}} \\ \beta_{l_1}^3 & \beta_{l_2}^3 & \cdots & \beta_{l_{2t}}^3 \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{l_1}^{2t-1} & \beta_{l_2}^{2t-1} & \cdots & \beta_{l_{2t}}^{2t-1} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{2t-1} \\ \gamma_{2t} \end{bmatrix} = \begin{bmatrix} \mathrm{S}_1 \\ \mathrm{S}_3 \\ \vdots \\ \mathrm{S}_{2t-1} \end{bmatrix} \quad (13)$$

The complexity can be significantly reduced for only half size matrix, Bodd and Sodd, used in (13).

Following steps illustrate the details of the efficient Implementation of CEMS.

Input : $\underline{\mathcal{B}}$, $\underline{S_{odd}}$ and $\underline{\Gamma} = 0$
1) Construct the β-matrix $\mathbf{B_{odd}}$ with $\underline{\mathcal{B}}$
2) $\underline{\Delta_{odd}} = \mathbf{B_{odd}} \times \underline{\Gamma} + \underline{S_{odd}}$
3) if $\underline{\Delta_{odd}}$ is a geometrical sequence
Go to 4)
else
if $\underline{\Gamma} == 2^{2t} - 1$
Failed Decoding
else
$\underline{\Gamma} = \underline{\Gamma} + 1$
Go to 2)
4) Find l_{loss} with $\underline{\Delta_{odd}}$
Output : $\underline{\Gamma}$ and L_{loss}

By iteratively counting Γ value, a heuristic search for all binary combinations can be completed. At each iteration, the solver can verify whether the geometrical sequence check stands or not.

b) VLSI ARCHITECTURE FOR THE COMPENSATION SOFT BCH DECODER

(1) Error Locator Evaluator

As shown in Fig. 27, error locator evaluator architecture includes the reliability part, the error locator part and the error location part. The upper part is the reliability part which stores the reliabilities of 2t least reliable candidates R_{l1},R_{l2}, . . .,R_{l2t} . The medium part is the error locator part to construct the error locator set B. Because the error locator of the i-th location is αi, the error locator of (i+1)-th locations is α times the error locator of i-th location. The error locator can be computed by multiplying α−1 with register REG if the input is serial in from the highest degree coefficient of R(x).

Thus, the error locator part can use a constant multiplier to calculate the error locator of each input. Notice that register REG initially contains the error locator of the first input. The bottom part is the error location part. The decoding method focuses on the least reliable bits instead of the whole codeword, so the error location part uses a counter to compute the error location li corresponding to each Rli for serial input. Hence, the Chien search procedure is no longer required and a lot of redundant decoding latencies can be eliminated.

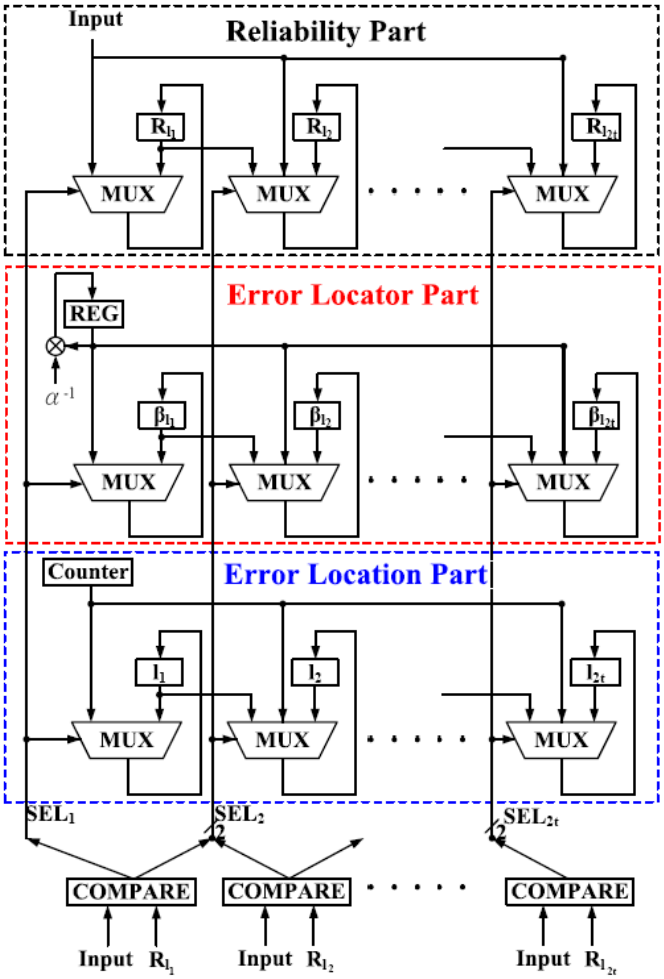


Fig. 27. Error Locator Evaluator Architecture for Serial Input

Error locator evaluator classifies the soft inputs to choose 2t least reliable inputs as the candidate reliabilities R_{l1},R_{l2}, . . . , R_{l2t} . Their corresponding error locators βli and error locations li are also calculated and stored in registers. Error locator evaluator compares the soft inputs with R_{li}, and then generates the select signals SELi to control the multiplexers. In the i-th stage, if the input is smaller than R_{li}−1 , the i-th stage value is updated with (i-1)-th stage value. If the input is greater than R_{li}−1 and smaller than R_{li}, the i-th stage value is updated with the input value. Otherwise, the i-th stage holds its current value.

(2) Compensation Error Magnitude Solver (CEMS)

The compensation error magnitude solver (CEMS) in Fig. 28 is employed to evaluate (13) while given Sodd and B. Totally 2t2 registers are used to store each entry in the Bodd matrix. The

initial value of registers in each row is set as B so that the output of the SQUARE will always be βli2 for first t-1 cycles. Iteratively multiplied byβli2, the bottom registers generate β_{li}^{2j+1} for i = 1 ~ 2t and j = 0 ~ t-1. Thus, totally only 2t multipliers are used for Bodd calculation. After t-1 cycles, Bodd is constructed and the registers will stop update. Matrix multiplication is evaluated in the following 22t cycles. By counting Γ value, a heuristic search for all binary combinations can be completed. At each iteration, each β_{lij} value will be calculated with γ_i, and the solver can verify whether the geometrical sequence check stands or not. If Δodd is a geometrical sequence, then $\delta_i \times \delta_{12} = \delta_{i+2}$. CEMS uses t multipliers to check the relation and uses a look up table (LUT) for looking for lloss from δ1.

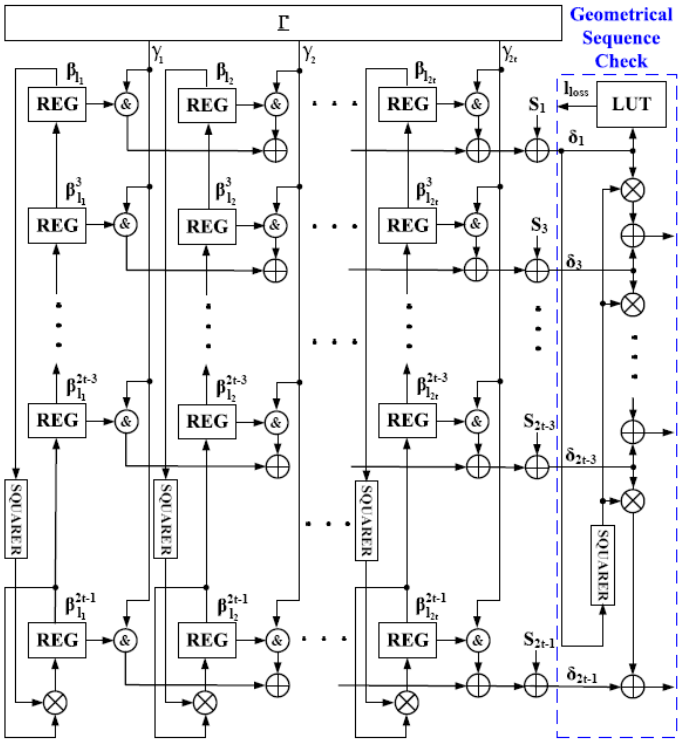


Fig. 28. Compensation Error Magnitude Solver Architecture

(3) Architecture Comparison

The architectures of a hard BCH decoder and the proposed soft BCH decoder are compared in Table 6. In finite field operation, the complexity of a multiplier is much higher than a register. Because of fewer multipliers, the proposed soft BCH decoder with more registers and additional LUT has similar hardware complexity as the hard BCH decoder with inversionless

Berlekamp-Massey (iBM) algorithm [B-15] Moreover, searching error locations at error locator evaluator procedure leads to a lot of latency saving. Therefore, the proposed soft BCH decoder can provide higher throughput with almost the same hardware complexity as compared to the traditional hard BCH decoder. For example, for BCH (255,239) code, the proposed soft BCH decoder has 20 registers, 1 LUT and 5 multipliers while the hard BCH decoder has 12 registers and 9 multipliers. Furthermore, the proposed decoder also has only 53% latency as compared with traditional hard BCH decoder.

Table 6
Comparison Table for A (n, k, t) BCH Code

	(n,k,t) Hard BCH with iBM	(n,k,t) Soft BCH Proposed	(255,239,2) Hard BCH with iBM	(255,239,2) Soft BCH Proposed
register	$5t+2$	$2t^2+6t$	12	20
multiplier	$3t+3$	$3t-1$	9	5
constant multiplier	$3t$	$t+1$	6	3
square	0	$2t+1$	0	5
LUT	0	1	0	1
latency	$2n+2t$	$n+2^{2t}+t-1$	514	272

c) SIMULATION AND IMPLEMENTATION RESULTS

Simulation and implementation results for our proposed soft BCH decoder are presented in this section. Fig. 29 shows the performance comparison for 2-error-correcting (255,239) BCH code under BPSK modulation in AWGN channel. The achieved coding gain is about 0.75dB over the hard BCH decoder at BER = 10^{-5}. Our proposed decoder can outperform 0.35dB and 0.2dB coding gain as compared with GMD [B-4] and sub-optimum MAP [B-7] respectively.

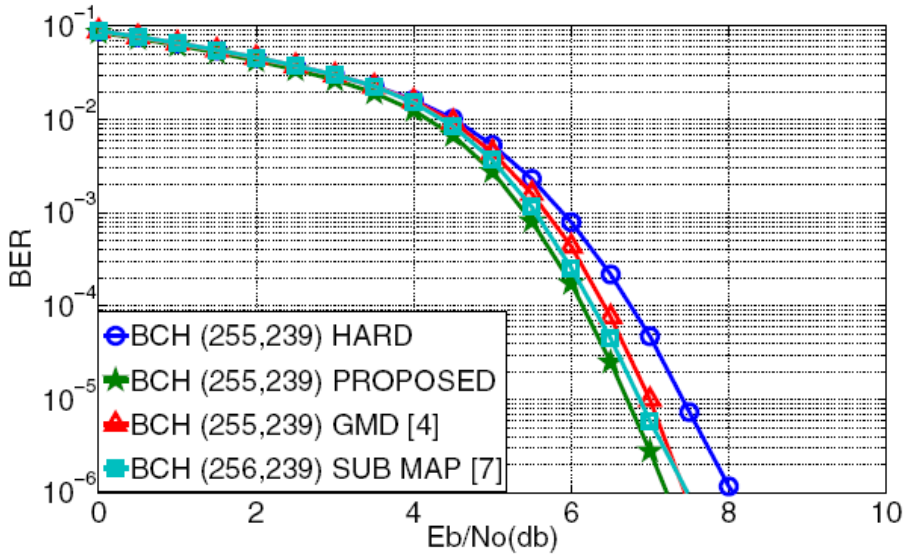


Fig. 29. Simulation results for BCH (255,239) code

The BCH (255,239) decoder is implemented with hard decision and soft decision methods and demonstrated in Table 7. The hard BCH decoder uses iBM algorithm to solve key equation and needs Chien search to get error locations. Computing error locations without Chien search, the soft BCH decoder has almost half latency of the hard BCH decoder. Hence, the soft BCH decoder has much better throughputs than the hard BCH decoder. According to the post-layout simulations, the soft BCH decoder saves 47.1% clock cycle latency with similar gate count and operation frequency as compared with the hard BCH decoder in standard CMOS 90nm technology.

Table 7

Summary of Implementation Results

	Hard BCH (255,239) t = 2	Soft BCH (255,239), t = 2
Technology	90nm	90nm
Architecture	iBM + Chien Search	CEMS w/o Chien Search
Operation Frequency	360MHz (Post Layout)	360MHz (Post Layout)
Core Area	$14400\mu m^2$	$13225\mu m^2$
Gate Count	4.38K	4.06K
Latency	514	272
Throughput	167.4Mb/s	316.3Mb/s

3. Soft RS Decoder Chip for Optical Communication System

a) Proposed Soft RS Decoding Algorithm

First of all, based on the received soft information, η least reliable positions (LRPs), [l0, l1, ..., lη−1], are defined and S(x) is calculated simultaneously. The candidate sequences are generated according to Gray code based bit flipping method, leading to only one bit of these LRPs flipped between each successive candidate. As a result, S(i+1)(x) for the (i + 1)-th candidate can be updated with the method in step 2. of Algorithm 1. S(i+1) j is the j-th coefficient of S(i+1)(x) and e'k ×αlk×j is the compensation value, which can be viewed as the error pattern induced by the bit-flipping operation of k-th LRP. After updating the syndrome S(i+1)(x) and calculating the corresponding Λ(i+1)(x), we set a condition that only the Λ(i+1)(x) with degree less than t will be sent to Chien search to find the error locations because it's highly possible for the Λ(i+1)(x) to be in the limit of correction capability. If the condition is met, the candidate sequence will be decoded as the output message and the decoding procedure will be terminated. Otherwise, next candidate will be generated to repeat above-mentioned steps. If no one meets the condition among all 2η−1 candidates, the received signal will be decoded without the condition, and the error correction capability as hard RS decoders is guaranteed.

Algorithm 1. : Decision-Confined Algorithm
Input : $R(x)$ *and* η*-bit integers* $\gamma = \gamma' = 0$.
step 1.
Calculate syndrome $S(x)$.
Evaluate η *LRPs,* $L = [l_0, l_1, ..., l_{j-1}]$, *and corresponding error values,* $E' = [e'_1, e'_2 \ldots, e'_\eta]$.
step 2.
for $i = 0, i < 2^\eta - 1, i = i + 1$:
$\gamma' = \gamma$, $\gamma = i \oplus (i >> 1)$ *(Gray code)*
Find the bit different between γ *and* γ', *k-th bit*
Update syndrome $S^{(i+1)}(x)$:
$S_j^{(i+1)} = S_j^{(i)} + e'_k \times \alpha^{l_k \times j}$, $1 \le j \le 2t$.
Calculate $\Lambda^{(i+1)}(x)$ *from KES with* $S^{(i+1)}(x)$.
if $(deg(\Lambda^{(i+1)}(x)) < t)$
go to **step 4.**
else if $(i = 2^\eta - 2$ *and* $deg(\Lambda^{(i+1)}(x)) = t)$
go to **step 3.**
end for
step 3.
Calculate $\Lambda(x)$ *with* $S(x)$.
step 4.
Find error locations and evaluate error values to obtain $e(x)$.
Output : $\hat{C}(x) = R(x) \oplus e(x)$.
End

b) VLSI Architecture for Soft RS Decoder

For the 2.5 Gb/s requirement of the optical communication systems, a soft RS (255,239) decoder with three pipelined stages based on our decision-confined decoding algorithm is presented and the decoding scheme is shown in Fig. 30. The following subsections will show the unique parts of our proposal in contrast to conventional hard decoders.

(1) Syndrome Updater

According to the method in step 2 of Algorithm 1, the candidate syndrome S(i+1)(x) can be updated from S(i)(x) by utilizing a look-up table (LUT) instead of recalculating it with syndrome calculator for further cost efficiency. Note that there are at most 25 candidates for each received message and 259 computational cycles for each pipelined stage. Thus it has 8 computational cycles for every S(i+1)(x) and Λ(i+1)(x). As a result, the finite field multipliers (FFMs) and the squares can be shared to compute 16 compensation values for further hardware reduction. In our design, it only costs 4 FFMs and 2 squares for the calculation of all compensation values as shown in Fig. 31.

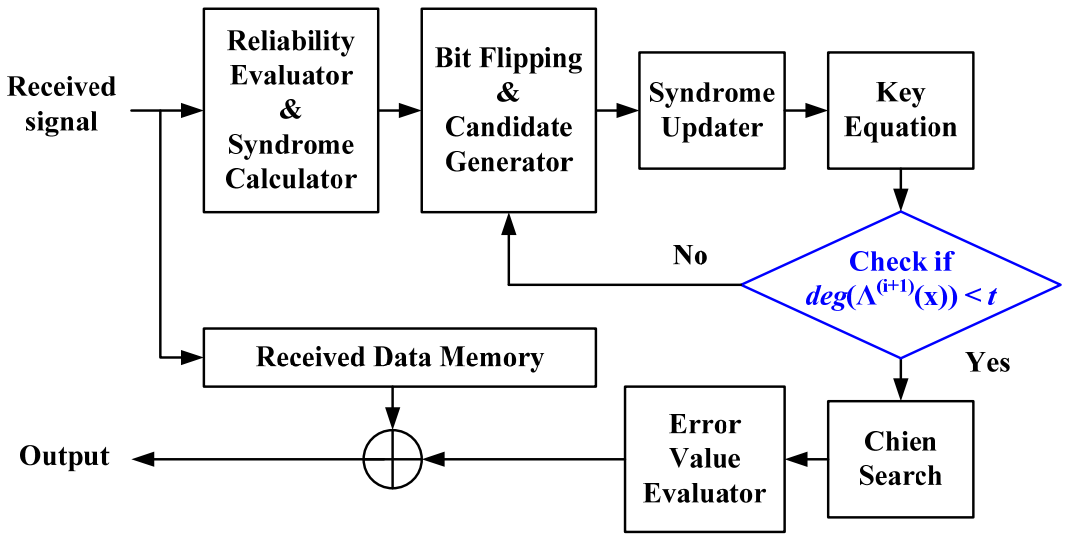


Fig. 30. Decoding scheme of the proposed soft RS decoder

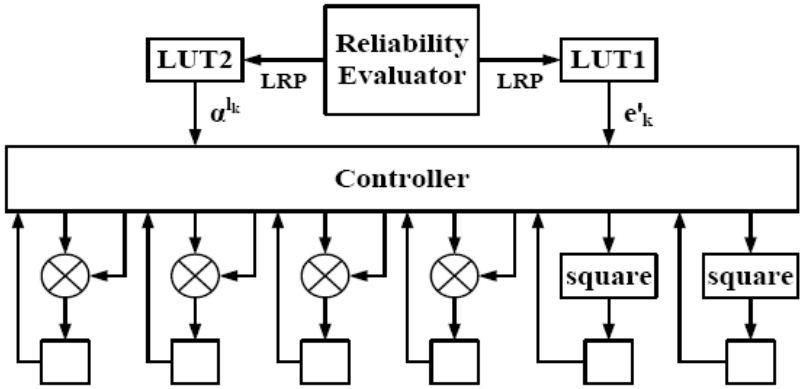


Fig. 31. Syndrome updater

(2) Half-iteration RiBM

The conventional KES needs 2t iterations to solve the key equation : $\Omega(x) = S(x) \times \Lambda(x) \bmod x^{2t}$. For RS (255,239), it will cost 16 cycles to calculate $\Lambda(x)$. Instead of using two KES to meet 8 cycles timing constraint, which results in high complexity and difficult signal controlling, we propose a half-iteration RiBM algorithm on the basis of [B-17] and [B-20] to shorten the latency of KES. Combining the advantages of homogeneous architecture and half computation latency, half-iteration RiBM can fully match our desire for KES. According to half-iteration RiBM algorithm, the structure of the processing element of half-iteration RiBM (H-PE) is depicted in Fig. 32 and the KES can be implemented with 2t + 1 H-PEs as illustrated in Fig. 33.

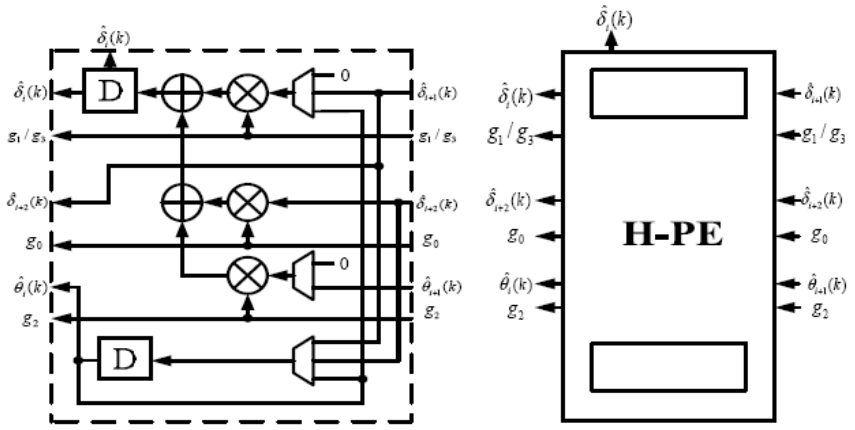


Fig. 32. The processing element of Half-iteration RiBM (H-PE)

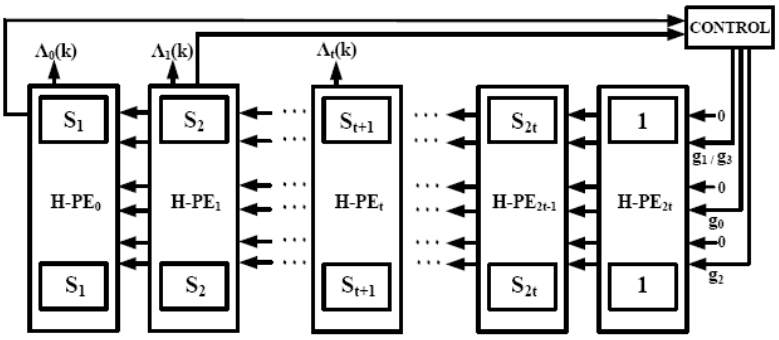


Fig. 33. The homogeneous architecture of Half-iteration RiBM

(3) BP-based Error Value Evaluator

Conventionally, after Chien search evaluates the error locators Xi's, the corresponding error values ei's can be calculated with Λ(x) and Ω(x) based on the Forney's algorithm. From another approach, the BP based method [B-18] can compute the error values by solving the Vandermonde relation between the syndrome Si's and error locators Xi's as following form.

$$\begin{bmatrix} X_1 & X_2 & \cdots & X_8 \\ X_1^2 & X_2^2 & \cdots & X_8^2 \\ \vdots & \vdots & \ddots & \vdots \\ X_1^8 & X_2^8 & \cdots & X_8^8 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_8 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_8 \end{bmatrix}$$

Since the Forney's algorithm and BP-based method consume nearly the same hardware costs, our half-iteration RiBM method removes the calculation of Ω(x) for further area efficiency. Based on the BP method, the error value evaluator can be implemented with the architecture as shown in Fig. 34.

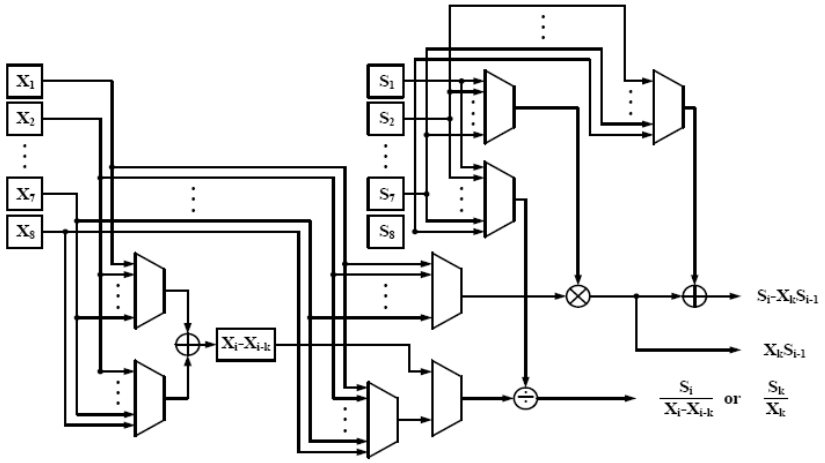


Fig. 34. BP-based error value evaluator

c) Simulation and Implementation Result

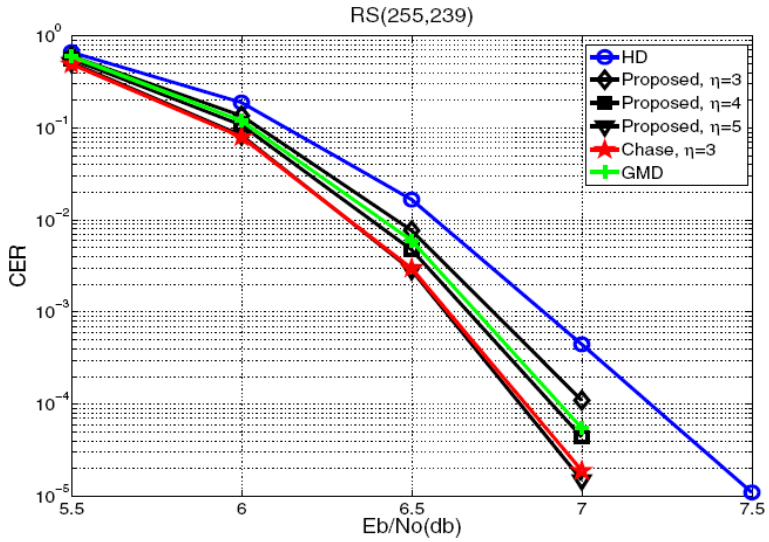


Fig. 35. Performance of the proposed soft decoding algorithm

Fig. 35 shows the RS (255,239) simulation results for our proposed algorithm with different η under BPSK modulation and AWGN channel. The performance gain at 10^{-4} CER is 0.4 dB with η = 5 over the hard decoding. Compared with Chase algorithm with η = 3, our proposed method can achieve competitive coding gain with η = 5. Although it needs more LRPs, the average computation complexity of our proposal is much less than Chase algorithm. For instance, at E_b/N_0 = 7, according to our approach with η = 5, the average computation times of syndrome updater, KES, Chien search and error value evaluator are 1.07, 1.07, 1 and 1 respectively. However, the Chase algorithm with η = 3 consumes 23 calculation for all the decoding blocks.

Table 8

COMPARISON OF TIME AND HARDWARE COMPLEXITY WITH SOFT RS DECODER

	Coding Gain @ 10^{-4} CER	$GF(2^8)$ Constant Multiplier	$GF(2^8)$ Variable Multiplier	$GF(2^8)$ Adder	Mux (bits)	ROM (bytes)	RAM (bytes)	Reg. (bits)	# of pipeline stage	Latency (each stage)	Normalized XOR Gate Counts
Proposed	0.4 dB ($\eta = 5$)	32	68	85	984	520	256×3	834	3	259	21315
LCC [E-4]	0.4 dB ($\eta = 3$)	8	66	108	1109	1024	68+256×8	1606	4	528	37915

* Normalized XOR gate counts. Constant multiplier : 20, Variable multiplier : 100, Mux & Memory cell : 1, Register : 3.

Table 9

COMPARISON OF TIME AND HARDWARE COMPLEXITY WITH HARD DECISION RS DECODER

	Proposed	pRiBM [E-5]	pDCME [E-6]	DCME [E-7]
Code Type	Soft RS (255, 239)	Hard RS (255, 239)	Hard RS (255, 239)	Hard RS (255, 239)
Technology	$90nm$	$90nm$	$0.13\mu m$	$0.25\mu m$
Operation Frequency	320MHz (Measurement)	690MHz (Synthesis)	660MHz (Synthesis)	200MHz (Synthesis)
Gate Count	45.3 K	43.6 K	53.2 K	42.2 K
Throughput	2.56 Gb/s	5.52 Gb/s	5.28 Gb/s	1.6 Gb/s
Coding Gain	0.4 dB @ 10^{-4} CER	-	-	-

Table 8 shows the comparison with LCC-based soft RS decoder. Our proposal can achieve more than 40% area reduction while the assumption is even not including the cost of decision making unit consumed in [B-19]. In addition, our design can operate with only half latency for each pipeline and less pipelined stages. Fig. 36 shows our decoder chip which is the first soft RS decoder chip in our understanding. Hence,

Table 9 illustrates the implementation results of our soft RS decoder with other hard RS decoders. Implemented in 90nm CMOS process, our chip with 45.3K gates is comparable with a conventional hard decoder. Moreover, it can fit well for 10- 40 Gb/s with 16 RS decoders in optical fiber systems and 2.5 Gb/s GPON applications with 0.4 dB coding gain over hard decoders at 10−4 CER.

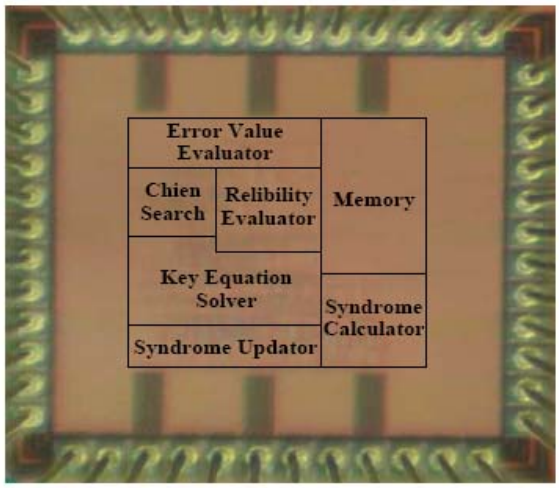


Fig. 36. Microphoto of soft RS (255,239) chip

C. Viterbi Decoder

1. A Low-Power Viterbi Decoder Based on Scarce State Transition and Variable Truncation Length

a) Power Reduction with Variable Truncation Length

As indicated in the Viterbi algorithm, the decoder output is the codeword that minimizes the conditional probability of the received sequence. Therefore, the entire received sequence should be stored and analyzed before any decoding output. Nevertheless, the received sequence length may be large, and the survivor path should be truncated to reduce storage requirement and decoding latency. If the truncation length T is large enough, about five times constraint lengths, the performance can achieve that of maximum-likelihood decoding.

Fig. 37 illustrates the survivor paths stored in a survivor memory. All the 2υ survivor paths will merge with a high probability for a 2υ-state Viterbi decoder. Consequently, it is more efficient to store the merged path rather than the 2υ paths after the merged stage. The truncation lengths depend strongly on the channel conditions as listed in Table 10. Based on the variable truncation length property [C-9], we design a path merging detection unit to reduce the power consumption in the survivor memory.

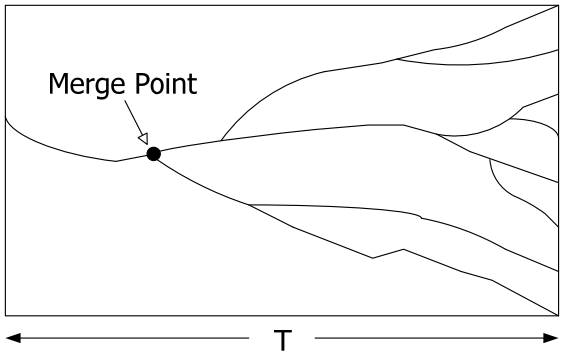


Fig. 37. Survivor paths stored in the survivor memory

Table 10
Average required truncation length for path merging in different channel condition

Eb/N_0	1.0	2.0	3.0	4.0	5.0
Truncation length	33.78	26.86	23.34	21.54	20.54

(1) VARIABLE TRUNCATION LENGTH

The Viterbi decoder for the MB-OFDM UWB system has 64 states. Fig. 38 illustrates the survivor memory on the radix-4 trellis. D0 to D63 are the decisions provided by the ACS units for selecting survivor paths. Base on path merging property, the 64 states tend to be equivalent from the left stages to the right stages, which are more reliable.

The path merging detection unit will find the merge point, or stage in the trellis. Obviously, if contents of all the 64 survivors are equivalent at the same stage, the 64 survivor paths have merged. However, it is complex to check all 64 states concurrently. To reduce the hardware complexity, our proposal detects path merging by dividing 64 states into 16 groups that are verified separately. The simulation results show that this scheme has no performance loss. We assume the 64 survivor paths have merged and the value in state 0 is already reliable if every group (the circles in Fig. 38) contains equivalent values at the same stage. After detecting the merged point, we apply clock gating to the registers in the shadow region and directly shift out the value. The state 0 path is considered as the correct one, and the others are dropped.

Fig. 39 illustrates the survivor memory architecture with variable truncation length. The registers of each stage are connected to the path merging detection unit that decides the merge point and generates clock gating signals of each stage. Based on the scheme, we can adjust truncation length dynamically, depending on the channel. In high SNR environments, a shorter truncation length is required and the clock gating can be applied to more registers, resulting in a power efficient survivor memory.

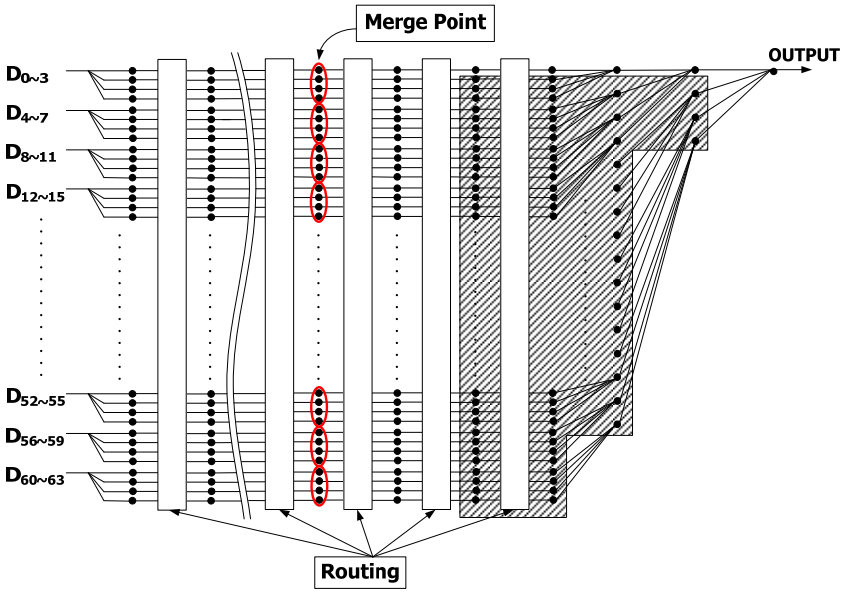


Fig. 38. Path merging detection scheme

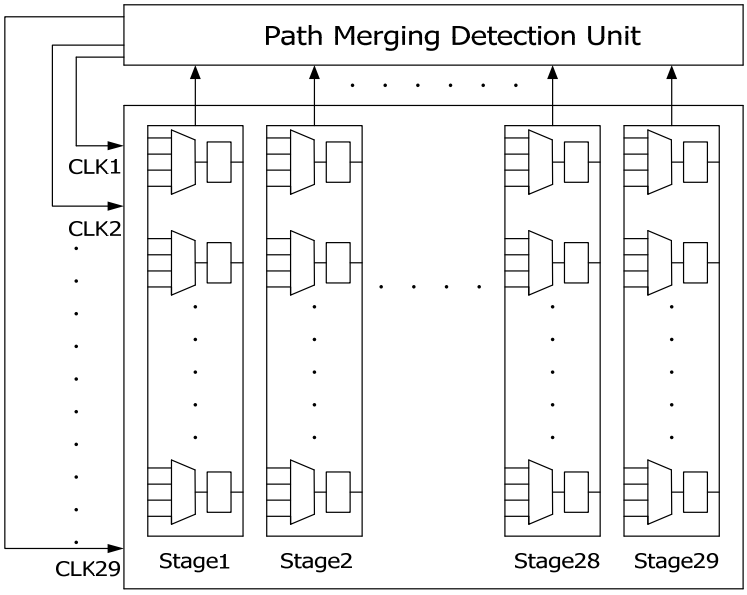


Fig. 39. Survivor memory architecture with variable truncation length

(2) DESIGN PARAMETERS AND PERFORMANCE SIMULATION

The Viterbi decoder, based on the register-exchange approach, combines the SST and the path merging detection schemes to reduce power dissipation. The design parameters of the proposed Viterbi decoder are listed in Table 11.

Fig. 40 shows the performance simulation result in the BPSK modulation. Notice that the performance of the conventional scheme, the SST scheme, and the proposed scheme are approximately the same. Compared with the floating point case, the performance degradation

of 8-level soft-decision is less than 0.5dB. The proposed variable truncation length scheme still preserves the error performance.

Table 11
Design parameters of proposed Viterbi decoder

Technology	1.2V 0.13-μm 1P8M CMOS
State Number	64
Code Rate	1/3
Soft-Decision	8-levels
BM Width	6 bits
PM Width	9 bits
Max. Truncation length	64
ACS structure	radix-4

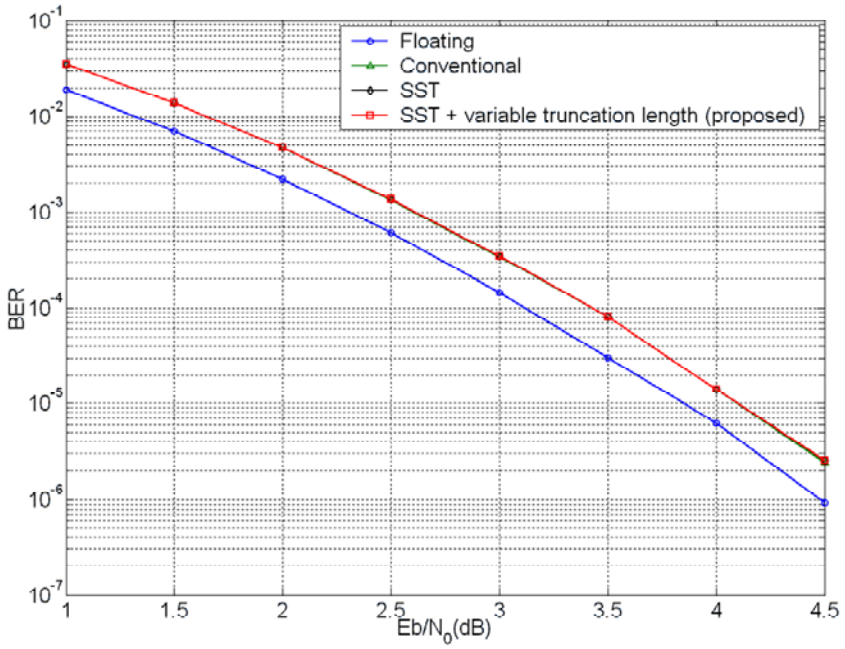


Fig. 40. Simulation results in AWGN channel, BPSK, 8-level soft decision and code rate=1/3

b) Power Simulation

We analyze the power dissipation of three implementations: the conventional register-exchange approach, the SST scheme without and with the variable truncation length scheme. Table 12 lists the gate counts of these implementations. In different channel

environments, we compare the power consumption of the three structures. Fig. 41(a) and Fig. 41(b) respectively reveal the post-layout power estimation of the whole Viterbi decoder and the survivor memory. The operating frequency is 250MHz and the corresponding data rate is 500Mbps due to the radix-4 ACS structure.

For the conventional design, the channel conditions are ineffective in the power dissipation. In the SST only implementation, the decoder power dissipation is reduced in high SNR environments; however, the power reduction is not obvious due to the complex signal wire routing. In the proposed design combining the SST and the variable truncation length, the decoder power has a significantly reduction as shown in the figures. Fig. 41(b) shows survivor memory power only to highlight the effect of the dynamic truncation length. As the channel condition is good enough, the variable truncation length scheme lowers more than 60% survivor memory power.

Fig. 42 shows the power profiling of the conventional register-exchange structure and the proposed decoder as E_b/N_0 is 5.0 dB. The corresponding bit error rate in this channel condition is $2.56e^{-6}$. From Fig. 42(a), the survivor memory is a power intensive block in conventional decoder designs. With SST and variable truncation length schemes, the ratio of survivor memory power is reduced significantly (see Fig. 42(b)). Furthermore, the SST unit and the path merging detection unit consume less than 2% of the decoder power.

Table 12
The gate counts of different implementations

Implementation	Gate counts
Conventional RE approach	108.5k
SST scheme	109.0k
Proposed	116.9k

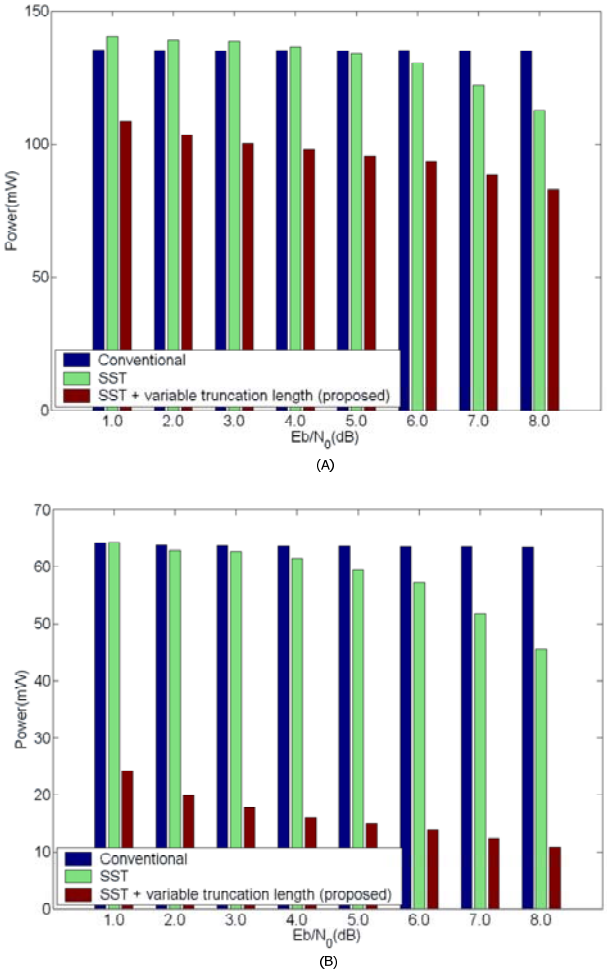


Fig. 41. Comparison of (A) Decoder power (B) Survivor memory power at 500Mbps

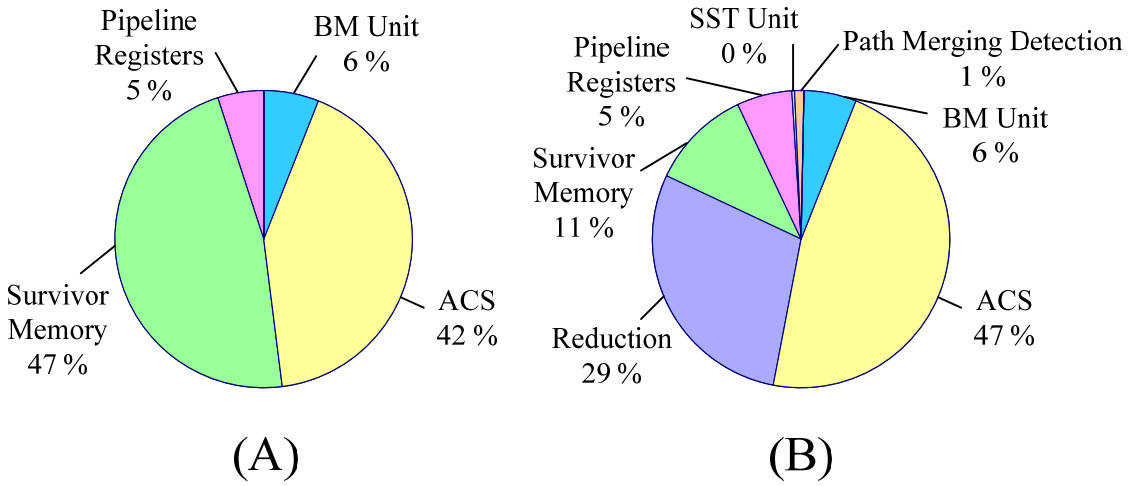


Fig. 42. The power profiling of (A) Conventional structure and (B) Proposed structure as E_b/N_0 is 5.0dB

2. A Low Power Differential Cascode Voltage Switch with Pass Gate Pulsed Latch for Viterbi Decoder

a) DCVSPG Pulsed Latch

In nano-scale technologies, pulsed latches address some of the challenges associated with realizing energy-efficiency flip-flops. First, only cascading a pulse generator and a conventional latch cannot considerably reduce total power consumption. Although the clock loading of pulsed latches is reduced, the internal node capacitance of the pulse generator is not. Second, the depth of the inverter chains of a pulse generator is increased to hold the pulse width since the propagation delay of inverters decreases in advanced technologies. Nevertheless, increasing the number of inverters would increase the power overhead. Third, the leakage current of more advanced technologies is higher. In view of the three challenges, a DCVSPG pulsed latch with a low swing pulse generator is proposed.

A DCVSPG pulsed latch is designed to realize an energy-efficient flip-flop. Fig. 43(a) presents the proposed DCVSPG pulsed latch, which is constructed using a low-swing pulse generator and a DCVSPG latch. This low-swing pulse generator generates an inverted clock using a three-stage inverter chain, a gated NMOS and a gated PMOS. The DCVSPG latch captures the input datum when both opposite clock and clock signals are high. The details of the low swing pulse generator and DCVSPG latch are as follows.

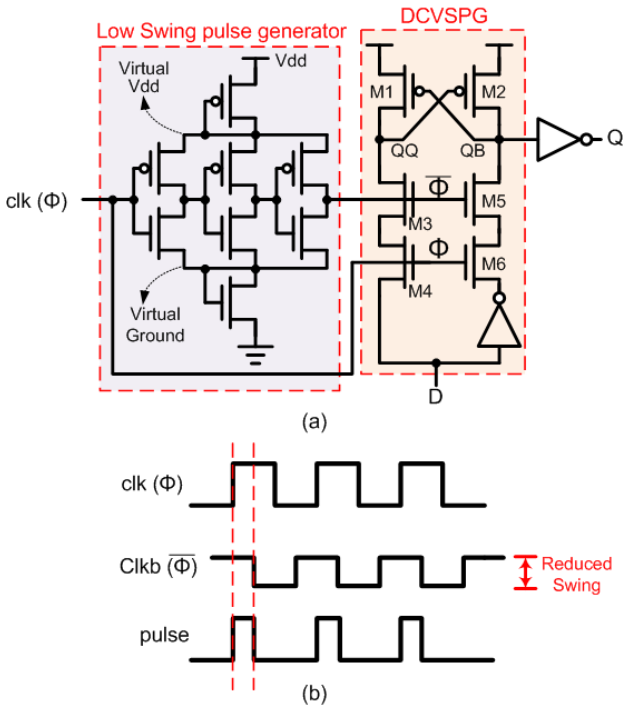


Fig. 43. (a) DCVSPG pulsed latch (b) Waveform of implicit pulse generator

(1) DCVSPG Latch

The DCVSPG latch is implemented using a differential cascode voltage switch with pass gate logic. Two cross-coupled PMOS transistors, M1 and M2, form the circuit load. Below the PMOS load, NMOS transistors form the n-channel logic evaluation. The DCVSPG latch captures input data in a transparent window generated by an implicit-pulse generator. Fig. 43(b) displays the corresponding waveform that is used to generate a transparent window. An implicit-pulse generator uses an odd-stage inverter chain to create a delayed signal of the opposite clock (clkb). According to the signals clk and clkb, an implicit pulse is generated as a transparent window by turning on NMOS pass transistors (M3 and M4, M5 and M6). The DCVSPG latch samples input data only in this transparent window. In a transparent window, the DCVSPG latch captures input data via four pass transistors. When the input datum is logic 1, the node QB is discharged to ground along the pull down path (M5 and M6). Accordingly, the output Q is changed to logic 1 following one propagation delay associated with an inverter. After the node QB is discharged, the node QQ is charged by the PMOS, M1. Additionally, the operation of sampling logic 0 is similar to that of sampling logic 1. When the input datum is logic 0, the node QQ is discharged to ground along the other pull down path (M3 and M4). M2

is turned on after the node QQ is discharged. Therefore, node QB is charged to logic 1 by this PMOS. Then, logic 0 is propagated to the output Q. Hence, capturing of logic 0 dominates the clock-to-Q delay. The DCVSPG pulsed latch provides high-speed data capturing during a transparent window using a differential cascode voltage switch.

(2) Low-Swing Pulse Generator

A major advantage of pulsed latches is the low clock load, which is associated with low power consumption in a clock tree. However, the pulsed latch has a penalty of the pulse generator. The pulse generator typically utilizes a delay inverter chain to produce the transparent window. The inverter chain is always switched as the clock switches. Accordingly, the pulse generator dissipates considerable power, even if the data undergo no transition. Moreover, in nano-scale technologies, leakage power dominates the overall power consumption. The inverter chain in a pulsed latch increases the number of leakage paths from the supply voltage to the ground. This increase causes the leakage of much power, which dominates the power consumption in a pulse generator.

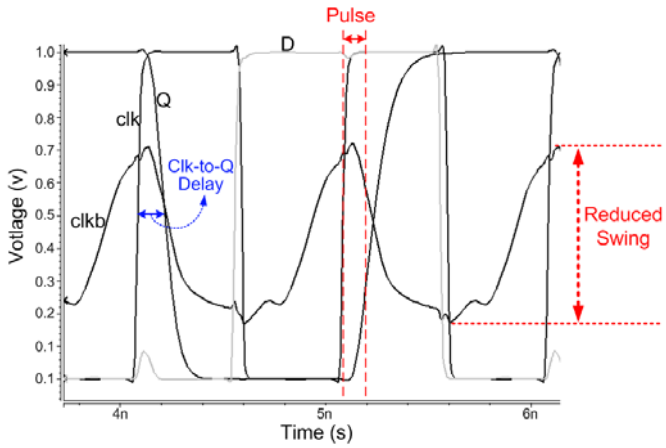


Fig. 44. Waveform of DCVSPG pulsed latch

A low-swing pulse generator is proposed for a DCVSPG pulsed latch to generate a low-swing inverted clock. The proposed low-swing pulse generator reduces the voltage swing in internal nodes to reduce switching power, and further reduces leakage power by gated diodes. Fig. 43(a) presents a DCVSPG pulsed latch with a low-swing pulse generator. To reduce the leakage power and switching power of the pulse generator, a transistor stacking

scheme is employed in the pulse generator. In the proposed low-swing pulse generator, a gated PMOS is connected between the supply voltage (Vdd) and virtual Vdd. Furthermore, a gated NMOS is inserted between the ground and the virtual ground. The purpose of these two gated transistors is to form a low-swing clock between virtual Vdd and virtual ground. Therefore, the voltage swing is reduced from Vdd to Vdd $-|V_{tp}|-V_{tn}$. Fig. 44 displays the corresponding waveform of the pulse generator and DCVSPG latch. The pulse width determines the hold time of the proposed DCVSPG pulsed latch. The proposed low-swing pulse generator has two advantages. First, stack transistors reduce leakage current and increase the propagation delay time. The delay chain of the DCVSPG pulsed latch is implemented to generate a transparent window. In advanced technologies, the number of inverters in a delay chain must be increased in a pulsed latch to guarantee that the width of the transparent window suffices for capturing the datum. Increasing the number of inverters causes substantial power consumption. Therefore, stack transistors increase the propagation delay without the need to add inverters. Second, reducing the voltage swing of the inverter chain reduces the switching power. However, the low-swing inverter chain decreases the noise immunity. Table 13 lists the static noise margin (SNM) of 3-stage low-swing and full-swing inverter chains. The average SNM of the low-swing inverter chain is reduced by 39% compared to that of the full-swing inverter chain.

Table 13
Static Noise Margin (SNM) of Three-Stage Inverter Chains

	Low-Swing	Full-Swing
SNM_{high}	291 mV	548 mV
SNM_{low}	287 mV	400

b) A Low-Power Radix-4 Viterbi Decoder Based on DCVSPG Pulsed Latch with Sharing Technique

(1) DCVSPG Pulsed Latch with Sharing Technique

With state number increasing, the truncation length should be longer. In this condition, the register number will increase rapidly. Moreover, the punctured convolutional codes need

longer truncation length to maintain performance. Fig. 45 shows the performance of punctured convolutional code, the result shows that the truncation length should be lengthened with high code rate punctured code. Last but not least, the path merge phenomenon will appear too late in low SNR conditions. In noisy channel, the switching activities are not reduced. Based on these features, it is feasible to optimize the registers. More power saving of register, more power saving of SMU.

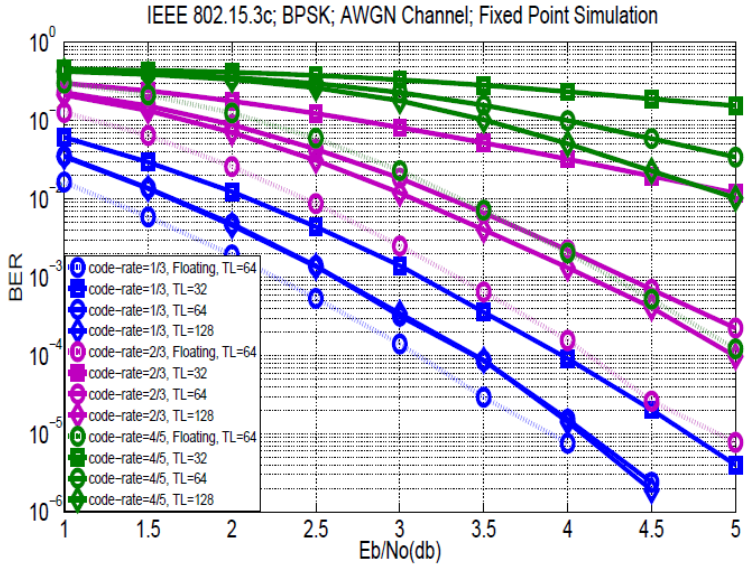
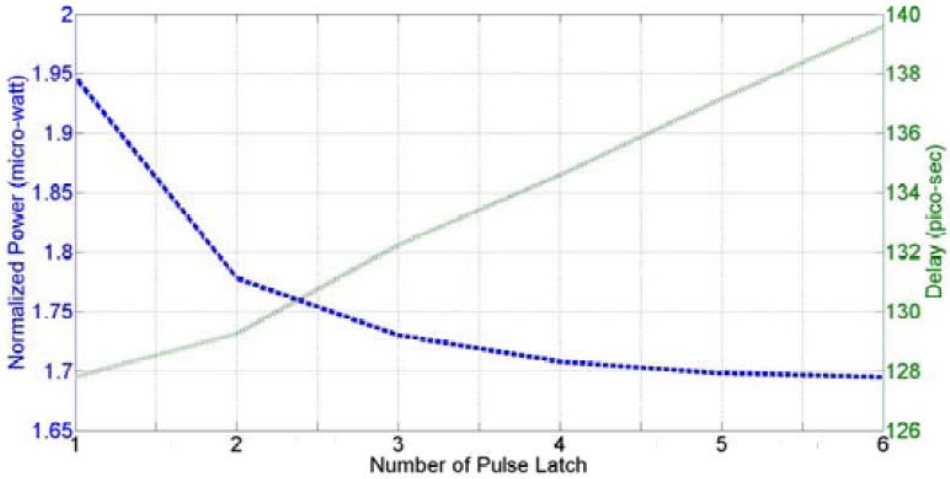


Fig. 45. Performance of punctured code with different truncation length

To address this issue, the DCVSPG pulsed latch with sharing technique has been developed. In order to reduce more power dissipation and area cost, the low-swing pulse generator can be shared by different number pulsed latches. In other words, one low-swing pulse generator could trigger more than one pulsed latches. However, the CLK-Q delay will become worse with increasing number of pulsed latches. To achieve a balance between delay and power consumption, the power-delay analysis plays an important role. With respect to the SMU of Viterbi decoder, the output loading of analysis and clock frequency are 8.36fF (loading of flip-flop in the SMU) and 250 MHz respectively. Moreover, we adopt high switching activity (α =1) input patterns to simulate low SNR conditions of communication



systems.

Fig. 46 shows the plot of power versus delay based on different number pulsed latches. According to the simulation result, the power consumption reduces obviously in small number pulse latches. But with increasing latches, this reduction turns into saturated and delay rises rapidly.

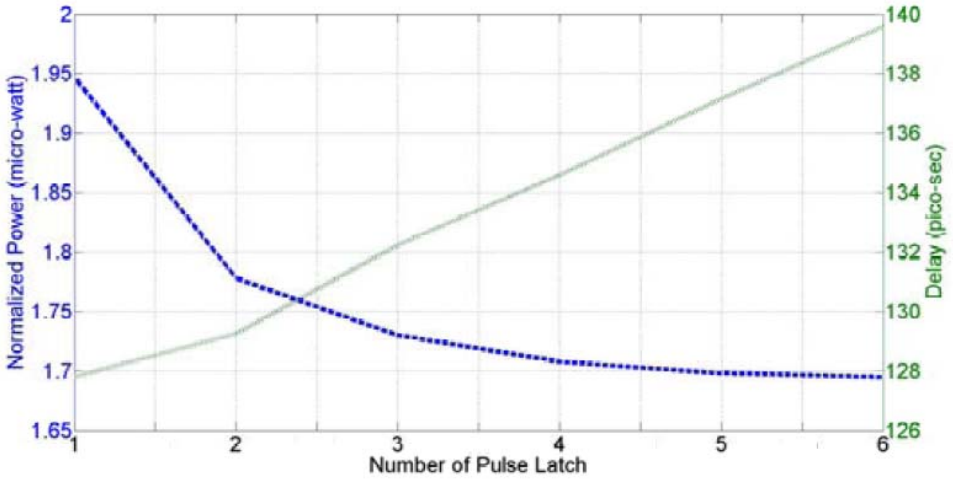


Fig. 46. Sharing analysis of different number of pulsed latch

(2) Proposed Radix-4 RE-based Viterbi Decoder

In the Viterbi decoder design, two important issues have to be considered carefully: speed and power. For high-speed applications, the high-radix architecture [C-11] becomes popular, to perform multi-step of trellis in one cycle. The data rate becomes n times based on radix-2n architecture, in other words, processing n bits in one cycle. In addition, without loss generality, the area and power overheads are also n times. Moreover, to achieve high data rate, the designer usually adopt RE-based SMU. The RE approach is the simplest technique which

assigns a set of registers to each state. These registers store the decision bits produced by ACSU. At each time step, these registers change the contents to update new decision bits. Based on RE approach, the way to trace back the survivor path is neglected and the latency could be reduced. However, the power consumption of RE method is larger due to exchanging registers. Based on the previous discussions and UWB system requirement, a radix-4 architecture that two flip-flops required in one basic unit in SMU is suitable for the Viterbi decoder. A radix-4, RE-based Viterbi decoder process 2 bits at a time, that is, two bit flip-flops are needed. Hereby, the proposed low-power RE-based Viterbi decoder by the DCVSPG pulsed latch with sharing technique is implemented for UWB system. The architecture of proposed RE-based SMU is shown in Fig. 47. To reduce the power dissipation of SMU, the DCVSPG pulsed latch with sharing technique is adopt to replace the flip-flops which are the power-hungry components in the SMU.

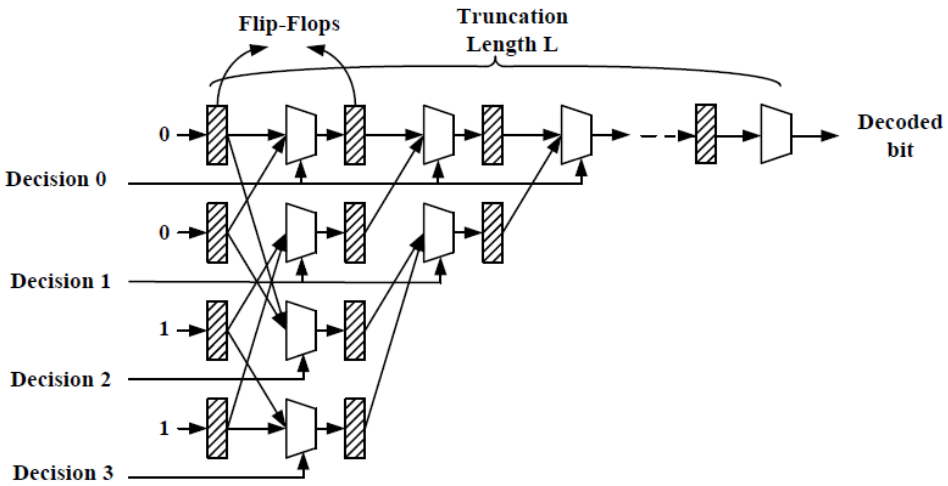


Fig. 47. SMU architecture design

c) *SIMULATION AND IMPLEMENTATION RESULTS*

Simulation and implementation results for our proposed Viterbi decoder are presented in this section. Table 14 presents the implementation results of two Viterbi decoders with C^2MOS flip-flops and DCVSPG pulsed latches, respectively. In the Viterbi decoder with DCVSPG pulsed latches, only the SMU block is implemented via DCVSPG pulsed latches, and other blocks are still synthesized using C^2MOS flip-flops, obtained from the UMC 90nm low power

cell library. In the Viterbi decoder using C^{2}MOS flip-flops, all the blocks are implemented by C^{2}MOS flip-flops, including the SMU block. To valid the design of the SMU using DCVSPG pulsed latches, the output pin of the decoded bit and three by-pass input pins (1 bit for input data and 2 bits for controlling the exchange of registers) are connected to the SMU to trace all the paths in SMU. The operating frequency and the throughput are 250 MHz and 500 Mb/s, respectively, to meet the requirement of UWB systems. The total number of gates and core size of the Viterbi decoder are 119K and 0.372mm^2, respectively. Moreover, the power consumption is 56.86mW at 0.9V, estimated from the post-layout simulation. By comparison with C^{2}MOS flip-flops, the DCVSPG pulsed latch not only reduces the power consumption of the Viterbi decoder by 21% but also reduces the core area by 12%. Fig. 48 presents the power distributions of the core power with C^{2}MOS flip-flops and DCVSPG pulsed latches. The power consumption of SMU is 70% of the total power consumption in a C^{2}MOS-based Viterbi decoder. The DCVSPG pulsed latch can reduce power consumption by 22% by reducing the power consumed by flip-flops and the clock tree of the SMU. Hence, the proposed DCVSPG pulsed latch is a power-efficient approach for implementation the SMU in a Viterbi Decoder.

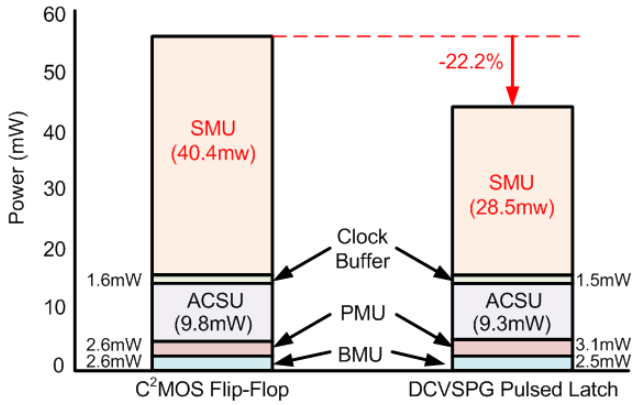


Fig. 48. Power Distributions of the Core Power with C^{2}MOS flip-flop and DCVSPG Pulsed Latch

Table 14
Implementation Results of Viterbi Decoder

	C^{2}MOS Flip-Flop	DCVSPG Pulsed Latch
Technology	UMC 90nm CMOS Process	
Supply Voltage	0.9 v	

State Number	64 States	
Truncation length	128-bit	
Gate count	126.8k	119.21k
Code rate / ACS architecture	(1/3) / Radix-4	
Throughput	500Mbps @ 250 MHz	
Core area (mm^2)	0.4225 (1)	**0.372 (0.88)**
Core Power (mW)	57.0 (1)	**44.9 (0.788)**

The pulse sharing technique is also implemented for a Viterbi decoder, with a radix-4 architecture. In this architecture, two bits must be stored in each time step. The performance of Viterbi decoder is shown in Fig. 49. According to the BER curves, the performance improvement will reach a limit when truncation length equal to 64 with 8-level soft-decision, 9-bit PM width and 6-bit BM width. Therefore, we select 64 as maximum truncation length. Table III lists the implementation results of the proposed Radix-4 Viterbi decoder based on UMC 90-nm 1P9M CMOS process. The operating frequency and data rate are 250 MHz and 500 Mb/s to meet the requirement of UWB system. Based on the DCVSPG pulsed latch with sharing technique, the total gate count is 84.35K, and the power consumption is 63.97 mW at 0.9V, estimated by post-layout simulation from the Cadence Ultrasim software. With the sharing technique, smaller clock tree loading could be realized. Fig. 50 shows the comparison of proposed decoder with UMC cell-based Viterbi decoders. Compared to cell-based decoder, the proposed DCVSPG with sharing technique Viterbi decoder can achieve 22.3% power saving.

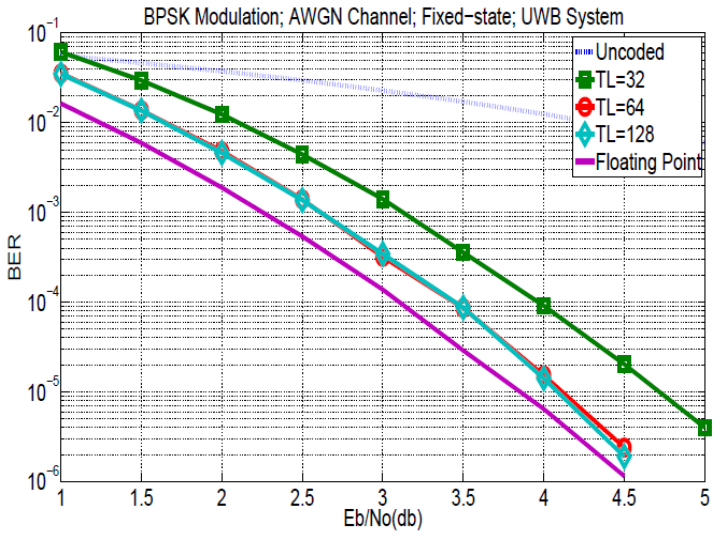


Fig. 49. Performance of Viterbi decoder with sharing technique

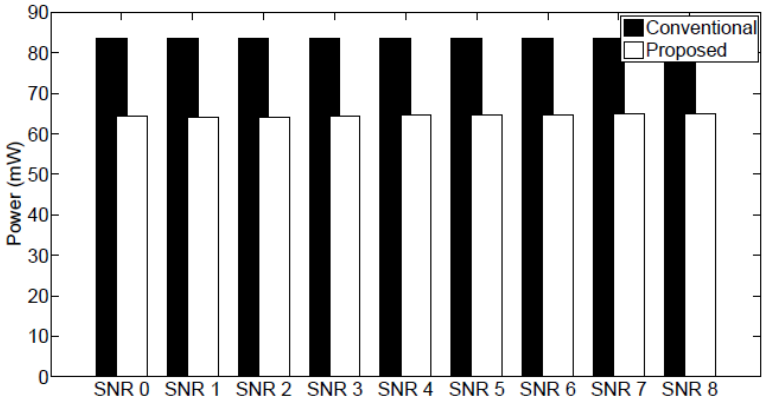


Fig. 50. Power comparison of Viterbi decoders

IV. 結論與討論

Among three years, we propose three kinds of modified FEC decoders for this low-power base-band processor and describe those contributions as follows:

A high-throughput and power-efficient LDPC decoder is presented. Utilizing the characteristic of variable-node-centric sequential scheduling, the proposed decoding algorithm could reduce the maximum iteration number without performance loss. In addition, the single pipelined architecture and modified CNU can save 73% message storage memory and decrease the sorter size, resulting in a low-complexity design. After implementation in 90nm technology, the test chip occupies 3.84 mm^2 and supports maximum 11.5 Gbps data rate under 1.4V supply voltage.

We have presented a LDPC-CC decoder design that targets high-throughput, low-cost and low-power. The test chip of (491, 3, 6) time-varying LDPC-CC supporting five code-rates is implemented in 90 nm CMOS technology. The decoder containing 5 processors occupies 2.24 mm^2 and provides twice faster decoding convergence speed. Maximum throughput 2.37Gbps is measured under 1.2 V supply with 0.024 nJ/bit/proc energy efficiency. The power can be scaled down to 90.2 mW with lowered throughput 1.58 Gbps at 0.8 V supply. The pro-posed design methodologies would make LDPC convolutional codes more competitive to the other error-control codes.

In BCH code, we provide soft BCH decoders suitable for digital video broadcasting. From the simulation, the proposed soft decoders perform much lower complexity with similar system performance as compared with conventional hard decoders. The proposed soft BCH decoders can save 61.2% gate-count for BCH (762,752) with H-EMS to correct 1 error. For BCH (32400, 32208) with BPEMS, which can correct 12 errors, 50.0% gate-count and 47.4% clock cycle latency are saved compared with traditional hard BCH decoders in CMOS 90nm technology.

We also provide an improved soft BCH decoder which performs better performance and comparable hardware complexity as compared to the conventional hard BCH decoder. The

complexity is reduced by dealing with the least reliable bits, and the error correcting ability is enhanced by compensating an extra error outside the least reliable set. In addition, Chien search can be eliminated with a counter that evaluates error locations in the proposed error locator evaluator procedure. Thus, a lot of redundant decoding latencies can be eliminated and higher throughputs can be achieved without parallelism. From the experimental results of BCH (255,239) code, the proposed soft decoder can give 0.75dB coding gain over the hard BCH decoder at BER = 10^{-5}. Also, it can achieve 316.3 Mb/s throughputs while reducing 7% gate-count as be compared with the 167.4Mb/s traditional hard BCH decoder in CMOS 90nm technology.

A novel decoding algorithm and its area-efficient architecture for soft RS codes is provided in this project. By confining the degree of error-locator polynomial, our approach has only one candidate sequence to be decoded. For RS (255,239) codes, our method can achieve 0.4 dB coding gain at 10^{-4} CER over hard decoders. Unlike Chase-type methods using several hard RS decoders and determining the most probable candidate, our proposal only demands one, leading to significant hardware complexity reduction. According to the measurement results, the proposed soft RS decoder can achieve 2.56 Gb/s throughput with 45.3 K gate. As a result, our proposal can provide more powerful correcting ability with a high-speed and area efficient solution for optical communications applications.

A low-power Viterbi decoder combining the SST and the variable truncation length scheme is presented. Reducing the state transition activities, the SST approach reduces the dynamic power in the decoder. The variable truncation length scheme provides a more efficient survivor memory management that pursues the sufficient truncation lengths for real channel conditions. Consequently, the redundant data movement can be abandoned for low power. Furthermore, high SNR environment causes the survivors merge rapidly, leading to more gated register elements. After implemented with the 0.13-μm cell based design flow, the fix-point error performance, the gate count, and the power dissipation has been examined by applying the presented low power techniques. Experimental results indicate the power reduction of the

whole decoder and the survivor memory unit can achieve more than 30% and 60% respectively, while the overhead of 8% gate count due to additional control logics is required.

We also provide another solution for a low-power Viterbi decoder by differential cascade voltage switch with pass gate pulsed latch. The DCVSPG pulsed latch is composed of a low swing pulse generator and a DCVSPG latch. The low-swing pulse generator generates an implicit pulse to capture the input datum. Additionally, the low-swing pulse generator not only reduces both the switching power and the leakage power by stacking gated transistors but also decreases the loading of the clock tree. During a transparent window that is produced by the low-swing pulse generator, the DCVSPG latch in the proposed pulsed latch performs energy-efficient data capture. The simulation results reveal that the proposed DCVSPG pulsed latch, based on UMC 90 nm CMOS technology, has lower energy consumption than other flip-flops. The proposed DCVSPG pulsed latch for the Viterbi decoder can reduce the power consumption by 22.2% below that achieved using a C^2MOS flip-flop, obtained from the UMC 90 nm low-power cell library. To reduce more power consumption, the low-swing DCVSPG pulsed latch with sharing technique is introduced to reduce the switching power and leakage power; in addition, less clock loading could be achieved. Experimental results indicate the proposed radix-4 Viterbi decoder with 84.35K gate counts can achieve 500 Mb/s and consume 63.97 mW at 0.9V with the energy efficiency 0.128 nJ/bit.

V. 參考文獻

[A-1] R. G. Gallager, Low-Density Parity-Check Codes. Cambridge, MA: MIT Press, 1963.

[A-2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," Electron. Lett., vol. 33, no. 6, pp.457–458, Mar. 1997.

[A-3] M. Mansour and N. Shanbhag, "High-throughput LDPC decoders," IEEE Trans. on VLSI Systems, vol. 11, no. 6, pp. 976–996, Dec. 2003.

[A-4] Part 3: carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specificaions, IEEE Std. P802.3an-2006, Sept. 2006.

[A-5] Part 15.3: wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPANs), IEEE Std. P802.15.3c-DF8, 2009.

[A-6] J. Zhang and M. Fossorier, "Shuffled iterative decoding," IEEE Transactions on Communications, vol. 53, no. 2, pp. 209–213, Feb. 2005.

[A-7] Y. K. Lin, C. L. Chen, Y. C. Liao, and H. C. Chang, "Structured LDPC codes with low error floor based on peg tanner graphs," in IEEE Int. Sympo. Circuits and Systems (ISCAS'08), May 2008, pp. 1846–1849.

[A-8] A. Darabiha, A. C. Carusone, and F. R. Kschischang, "A 3.3-Gbps bitserial block-interlaced min-sum LDPC decoder in 0.13-μm CMOS," in Proc. IEEE CICC'07, Sept. 2007, pp. 459–462.

[A-9] X. Y. Shih, C. Z. Zhan, C. H. Lin, and A. Y. Wu, "A 19-mode 8.29mm2 52-mW LDPC decoder chip for IEEE 802.16e system," in Proc. Int. Sympo. VLSI Circuits (SOVC'07), June 2007, pp. 16–17.

[A-10] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," IEEE Transactions on Information Theory, vol. 45, no. 6, pp. 2181 – 2191, Sep. 1999.

[A-11] E. Matus, M. Tavares, M. Bimberg, and G. Fettweis, "Towards a GBit/s programmable decoder for LDPC convolutional codes," in IEEE International Symposium on Circuits and Systems (ISCAS), May 2007, pp. 1657 – 1660.

[A-12] M. Bimberg, M. Tavares, E. Matus, and G. Fettweis, "A high-throughput programmable decoder for LDPC convolutional codes," in IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP), July 2007, pp. 239 – 246.

[A-13] Z. Chen, S. Bates, and W. Krzymien, "High throughput parallel decoder design for

LDPC convolutional codes," in IEEE International Conference on Circuits and Systems for Communications (ICCSC), May 2008, pp. 35 – 39.

[A-14] Z. Chen, T. Brandon, D. Elliott, S. Bates, W. Krzymien, and B. Cockburn, "Jointly designed architecture-aware LDPC convolutional codes and high-throughput parallel encoders/decoders," IEEE Transactions on Circuits and Systems I, vol. 57, no. 4, pp. 836 – 849, Apr. 2010.

[A-15] A. Pusane and et. Al, "Reduced complexity decoding strategies for LDPC convolutional codes," in Proc. IEEE ISIT 2004, 2004, p. 490.

[A-16] Amendment Text Proposal on Rate Compatible LDPC-Convolutional Codes, IEEE Std. C802.16m-09/0339.

[A-17] T. L. Brandon and et. Al, "A compact 1.1-gb/s encoder and a memory-based 600-mb/s decoder for LDPC convolutional codes," IEEE Trans. Circuits and Syst. I, vol. 56, no. 5, pp. 1017–1029, May 2009.

[A-18] C. Studer and et. Al, "A 390mb/s 3.57mm2 3GPP-LTE turbo decoder ASIC in 0.13μm CMOS," in IEEE ISSCC 2010, 2010.

[B-1] C. R. Baugh and B. A. Wooley, Theory and Practice of Error Control Codes. Addison-Wesley, 1983

[B-2] Framing Structure, Channel Coding and Modulation for Digital Television Terrestrial Broadcasting System, NSPRC Std. GB 20 600-2006, 2007

[B-3] Digital Video Bracasting (DVB) Second Generation System for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications, ETSI Std. En 302 307, 2005

[B-4] G. D. Forney, "Generalized Minimum Distance Decoding," IEEE Trans. Inform. Theory, vol. 12, p. 125V131, Apr. 1966

[B-5] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," IEEE Trans. Inform. Theory, vol. IT-18, p. 170V182, Jan. 1972

[B-6] M. Lalam, K. .Amis, D. Lerous, D. Feng, and J. Yuan, "An Improved Iterative Decoding Algorithm for Block Turbo Codes," IEEE Int. Symp. on Info. Theory, pp. 2403–2407, July 2006

[B-7] W. J. ReidIII, L. L. Joiner, and J. J. Komo, "Soft Decision Decoding of BCH Codes Using Error Magnitudes," IEEE Int. Symp. on Info. Theory, p. 303, June 1997

[B-8] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," IEEE Trans. Inform. Theory,, vol. 20, pp. 284–287, Mar.

1974

[B-9] A. Bj¨orck and V. Pereyra, “Solution of Vandermonde Systems of Equations,” Math. Computation, vol. 24, pp. 893–903, Oct. 1970

[B-10] J. Hong and M. Vetterli, “Simple Algorithms for BCH Decoding,” IEEE Trans. on Communication, vol. 43, pp. 2324–2333, Aug. 1995

[B-11] C. Parr, “Efficient VLSI Architectures for Bit-Parallel Computation in Galois Fields,” Ph.D. dissertation, Inst. for Experimental Mathematics of Univ. of Essen Germany, 1994

[B-12] F. Therattil and A. Thangaraj, “A Low-complexity Soft-decision Decoder for Extended BCH and RS-Like Codes,” IEEE Trans. Inform. Theory, p. 1320V132, Sept. 2005.

[B-13] Y. Chen and K. Parhi, “Small Area Parallel Chien Search Architectures for Long BCH Codes,” IEEE Trans. on VLSI, vol. 12, no. 5, pp. 545–549, May 2004.

[B-14] J. Cho and W. Sung, “Strength-Reduced Parallel Chien Search Architecture for Strong BCH Codes,” IEEE Trans. on Circuits and Systems II, vol. 55, no. 5, pp. 427–431, May 2008.

[B-15] I. S. Reed, M. T. Shih, and T. K. Truong, “VLSI Design of Inverse-Free Berlekamp-Massey Algorithm,” Proc. Inst. Elect. Eng, vol. 138, pp. 295–298, Sept. 1991

[B-16] D. Sarwate and N. Shanbhag, “High-speed architectures for Reed-Solomon decoders,” IEEE Trans. VLSI Syst., vol. 9, no. 5, pp. 641–655,Oct. 2001.

[B-17] A. Raghupathy and K. J. R. Liu, “Algorithm-Based Low-Power/High-Speed Reed-Solomon Decoder Design,” *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 47, no. 11, pp.1254–1270, Nov. 2000.

[B-18] D. Sarwate and N. Shanbhag, “High-speed architectures for Reed-Solomon decoders,” *IEEE Trans. VLSI Syst.*, vol. 9, no. 5, pp. 641–655, Oct. 2001.

[B-19] X. Zhang, “High-speed VLSI architecture for low-complexity Chase soft-decision Reed-Solomon decoding,” *IEEE Inform. Theory and Application Workshop*, pp. 422–430, Feb 2009.

[B-20] J.-I. Park, K. Lee, C.-S. Choi, and H. Lee, “High-speed low-complexity Reed-solomon decoder using pipelined berlekamp-massey algorithm,” *IEEE Int. SoC Design Conference (ISOCC)*, pp. 452–455, 2009.

[B-21] S. Lee, H. Lee, J. Shin, and J.-S. Ko, “A High-Speed Pipelined Degree-Computationless Modified Euclidean Algorithm Architecture for Reed-Solomon Decoders,” *IEEE Int. Symp. on Circuits and Systems (ISCAS).*, pp. 901-904, May 2007.

[B-22] J. Baek and M. Sunwoo, “New Degree Computationless Modified Euclid Algorithm and Architecture for Reed-Solomon Decoder,” *IEEE Trans. VLSI Syst.*, vol. 14, no. 8, pp.

915–920, 2006.

[C-1] A. J. Viterbi, “Error bounds for convolutional codes and asymptotically optimum decoding algorithm,” IEEE Trans. Inform. Theory, vol. IT-13, no. 2, pp. 260-269, Apr. 1967.

[C-2] G. Feygin and P. G. Gulak, “Architectural tradeoffs for survivor sequence memory management in Viterbi decoders,” IEEE Transactions on Communications, vol. 41, no. 3, pp. 425-429, March, 1993.

[C-3] C. C. Wu, “A High-Performance and Low-Power Viterbi Decoder,” MS. thesis, National Chiao Tung Univ., 2003.

[C-4] A. Batra et al., “Multi-band OFDM physical layer proposal for IEEE 802.15 task group 3a,” submitted to IEEE P802.15 working group for WPANs, Sept. 2004.

[C-5] P. J. Black and T. H. Meng, “A 140-Nb/s, 32-state, radix-4, Viterbi Decoder,” IEEE J. Solid-State Circuits, vol. 27, no. 12, pp. 1877-1885, Dec. 1992.

[C-6] S. Kubota and S. Kato, “Novel Viterbi Decoder VLSI Implementation and its Performance,” IEEE Trans. Commun., vol. 41, no. 8, pp. 1170-1178, Aug. 1993.

[C-7] L. H. C. Lee, D. J. Tait, and P. G. Farrell, “Scarce-State-Transition Syndrome-Former Error-Trellis Decoding of (n,n-1) Convolutional Codes,” IEEE Trans. Commun., vol. 44, no.1, pp. 7-9, Jan. 1996.

[C-8] S. Kubota, K. Ohtani, and S. Kato, “High-Speed and High-Coding-Gain Viterbi Decoder with Low Power Consumption Employing SST (Scarce State Transition) Scheme,” Electron. Lett., vol. 22, no.9, pp. 491-493, Apr. 1986.

[C-9] C. C. Lin, Y. H. Shih, H. C. Chang, and C. Y. Lee, “Design of a Power-Reduction Viterbi Decoder for WLAN Applications,” IEEE Trans. Circuit and Syst. I, vol. 52, no. 6, pp. 1148-1156, June 2005.

[C-10] D.-J. Lin, C.-C. Lin, C.-L. Chen, H.-C. Chang, and C.-Y. Lee, “A Low-Power Viterbi Decoder Based on Scarce State Transition and Variable Truncation Length,” *Proceedings of the IEEE International Symposium on VLSI Design, Automation and Test* (**2007**), pp. 99-102.

[C-11] A. K. Yeung and J. M. Rabaey, “A 210-Mb/s Radix-4 Bit-level Pipelined Viterbi Decoder,” *IEEE Int. Solid-State Circuits Conf. Dig. Tech*, pp. 88-89, Feb. 1995

VI. 計畫成果自評

在此計畫執行三年中，我們提供七個可應用於無線通訊之低功耗基頻處理器元件分別為：

I. A 11.5-Gbps LDPC Decoder Based on CP-PEG Code Construction
II. A 2.37Gb/s 284.8mW Rate-Compatible (491, 3, 6) LDPC-CC Decoder
III. A Soft BCH Decoder Chip for DVB-S2 System
IV. An Improved Soft BCH Decoder with One Extra Error Compensation
V. A Decision-Confined Soft RS Decoder
VI. A Low-Power Viterbi Decoder Based on Scarce State Transition and Variable Truncation Length
VII. A Low Power Differential Cascode Voltage Switch with Pass Gate Pulsed Latch for Viterbi Decoder

下表一至三是為本研究團隊對於今年執行的計畫進度與各別研究規劃，在表中，我們完成了六個研究的架構設計、下線與陸續完成量測；在此計畫三年的研究成果，希望能在未來為產業界、學術界盡一份心力。

表一　　子計畫一之研究規劃

月份 (2009)	01	02	03	04	05	06	07	08	09	10	11	12
A 11.5-Gbps LDPC Decoder Based on CP-PEG Code Construction												
LDPC Paper Survey												
CP-PEG Code Construction												
Sequential Scheduling Design												
Algorithm Simulation												
Architecture Design												
Cell-Based Design												
Chip Layout												

月份 (2010)	01	02	03	04	05	06	07	08	09	10	11	12
A 2.37Gb/s 284.8mW Rate-Compatible (491,3,6) LDPC-CC Decoder												
LDPC-CC Paper Survey												
Algorithm Simulation												
Scheduling												
CNU Architecture												
Decoder Architecture Design												
Cell-Based Design												
Chip Layout												

表二　　子計畫二之研究規劃

月份 (2008)	01	02	03	04	05	06	07	08	09	10	11	12
Soft BCH Decoders Using Error Magnitudes for Digital Video Broadcasting												
Soft BCH Paper Survey												
Soft BCH Algorithm Simulation for DMB-T												
Soft BCH Algorithm Simulation for DVB-S2												
Soft BCH Architecture Design												
Soft BCH for DMB-T APR												
Soft BCH for DVB-S2 Tape-out												

月份 (2009)	01	02	03	04	05	06	07	08	09	10	11	12
An Improved Soft BCH Decoder with One Extra Error Compensation												
Soft BCH Paper Survey												
Soft BCH Algorithm Development												
Soft BCH Algorithm Simulation												
Soft BCH Architecture Design												
Soft BCH Cell-Based Design												

月份 (2010)	01	02	03	04	05	06	07	08	09	10	11	12
A Decision-Confined Soft RS Decoder												
Soft RS Paper Survey												
Soft RS Algorithm Development												
Soft RS Algorithm Simulation												
Soft RS Architecture Design												
Soft RS Cell-Based Design												

表三　　子計畫三之研究規劃

月份 (2008)	01	02	03	04	05	06	07	08	09	10	11	12
A Low-Power Viterbi Decoder Based on Scarce State Transition and Variable Truncation Length												
Modified SST Algorithm												
Modified SST Architecture Design												
Modified SST VD FPGA Platform												
Modified SMU Prepare												
Modified SMU Design & Simulation												
Modified SST + SMU VD FPGA Platform												
Modified SST + SMU VD Chip Layout												
Modified SST + SMU VD IC Tape-out												

月份 (2010)	01	02	03	04	05	06	07	08	09	10	11	12
A Low Power Differential Cascode Voltage Switch with Pass Gate Pulsed Latch for Viterbi Decoder												
Low Power Latch Paper Survey												
DCVSPG Pulsed Latch Development												
DCVSPG Pulsed Latch Simulation												
Viterbi Decoder Architecture Design												
Viterbi Decoder Cell-Based Design												

感謝本計畫對我們的鼓勵，讓我們每年有穩定成長的貢獻，下者我們列出掛有本計畫的論文：

1. Yi-Min Lin, Chih-Lung Chen, Hsie-Chia Chang, and **Chen-Yi Lee**, "A 26.9K 314.5Mbps Soft (32400,32208) BCH Decoder Chip for DVB-S2 System," in *IEEE Journal of Solid-State Circuits,* vol.45, no.11, pp.2330-2340, Nov. 2010

2. Yi-Min Lin, Hsie-Chia Chang, and **Chen-Yi Lee**, "An Improved Soft BCH Decoder with One Extra Error Compensation," in *IEEE Int. Symposium on Circuits and Systems (ISCAS),* France Paris, May 2010

3. Yi-Min Lin, Chih-Lung Chen, Hsie-Chia Chang, and **Chen-Yi Lee**, "A 26.9K 314.5Mbps Soft (32400,32208) BCH Decoder Chip for DVB-S2 System," in *IEEE Asia Solid State Circuits Conf. (ASSCC), Taiwan Taipei*, Nov. 2009

4. Chic-Lung Chen, Kao-Shou Lin, Hsie-Chia Chang, Wai-Chi Fang, and **Chen-Yi Lee**, "A 11.5-Gbps LDPC Decoder Based on CP-PEG Code Construction," in *IEEE ESSCIRC Proceedings*, Greece Athens, Sep. 2009

5. Xin-Ru Lee, Hsie-Chia Chang, and **Chen-Yi Lee**, "A Low-Power Radix-4 Viterbi Decoder Based on DCVSPG Pulsed Latch with Sharing Technique," in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Kuala Lumpur Malaysia, Dec. 2010

6. Chih-Lung Chen, Yu-Hsiang Lin, Hsie-Chia Chang and **Chen-Yi Lee**, "A 2.37Gb/s 284.8mW Rate-Compatible (491,3,6) LDPC-CC Decoder," in *IEEE Symposium on VLSI Circuits*, pp. 134-135, June 2011

7. Chih-Hsiang Hsu, Yi-Min Lin, Hsie-Chia Chang, and **Chen-Yi Lee**, "A 2.56Gb/s Soft RS (255,239) Decoder Chip for Optical Communication Systems", *has been accepted by IEEE ESSCIRC Proceedings*

8. Yi-Min Lin, Hsie-Chia Chang, and **Chen-Yi Lee**," A MPCN-Based Parallel Architecture in BCH Decoders for NAND Flash Memories" , has been accepted by *IEEE Trans. Circuits Syst. II*

表四簡列出此計畫支持本研究室的相關研究成果，其中，有 6 件國內、外專利申請中、16 篇國際期刊與會議論文已發表於 IEEE（相關研究統計列出如附件所示），本研究室亦十分感謝國家科學委員會今年暨未來的持續支持與鼓勵。

表四 本計畫相關近年(2007-2011)研究貢獻

成果項目			96.01.01-100.07.31	成果項目			96.01.01-100.07.31
專 利	申請	國內（件數）	**3**	**論文**	期刊	國內（件數）	**0**
		國外（件數）	**3**			國外（件數）	**4+2***
		國內外合計件數	**6**			國內外合計件數	**4+2***
	獲得	國內（件數）	**3**		研討會	國內（件數）	**0**
		國外（件數）	**3**			國外（件數）	**7+3***
		國內外合計件數	**6**			國內外合計件數	**7+3***

*：今年度研究量產的論文數

VII. 附錄－ 2007-2011 本計畫相關之研究成果

Patents:

1. 林義閔、楊其衡、張錫嘉、李鎮宜，” 一種用以處理循環碼之方法及裝置,”中華民國專利申請第 099135609 號，99 年 10 月 19 日。

2. 翁政吉、唐正浩、張錫嘉、李鎮宜，”應用於迭代解碼之多層級網路架構及其傳輸方法,” 中華民國專利申請第 96149409 號，96 年 12 月 21 日。(公告號：200929892)(經濟部科專 95-EC-17-A-01-S1-048)

3. 陸志豪、廖彥欽、李鎮宜、許雅三、張錫嘉，”應用於低密度對稱檢查碼（LDPC）解碼器之運算方法及其電路,” 中華民國專利申請第 096128039 號，96 年 7 月 31 日。(公告號：200906073)(經濟部科專 93-EC-17-A-03-S1-0005)

4. 陸志豪、林建青、李鎮宜、許雅三、張錫嘉，”用於通訊系統的資料交換裝置及方法,” 中華民國專利發明 I339944 號，96 年 4 月 23 日。(國科會 NSC94-2220-E-009-027)

5. 陸志豪、林建青、李鎮宜、張錫嘉、許雅三，”多模多平行度資料交換方法及其裝置,” 中華民國專利發明 I339955 號，96 年 12 月 7 日。(經濟部科專 93-EC-17-A-03-S1-0005)。

6. 李鎮宜、林建青、林凱立、張錫嘉, “用於更新低密度配類核對（LDPC）碼解碼器之核對節點的方法及其裝置,” 中華民國專利發明 I291290 號，96 年 12 月 11 日。(國科會 NSC 93-2220-E-009-033)

7. 陸志豪、廖彥欽、李鎮宜、許雅三、張錫嘉，”應用於低密度對稱檢查碼（LDPC）解碼器之運算方法及其電路,” 日本專利申請特願 2008-082997 號，97 年 3 月 27 日。(經濟部科專 93-EC-17-A-03-S1-0005)

8. Yi-Min Lin, Chi-Heng Yang, Hsie-Chia Chang, and Chen-Yi Lee, “APPARATUS AND METHOD OF PROCESSING CYCLIC CODES” has been filed as U.S. Patent pending, 12/790,875, May 31, 2010.

9. Chih-Hao Liu, Yen-Chin Liao, Chen-Yi Lee, Hsie-Chia Chang, and Yar-Sun Hsu, “Operating Method Applied to Low Density Parity Check (LDPC) Decoder and Circuit Thereof”, has been filed as U.S. Patent pending, 11/939119, November 13, 2007. (pub. no. 20090037799) (經濟部科專 93-EC-17-A-03-S1-0005)

10. Chih-Hao Liu, Chien-Ching Lin, Chen-Yi Lee, Hsie-Chia Chang, and Yar-Sun Hsu, “Multi-mode multi-parallelism data exchange method and thereof,” U.S. Patent pending,

7719442 B2, March 13, 2008.（經濟部科專 93-EC-17-A-03-S1-0005）

11. Cheng-Chi Wong, Cheng-Hao Tang, Hsie-Chia Chang, and Chen-Yi Lee, ”Apparatus of multi-stage network for iterative decoding,” U.S. Patent pending, 7724163 B2, July 24, 2008.（經濟部科專 95-EC-17-A-01-S1-048）

12. Chih-Hao Liu, Chih-lung Chen, Chen-Yi Lee, Yar-Sun Hsu, and Hsie-Chia Chang, “Method and Apparatus for Switching Data in Communication System,” U.S. Patent pending, 7724772 B2, May 18, 2007.（國科會 NSC94-2220-E-009-027）

Journals:

1. Yi-Min Lin, Chi-Heng Yang, Chih-Hsiang Hsu, Hsie-Chia Chang, and Chen-Yi Lee,” A MPCN-Based Parallel Architecture in BCH Decoders for NAND Flash Memories” , has been accepted by *IEEE Trans. Circuits Syst. II*

2. Yi-Min Lin, Chih-Lung Chen, Hsie-Chia Chang, and Chen-Yi Lee, “A 26.9K 314.5Mbps Soft (32400,32208) BCH Decoder Chip for DVB-S2 System,” in *IEEE Journal of Solid-State Circuits*, vol.45, no.11, pp.2330-2340, Nov. 2010

3. Cheng-Chi Wong, Ming-Wei Lai, Chien-Ching Lin, Hsie-Chia Chang, and Chen-Yi Lee, “Turbo Decoder Using Contention-Free Interleaver and Parallel Architecture,” *IEEE Journal of Solid-State Circuits*, vol. 45, no. 2, pp.422-432, 2010.

4. Chih-Hao Liu, Chien-Ching Lin, Shau-Wei Yen, Chih-Lung Chen, Hsie-Chia Chang, Chen-Yi Lee, Yar-Sun Hsu and Shyh-Jye Jou, “Design of a Multimode QC-LDPC Decoder Based on Shift-Routing Network,” *IEEE Trans. Circuits Syst. I*, , vol.56, no.9, pp.734-738, September 2009.（SCI/EE）

5. Hsie-Chia Chang, Chien-Ching Lin, Fu-Ku Chang, and Chen-Yi Lee, “A Universal VLSI Architecture for Reed-Solomon Error-and-Erasure Decoders,” *IEEE Trans. Circuits Syst. I*, vol.56, no.9, pp.1960-1967, September 2009.

6. Chih-Hao Liu, Shau-Wei Yen, Chih-Lung Chen, Hsie-Chia Chang, Chen-Yi Lee, Yar-Sun Hsu, and Shyh-Jye Jou, “An LDPC Decoder Chip Based on Self-Routing Network for IEEE 802.16e Applications,” *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 684~694, March 2008.（SCI/EE）

Conference:

1. Chih-Hsiang Hsu, Yi-Min Lin, Hsie-Chia Chang, and Chen-Yi Lee, "A 2.56Gb/s Soft RS (255,239) Decoder Chip for Optical Communication Systems", *has been accepted by IEEE ESSCIRC Proceedings*

2. Chih-Lung Chen, Yu-Hsiang Lin, Hsie-Chia Chang and Chen-Yi Lee, "A 2.37Gb/s 284.8mW Rate-Compatible (491,3,6) LDPC-CC Decoder," in *IEEE Symposium on VLSI Circuits*, pp. 134-135, June 2011

3. Xin-Ru Lee, Hsie-Chia Chang, and Chen-Yi Lee, "A Low-Power Radix-4 Viterbi Decoder Based on DCVSPG Pulsed Latch with Sharing Technique," in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Kuala Lumpur Malaysia, Dec. 2010

4. Yi-Min Lin, Hsie-Chia Chang, and Chen-Yi Lee, "An Improved Soft BCH Decoder with One Extra Error Compensation," in *IEEE Int. Symposium on Circuits and Systems (ISCAS),* France Paris, May 2010.（EI）

5. Yi-Min Lin, Chih-Lung Chen, Hsie-Chia Chang, and Chen-Yi Lee, "A 26.9K 314.5Mbps Soft (32400, 32208) BCH Decoder Chip for DVB-S2 System," *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Hsinchu, Taiwan, November 2009, pp.373-376, (EI)

6. Shao-Wei Yen, Ming-Chih Hu, Chih-Lung Chen, Hsie-Chia Chang, Shyh-Jye Jou, and Chen-Yi Lee, "A 0.92mm2 23.4mW Fully-Compliant CTC Decoder for WiMAX 802.16e Application," *IEEE Custom Integrated Circuits Conference (CICC)*, , San Jose, California, October 2009, pp.191-194. (EI).

7. Chih-Lung Chen, Kao-Shou Lin, Hsie-Chia Chang, Wai-Chi Fang, and Chen-Yi Lee, "A 11.5-Gbps LDPC Decoder Based on CP-PEG Code Construction," *European Solid-State Circuits Conf. (ESSCIRC)*, Athens, Greece, September 2009.

8. Chih-Hao Liu, Chien-Ching Lin, Hsie-Chia Chang, Chen-Yi Lee, and Yar-Sun Hsu, "Multi-mode Message Passing Switch Networks Applied for QC-LDPC Decoder," in *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, Seattle, U.S., May 2008, pp.752-755. (EI)

9. Cheng-Chi Wong, Cheng-Hao Tang, Ming-Wei Lai, Yan-Xiu Zheng, Chien-Ching Lin, Hsie-Chia Chang, Chen-Yi Lee, and Yu-T. Su, "A 0.22nJ/b/iter 0.13μm Turbo Decoder Chip Using Inter-Block Permutation Interleaver," in *IEEE Custom Integrated Circuits Conference (CICC)*, San Jose, California, October 2007, pp.273-276.

10. Dah-Jia Lin, Chien-Ching Lin, Chih-Lung Chen, Hsie-Chia Chang, and Chen-Yi Lee, “A Low-Power Viterbi Decoder Based on Scarce State Transition and Variable Truncation Length,” in *IEEE Int. Symposium on VLSI Design, automation and Test (VLSI-DAT)*, Hsinchu, Taiwan, April 2007, pp.99~102. (EI)