

Low-Complexity Chase Decoding of Reed–Solomon Codes Using Channel Evaluation

Hao Wang ¹, Wei Zhang ¹, Yanyan Chang ¹, Jiajing Gao ¹ and Yanyan Liu ^{2,*}

¹ School of Microelectronics, Tianjin University, Tianjin 300072, China; ahao@tju.edu.cn (H.W.); tjuzhangwei@tju.edu.cn (W.Z.); yyxsdut@tju.edu.cn (Y.C.); echo_gao0214@tju.edu.cn (J.G.)

² College of Electronic Information and Optical Engineering, Nankai University, Tianjin 300071, China

* Correspondence: lyytianjin@nankai.edu.cn

Abstract: A novel time-varying channel adaptive low-complexity chase (LCC) algorithm with low redundancy is proposed, where only the necessary number of test vectors (TVs) are generated and key equations are calculated according to the channel evaluation to reduce the decoding complexity. The algorithm evaluates the error symbol numbers by counting the number of unreliable bits of the received code sequence and dynamically adjusts the decoding parameters, which can reduce a large number of redundant calculations in the decoding process. We provide a simplified multiplicity assignment (MA) scheme and its architecture. Moreover, a multi-functional block that can implement polynomial selection, Chien search and the Forney algorithm (PCF) is provided. On this basis, a high-efficiency LCC decoder with adaptive error-correcting capability is proposed. Compared with the state-of-the-art LCC (TV = 16) decoding, the number of TVs of our decoder was reduced by 50.4% without loss of the frame error rate (FER) performance. The hardware implementation results show that the proposed decoder achieved 81.6% reduced average latency and 150% increased throughput compared to the state-of-the-art LCC decoder.

Keywords: Reed–Solomon (RS) codes; low-complexity chase (LCC); algebraic soft-decision decoding (ASD); multiplicity assignment; low latency; VLSI



Citation: Wang, H.; Zhang, W.; Chang, Y.; Gao, J.; Liu, Y.

Low-Complexity Chase Decoding of Reed–Solomon Codes Using Channel Evaluation. *Entropy* **2022**, *24*, 424. <https://doi.org/10.3390/e24030424>

Academic Editor: Jerry D. Gibson

Received: 13 February 2022

Accepted: 17 March 2022

Published: 18 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The error number of the codeword in the time-varying channel has great randomness, which makes it redundant to realize channel decoding with fixed decoding parameters. Therefore, channel evaluation plays an important role in channel decoding. It is of great significance to analyze the reliability of the channel in real-time by the received code sequence and automatically adjust the decoding parameters so that the channel decoder can maintain an efficient and low redundancy state on the premise of meeting the decoding performance.

Low-complexity chase (LCC) [1] is an excellent algebraic soft-decision decoding (ASD) algorithm for medium-to-high rate Reed–Solomon (RS) codes, which can achieve comparable error correction performance with other ASD algorithms, e.g., the Kötter–Vardy (KV) algorithm [2] and Bit-level Generalized Minimum Distance (BGMD) algorithm [3], while having lower complexity. As the main benefit of the LCC decoder, one level of multiplicity makes it possible to replace the interpolation and factorization stages by reformulated inversionless Berlekamp–Massey (RiBM) algorithm [4,5].

However, the LCC algorithm needs to judge the error symbol number of the codeword to dynamically adjust the decoding parameters for successfully decoding, otherwise it may cause a lot of redundant operations. The research work on the LCC algorithm mainly focuses on the selection of test vectors (TVs) and the design of high efficient decoders in recent years. A unified syndrome computation (USC)-based LCC decoder was proposed in [6]. The hardware of multiplicity assignment (MA) was implemented using the received bit-level magnitudes to evaluate the symbol reliability values [7].

However, the low hardware speed of this module limits the performance of the whole decoder. After that, an early termination algorithm for improving the throughput of the serial LCC decoder was proposed in [8]. In [9], a novel set of TVs derived from the analysis of the symbol error probabilities was applied to the modified LCC decoding. A LCC decoding algorithm using the module basis reduction (BR) interpolation technique [10], namely the LCC-BR algorithm, was proposed to reduce decoding complexity and latency. In addition, the number of 1 in the first syndrome S_0 can be used to infer whether the number of errors is even or odd, so as to reduce the number of TVs by half [11].

Thus far, there have been some preliminary studies on the decoding of time-varying channels. In [12], three different unreliable symbol position numbers η are used to realize channel environment adaptation; however, no specific channel evaluation scheme is given. An et al. [13] explored a classification decoding method based on deep learning to save decoding time. One of the most critical problems in the time-varying channel is decoding redundancy, which results in a large decoding latency. The decoding latency greatly limits the high-speed processing ability of the LCC decoder for massive data.

This study aimed to dynamically adjust the decoding parameters, reduce the decoding redundancy and improve the real-time communication performance by evaluating the number of errors in the received code sequence. The main contributions of the paper are summarized as follows.

(1) A time-varying channel adaptive LCC decoding algorithm is presented based on the channel soft information. It evaluates the time-varying channel environment and generates a suitable number of TVs and syndromes to reduce decoding redundancy and decoding delay.

(2) To improve the hardware performance of the multiplicity assignment (MA) block, we provide a simplified MA scheme and its hardware architecture. We also propose a new multi-functional block that can implement polynomial selection, Chien search and Forney algorithm (PCF) for saving hardware resources.

(3) A high-performance time-varying channel adaptive LCC decoder is provided. The decoding delay of the decoder is greatly reduced, and the hardware efficiency is significantly improved.

The rest of the paper is as follows. Section 2 introduces the conventional LCC decoding algorithm. The time-varying channel adaptive LCC algorithm is presented in Section 3. Section 4 introduces the simplified MA scheme, PCF block and the proposed LCC decoder. The implementation results are provided in Section 5. Section 6 draws the conclusion.

2. LCC Decoding Algorithm

The (n, k) RS codes over finite field $GF(2^m)$ are modulated by binary phase-shift keying (BPSK) and transmitted over an additive white Gaussian noise (AWGN) channel, where n is the code symbol length, k is the message symbol length, m denotes the number of bits per symbol and $n = 2^m - 1$. The field elements are $GF(2^m) = \{0, 1, \alpha, \alpha^2, \dots, \alpha^{n-1}\}$ with the primitive element α . The LCC decoder compares the reliability of each symbol r_i in one codeword at the MA stage. The reliability of the i -th received symbol is defined by first hard-decision value y_i^{HD} and the second hard-decision value y_i^{2HD} as

$$\Gamma_i = \lg \left[p(r_i | y_i^{HD}) / p(r_i | y_i^{2HD}) \right]. \quad (1)$$

The smaller Γ_i is, the less reliable the symbol is. The hard-decision y_i^{HD} and y_i^{2HD} can be obtained by flipping the bit with the lowest reliability (i.e., the minimum level value). η unreliable symbols with the smallest values of Γ_i are selected to generate multiple TVs that are most likely to achieve successful decoding. 2^η TVs can be obtained by combining y_i^{HD} or y_i^{2HD} of the η unreliable symbols. If there exists a TV whose error symbol number is less than the decoding radius t in the TV set Γ (TV1, TV2, ..., TV 2^η), where $t = \frac{n-k}{2}$, the decoding can succeed.

3. Time-Varying Channel Adaptive LCC Algorithm

In the time-varying channel, the signal-to-noise ratio (SNR) changes rapidly with time, which makes the LCC decoding algorithm with fixed parameters cause a great deal of decoding redundancy and delay. To solve this problem, we propose a novel LCC decoding algorithm in Algorithm 1 to reduce the number of TVs and syndromes, and the flowchart is shown in Figure 1. The main idea of the algorithm is to evaluate the error symbol number of each codeword by counting the number of unreliable bits in order to dynamically adjust the number of TVs and the number of syndromes used to achieve successful decoding.

In step 1, the error symbol number and unreliable value of the codeword are evaluated by the received bit-level value $r_{i,j}$, where i is the symbol position and j is the bit position. The number of bits in each codeword whose bit-level value is less than the threshold Th1 (set to 0.3 in this paper) is counted. For each symbol, the smallest bit γ_i is selected, and the unreliable bit position set Δ is updated.

In step 2, two modes are set, one is the syndrome adaptive adjustment, and the other is the TV adaptive adjustment. First, if the syndromes calculated from the first test vector TV1 are all zero, then the codeword is correct. Otherwise, if the unreliable bit number (i.e., counter) is less than Th2, mode 1 is performed. The threshold Th2 used to distinguish $e \leq t$ and $e > t$ is the number of bits whose level values are less than Th1, where e is the number of error symbols.

If counter \leq Th2, which indicates that the number of error symbols does not exceed t , hard decision decoding (HDD) can be adopted, otherwise, the LCC decoding can be selected. On the premise of ensuring decoding performance, Th2 can be taken as data equal to $2t$. For the standard (255,239) RS codes, Th2 values can be 16 as shown in Table 1. If some performance is lost in exchange for decoding speed, the value of Th2 can be appropriately increased. The syndrome S , error location β and error value δ satisfy the following equations:

$$\begin{aligned} S_0 &= \delta_1\beta_1 + \delta_2\beta_2 + \delta_3\beta_3 + \dots + \delta_e\beta_e \\ S_1 &= \delta_1\beta_1^2 + \delta_2\beta_2^2 + \delta_3\beta_3^2 + \dots + \delta_e\beta_e^2 \\ &\vdots \\ S_{2t-1} &= \delta_1\beta_1^{2t} + \delta_2\beta_2^{2t} + \delta_3\beta_3^{2t} + \dots + \delta_e\beta_e^{2t}. \end{aligned} \tag{2}$$

When $e < t$, only the first $2e$ syndromes need to be calculated to get the error vector. The number of equations to be solved is reduced from $2t$ to $2e$, which can reduce the redundant calculation of $2t - 2e$ equations. If counter $>$ Th2, we should make full use of the channel soft information to achieve a larger decoding radius. The selected number of TVs can be determined by the counter, which is obtained by statistical regulation. Table 1 shows the corresponding relationship between the counter value of the received (255,239) RS code and the maximum possible number of error symbols and gives the corresponding values of t_a and the number of TVs. Table 2 shows the parameter correspondence of (127,119) RS code.

Table 1. The corresponding relationship between the counter value of received RS(255,239) code and the decoding parameters.

Counter	2	4	6	8	10	12	14	16	24	36	>36
Error num.	1	2	3	4	5	6	7	8	9	10	>10
t_a	1	2	3	4	5	6	7	8	8	8	8
TV num. ζ	1	1	1	1	1	1	1	1	4	8	16

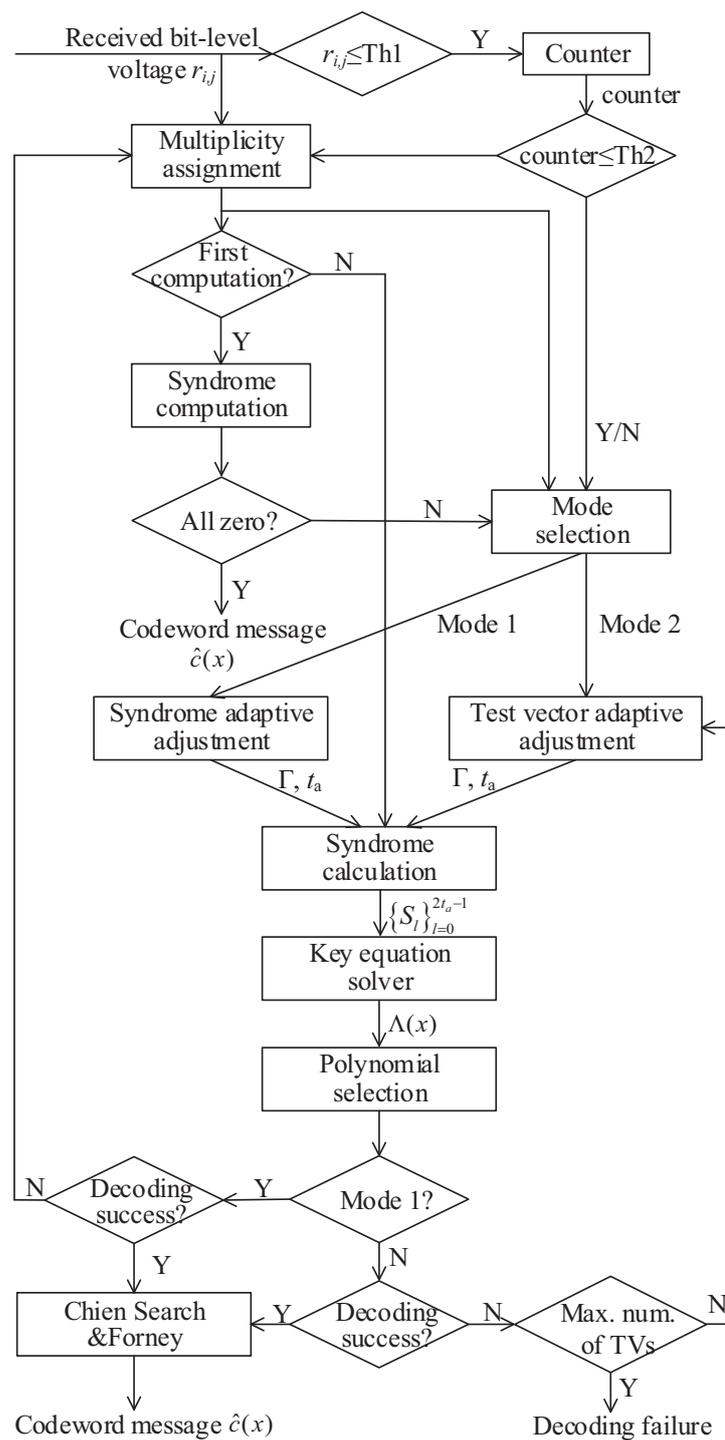


Figure 1. Data flow of time-varying channel adaptive LCC decoding.

Table 2. The corresponding relationship between the counter value of received RS(127,119) code and the decoding parameters.

Counter	2	4	6	8	14	>14
Error num.	1	2	3	4	5	>5
t_a	1	2	3	4	4	4
TV num. ζ	1	1	1	1	4	8

Algorithm 1: Time-Varying Channel Adaptive LCC Algorithm**Input:** received bit-level voltage $r_{i,j}$, $i(0 \leq i \leq n - 1)$, $j(0 \leq j \leq m - 1)$ **Output:** decoded codeword message $\hat{c}(x)$ **initialize:** counter = 0, unreliable bit set $\Sigma = \emptyset$, unreliable bit position set $\Delta = \emptyset$, number of unreliable positions p ;**Step 1:** time-varying channel evaluation**for** $i = 0$ to $n - 1$ **do** **for** $j = 0$ to $m - 1$ **do** **if** $r_{i,j} \leq Th1$ **then**

counter++;

$$y_{i,j}^{HD} = \begin{cases} 0, & r_{i,j} \leq 0 \\ 1, & r_{i,j} > 0 \end{cases};$$

 $\gamma_i = \min\{|r_{i,j}|\}$; Add γ_i to the index set Σ and keep the smallest p elements, update bit position set Δ .**Step 2:** mode selection**if** all syndromes of TV1 are zero **then**

declare decoding success.

else if counter $\leq Th2$ **then**

perform Mode 1.

else

perform Mode 2.

Mode 1: syndrome adaptive adjustmentThe adjusted decoding radius t_a is determined by the number of possible error symbols. The TV index set $\Gamma = \{\text{TV1}\}$ containing TV1 is obtained by hard decision.**Mode 2:** test vector adaptive adjustmentSet t_a as t and select the proper number ζ of TVs according to the number of counter. Obtain the TV index set $\Gamma = \{\text{TV1}, \text{TV2}, \dots, \text{TV}\zeta\}$ according to the selection order of the TVs.**Step 3:** decoding process**for** $b = 1$ to σ (σ is the number of TVs); **do** select or compute syndromes $\{S_l\}_{l=0}^{2t_a-1}$ of TV_b ; apply the Berlekamp–Massey algorithm with input $\{S_l\}_{l=0}^{2t_a-1}$ to calculate error locator polynomial $\Lambda(x)$; **if** the root number of $\Lambda(x)$ equals its degree **then** correct test vector TV_b and declare decoding success; break; **else**

declare decoding failure.

Step 4: error mode update**if** decoding failure and $t_a < t$ **then** calculate error locator polynomial $\Lambda(x)$ from the $2t$ syndromes $\{S_l\}_{l=0}^{2t-1}$ of TV1. **if** the root number of $\Lambda(x)$ equals its degree **then** correct test vector TV_1 and declare decoding success; break; **else**

declare decoding failure.

In addition, in order to find the TV that can be successfully decoded with fewer attempts, we introduce two TV selection and sorting methods. After determining the TV1, we need to find the TV2 based on the idea of decoding complementarity, then find the TV3 complementary to their decoding based on the previous two TVs and constantly repeat the above rules to obtain the TV index set $\Gamma = \{TV_1, TV_2, \dots, TV_\zeta\}$. It can be concluded from [11] that when the signal to noise ratio (SNR) exceeds 6.1 dB, the probability that the symbol error has a 1-bit error is above 99%.

The first syndrome is $S_0 = R(\alpha^0) = \sum_{l=1}^e E_l$, where $R(x)$ represents the received codeword and E_l is the error vector. According to the parity number of 1 contained in S_0 and the idea of compensation decoding, two sets of index sets, odd index set $\Gamma_o = \{TV_{o1}, TV_{o2}, \dots, TV_{o\zeta}\}$ and even index set $\Gamma_e = \{TV_{e1}, TV_{e2}, \dots, TV_{e\zeta}\}$, are tested. Table 3 shows the sequence of 16 TVs of decoding (255,239) RS codes obtained through simulation, and each TV is the best compensation for all previous TVs. The order of eight TVs for decoding (127,119) RS codes is given in Table 4. In addition, the corresponding optimal test order when the number of 1 in S_0 is odd or even is given.

Table 3. The selection order of test vectors for RS(255,239) codes.

Testing Order	TV1	TV2	TV3	TV4
Flipping pattern	00000000	11111000	10000110	11100100
Testing order	TV5	TV6	TV7	TV8
Flipping pattern	01011011	10100011	01110000	00111010
Testing order	TV9	TV10	TV11	TV12
Flipping pattern	11001001	00010101	10000000	01001110
Testing order	TV13	TV14	TV15	TV16
Flipping pattern	00101001	11010010	01000101	10011100

Table 4. The selection order of test vectors for RS(127,119) codes.

Testing Order	TV1	TV2	TV3	TV4
Flipping pattern	00000000	11100000	10011000	01000110
Testing order	TV5	TV6	TV7	TV8
Flipping pattern	00110000	00101001	11000101	10000010
Testing order	TV _e 1	TV _e 2	TV _e 3	TV _e 4
Even Flipping pattern	00000000	11100000	10011000	01010100
Testing order	TV _e 5	TV _e 6	TV _e 7	TV _e 8
Even Flipping pattern	00100011	10101100	01001000	10000011
Testing order	TV _o 1	TV _o 2	TV _o 3	TV _o 4
Odd Flipping pattern	00000000	11100000	10011000	00000100
Testing order	TV _o 5	TV _o 6	TV _o 7	TV _o 8
Odd Flipping pattern	01110010	11000101	10101010	00110001

In step 3, the $(n, n - 2t)$ RS code is regarded as $(n, n - 2t_a)$ RS code, and the first $2t_a$ syndromes are used to complete decoding. The number of iterations for solving the key equation changed from $2t$ times to $2t_a$ times, reducing $2(t - t_a)$ iterations. Most of the corresponding relations are reliable; however, there are a few counter values that can not accurately reflect the error symbol number of codewords. Therefore, we introduce the error mode update method in step 4 to reduce the impact of the above problems on the decoding performance. The decoding failed codewords in the syndrome adaptive adjustment are corrected by the $2t$ syndromes $\{S_l\}_{l=0}^{2t-1}$ of TV1 again.

The simulation results of the proposed LCC algorithm for (255,239) RS code are given and compared with several previous algorithms in Figure 2. The proposed LCC algorithm reduces the decoding complexity by dynamically adjusting the number of TVs. Compared with the current state-of-the-art LCC decoding algorithms [9–11], the proposed LCC decoding algorithm has achieved equivalent or better decoding performance with fewer TVs. The amount of computation of our algorithm is greatly reduced. We simulate

the time-varying channel, which randomly changes with frame in the range of [6.5, 8] dB and simulate the decoding performance of 800,000 codewords through different algorithms.

We give the total number and average number of TVs required to decode these codewords. As shown in Table 5, compared with the cases of TV = 8 and TV = 16 in [9], the proposed LCC decoding algorithm achieves better decoding performance while reducing TVs by 23.5% and 50.4%, respectively. Compared with the case of TV = 8 in [11], the proposed LCC decoding algorithm achieves a performance gain of 43.1% when the number of TVs is approximately equal.

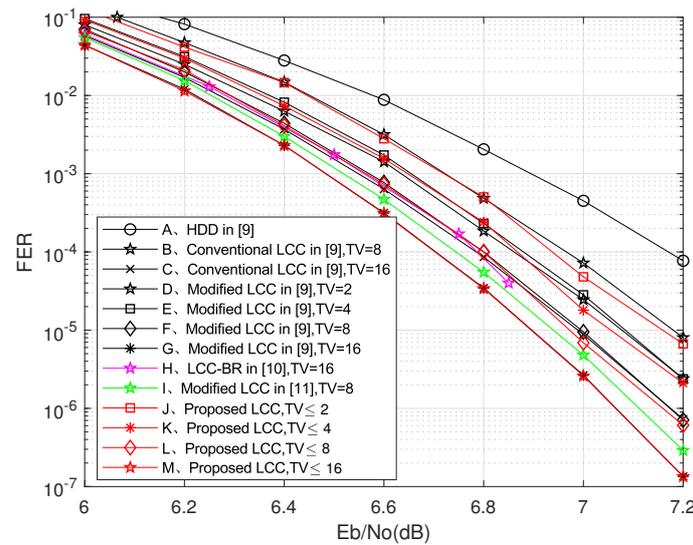


Figure 2. Frame error rate (FER) versus Eb/No of the proposed decoder compared with other algorithms for RS(255,239) codes.

Table 5. Comparison of the decoding performance for RS(255,239) codes.

Algorithm	Num. of TVs	Avg. Num. of TVs	FER
Conv. LCC in [9], TV = 8	6400 k	8	3.17×10^{-4}
Conv. LCC in [9], TV = 16	12,800 k	16	1.66×10^{-4}
Modified LCC in [9], TV = 8	6400 k	8	1.95×10^{-4}
Modified LCC in [9], TV = 16	12,800 k	16	7.96×10^{-5}
Modified LCC in [11], TV = 8	6400 k	8	1.34×10^{-4}
Proposed LCC TV ≤ 8	4899 k	6	1.74×10^{-4}
Proposed LCC TV ≤ 16	6347 k	8	7.63×10^{-5}

Figure 3 shows the decoding performance of the proposed decoder with other algorithms for (127,119) RS codes. The proposed scheme obtains better performance than the traditional LCC algorithm with fewer TVs. When the channel conditions are poor, the performance of the LCC decoding scheme based on a single test set is better because the probability of multi-bit errors in the symbol is greater when the channel conditions are poor, which makes the error parity judgment confused. However, when the channel conditions are good, the performance of the LCC decoding scheme based on the parity test set is better. Comparisons in the time-varying channel of [6.5, 8] dB are also provided. As shown in Table 6, the proposed LCC decoding algorithm reduces the number of TVs by 52.4%, which also achieves more competitive decoding performance than the traditional LCC decoding algorithm.

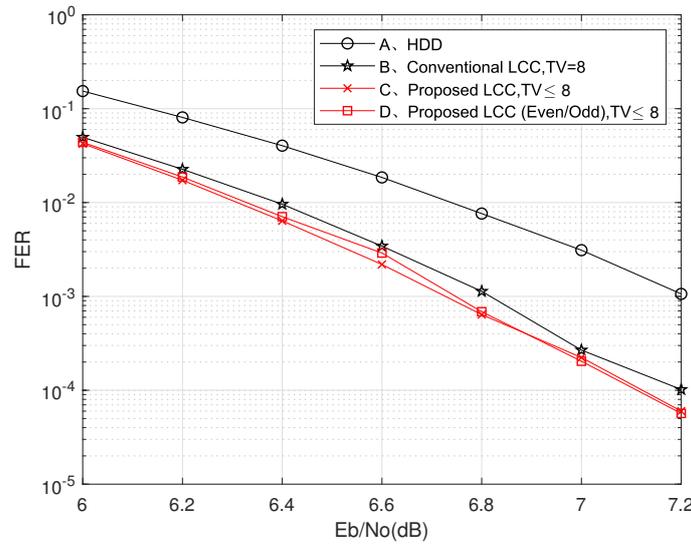


Figure 3. Frame error rate (FER) versus E_b/N_0 of the proposed decoder compared with other algorithms for RS(127,119) codes.

Table 6. Comparison of the decoding performance for RS(127,119) codes.

Algorithm	Conv. LCC, TV = 8	Proposed LCC, TV ≤ 8	Proposed LCC (Even/Odd), TV ≤ 8
Num. of TVs	6400 k	3049 k	3049 k
Avg. Num. of TVs	8	3.8	3.8
FER	7.88×10^{-4}	6.71×10^{-4}	6.76×10^{-4}

4. The Proposed Time-Varying Channel Adaptive LCC Decoder

This section presents the architecture of the proposed time-varying channel adaptive LCC decoder, which processes up to 16 test vectors in the set. As shown in Figure 4, the proposed decoder consists of four blocks: 16-parallel MA with channel evaluation, 16-parallel syndrome computation (SC), key equation solver (KES), 16-parallel PCF block. An $n \times m$ size RAM is used to cache the codeword.

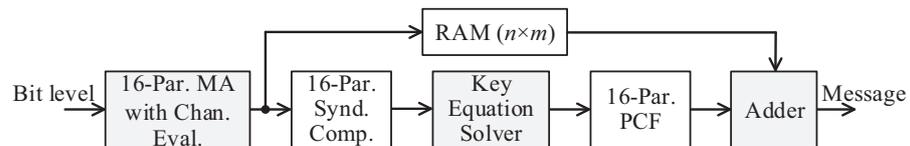


Figure 4. Block diagram for the time-varying channel adaptive LCC decoder.

The decoding chronogram of the proposed decoder is shown in Figure 5. As the timing diagram shows, when the 16-parallel MA block generates the TVs, the 16-parallel SC block only delays one clock cycle to start working, and both blocks need 16 clock cycles.

Thus, it only takes 17 clock cycles from the input of the first codeword A to the output of the first TV’s syndromes. It takes 16 clock cycles for the KES block to compute the error locator polynomial $\Lambda(x)$ and the error magnitude polynomial $\Omega(x)$ through the RiBM algorithm. At the same time, the 16-parallel SC block calculates the syndromes of the second TV. Then, the 16-parallel PCF block completes PS and Chain search (CS) after 16 clock cycles. If the order of the error location polynomial is lower than t , the decoder will complete the calculation of Forney algorithm by reusing the PCF block, which takes 16 clock cycles.

Otherwise, it will continue to judge the next TV. Since the first TV is the best hard decision value of the received codeword, it can be successfully decoded in most cases. If all

the syndromes are zero, the TV will be output directly. If $t_a < t$, the KES block only needs $2t_a$ clock cycles to calculate the equation; however, other blocks still need 16 clock cycles to complete their functions.

The latency of the proposed LCC decoder is $17 + 16 \times i + 16$ cycles. When i (i.e., the ordinal number of the TV) takes the maximum value of 16, the maximum latency of the proposed decoder is $17 + 16 \times 16 + 16 = 289$ cycles. When the minimum value of i is 1, the minimum latency of the proposed decoder is $17 + 16 \times 1 + 16 = 49$ cycles.

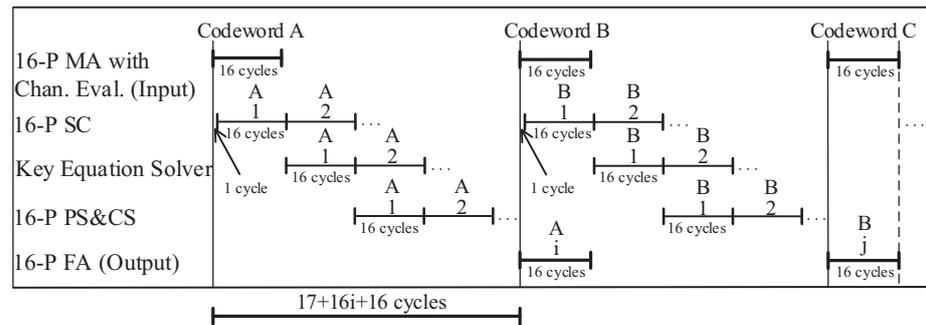


Figure 5. The timing diagram of the proposed LCC decoder.

4.1. Multiplicity Assignment with Channel Detection

The existing MA scheme [7] needs to realize $\log_2 m$ times pairwise comparisons through the comparator to select the most unreliable bit of each symbol, which greatly limits the hardware performance. To solve this issue, we propose a simplified MA scheme by making full use of the channel soft information. If the level value $r_{i,j}$ is less than Th1, it is considered as the most unreliable bit γ_i of this symbol.

$$\gamma_i = r_{i,j} \tag{3}$$

Otherwise, the first bit $r_{i,0}$ in the symbol is considered unreliable.

$$\gamma_i = r_{i,0} \tag{4}$$

Statistics show that the probability of two or more unreliable bits in a symbol is very small. If multiple unreliable bits appear in a symbol, the last bit less than Th1 is selected as the unreliable bit. Each update of the MA block keeps only the first and second decision values of the p most unreliable elements and their location information.

The simulation results in Table 7 show that the proposed MA block can achieve the maximum clock frequency of 385 MHz in SMIC 0.13 μm CMOS technology, whose throughput is 75% higher than the MA in [7]. In addition, the hardware area and power consumption are also reduced. A p -degree parallel MA architecture including p ($p = 16$) comparators is shown in Figure 6, which only needs $\lceil 255/p \rceil$ clock cycles to complete MA.

Table 7. The implementation results of the proposed MA module in 0.13 μm CMOS process at 200 MHz.

Module	Area (mm ²)	Gate Count (XORs)	f_{max} (MHz)	Throughput (Gb/s)	Power (mW)
MA [7]	0.0661	5561	220	1.76	6.0
Proposed	0.0625	5261	385	3.08	5.6

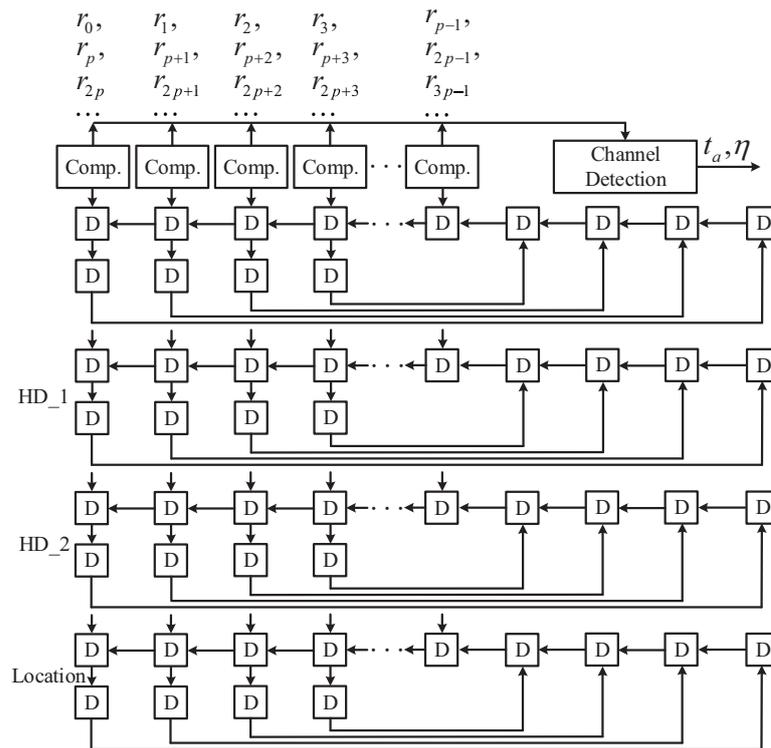


Figure 6. The architecture of the proposed MA.

4.2. The Architecture of the Proposed Pcf Block

The proposed p -parallel PCF block shown in Figure 7 is an upgraded module of [14], which only needs 16 clock cycles to realize PS and Chein search at the same time. If the polynomial is selected correctly, the block can reuse the basic unit to complete the p -way ($p = 16$) Forney algorithm.

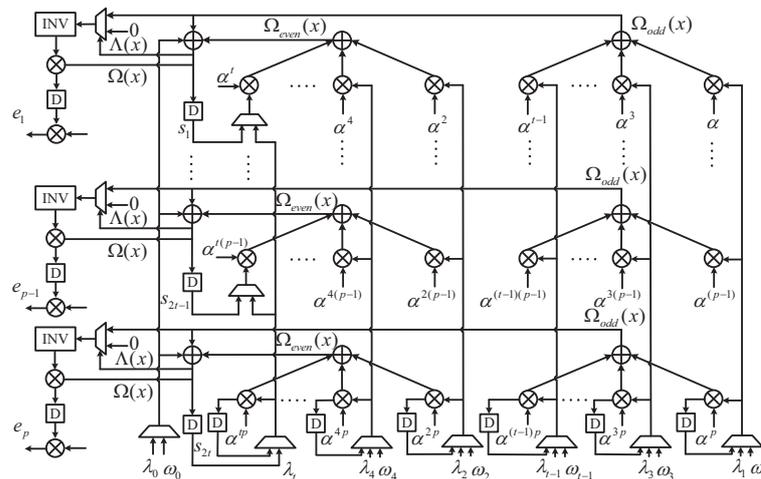


Figure 7. The architecture of the proposed PCF.

5. Implementation Results

The proposed time-varying channel adaptive LCC decoder of (255, 239) RS code was implemented using 0.13 μm and 65 nm CMOS process. As shown in Table 8, the maximum clock frequency of the proposed decoder is 385 MHz in 0.13 μm process and 550 MHz in 65 nm process by Synopsys design tools. Compared with the current state-of-the-art LCC decoders [9,11], the proposed LCC decoder achieves equivalent or better coding gain. However, the high clock frequency and additional hardware resources also increase the power consumption of the proposed decoder.

Table 8. Implementation results of the LCC decoders.

Architecture	[7]	[9]	[11]	Proposed	Proposed
Tech.	0.13 μm	65 nm	65 nm	0.13 μm	65 nm
f_{max} (MHz)	220	550	550	385	550
Power (mW@MHz)	-	23.0@550	21.5@550	259.2@385	31.5@550
Throughput (Gbps)	1.6a	4.4a	8.8a	15.7b	22.4b
Gate count (kXORs) with buffer	27.9	31.1	34.9	55.4	56.8
Coding gain (dBs@FER)	0.37@ 10^{-6}	0.56@ 10^{-6}	0.50@ 10^{-6}	0.56@ 10^{-6}	0.56@ 10^{-6}
Latency (clock cycles)	400a	528a	272a	50b	50b

a Fixed value b Average value.

For the time-varying channel with the range of [6.5, 8] dB, the proposed decoder requires an average of 1.06 TVs to complete decoding. The average latency of the decoder is $17 + 1.06 \times 16 + 16 = 50$ cycles, which is 81.6% smaller than that of the LCC decoder in [11]. The average throughput of the proposed decoder can be calculated by formula (5).

$$\text{Throughput} = \frac{\# \text{ of bits processed each time}}{\# \text{ of clock cycles needed} \times T_{\text{min}}} \quad (5)$$

The estimated throughput is $(255 \times 8 \times 385)/50 = 15.7$ Gb/s in 0.13 μm process. The estimated throughput is $(255 \times 8 \times 550)/50 = 22.4$ Gb/s in the 65 nm process.

Compared with [7,9,11], the throughput (5) of the proposed decoder is increased by 8.8, 4.1 and 1.5 times, respectively. Compared with the state-of-the-art LCC decoder, the proposed decoder has lower latency and higher throughput, which indicates that the proposed decoder has better hardware performance.

6. Conclusions

In this paper, we derived a time-varying channel adaptive LCC algorithm, which can dynamically adjust the decoding parameters through channel detection. The proposed algorithm can greatly reduce the redundant TVs and key equations, thus, reducing the decoding complexity and latency. To evaluate the algorithmic performance, we performed effect analysis in the time-varying channel.

Compared to the state-of-the-art LCC algorithm, the number of TVs required was reduced by 50.4%, and the average latency of the decoder was reduced by 81.6%. To reduce the pairwise comparison times of selecting the most unreliable bit for each symbol in the MA block, a simplified scheme assisted by setting a bit-level threshold was proposed to improve its hardware efficiency. In addition, a PCF block was proposed to reduce the consumption of hardware resources.

Based on the above techniques, the hardware architecture of the proposed time-varying channel LCC decoder is presented. Compared to previously published papers, the implementation results show that the proposed LCC decoder achieved at least 150% throughput improvement. Future work will focus on high-performance time-varying LCC decoders.

Author Contributions: All authors have made great contributions to the work. Conceptualization, H.W., W.Z. and Y.C.; Software, H.W., Y.C. and J.G.; Writing—original draft, H.W.; Writing—review and editing, W.Z., Y.C., J.G. and Y.L.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Quanzhou Tianjin University Institute of Integrated Circuits and Artificial Intelligence Open Research Fund Project.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the reviewers for their constructive comments that have helped improving the overall quality of the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bellorado, J. Low-Complexity Soft Decoding Algorithms for Reed–Solomon Codes. Ph.D. Thesis, Harvard University, Cambridge, MA, USA, 2006.
2. Koetter, R.; Vardy, A. Algebraic soft-decision decoding of Reed–Solomon codes. *IEEE Trans. Inf. Theory* **2003**, *49*, 2809–2825. [[CrossRef](#)]
3. Jiang, J.; Narayanan, K.R. Algebraic soft-decision decoding of Reed–Solomon codes using bit-level soft information. *IEEE Trans. Inf. Theory* **2008**, *54*, 3907–3928. [[CrossRef](#)]
4. García-Herrero, F.; Valls, J.; Meher, P. K. High-speed RS(255, 239) decoder based on LCC decoding. *Circuits Syst. Signal Process.* **2011**, *30*, 1643–1669. [[CrossRef](#)]
5. Sarwate, D. V.; Shanbhag, N.R. High-speed architectures for Reed–Solomon decoders. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2001**, *9*, 641–655. [[CrossRef](#)]
6. Zhang, W.; Wang, H.; Pan, B. Reduced-complexity LCC Reed–Solomon decoder based on unified syndrome computation. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2013**, *21*, 974–978. [[CrossRef](#)]
7. Peng, X.; Zhang, W.; Ji, W.; Liang, Z.; Liu, Y. Reduced-complexity multiplicity assignment algorithm and architecture for low-complexity chase decoder of Reed–Solomon codes. *IEEE Commun. Lett.* **2015**, *19*, 1865–1868. [[CrossRef](#)]
8. Luo, H.; Zhang, W.; Wang, Y.; Hu, Y.; Liu, Y. An algorithm for improving the throughput of serial low-complexity chase soft-decision Reed–Solomon decoder. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 3539–3542. [[CrossRef](#)]
9. Valls, J.; Torres, V.; Canet, M.J.; García-Herrero, F.M. A test vector generation method based on symbol error probabilities for low complexity chase soft-decision Reed–Solomon decoding. *IEEE Trans. Circuits Syst. I Reg. Papers* **2019**, *66*, 2198–2207. [[CrossRef](#)]
10. Xing, J.; Chen, L.; Bossert, M. Low-Complexity Chase Decoding of Reed–Solomon Codes Using Module. *IEEE Trans. Commun.* **2020**, *68*, 6012–6022. [[CrossRef](#)]
11. Jeong, J.; Shin, D.; Shin, W.; Park, J. An Even/Odd Error Detection Based Low-Complexity Chase Decoding for Low-Latency RS Decoder Design. *IEEE Commun. Lett.* **2021**, *25*, 1505–1509. [[CrossRef](#)]
12. Zhu, J.; Zhang, X. Efficient Reed–Solomon decoder with adaptive error-correcting capability. In Proceedings of The 19th Annual Wireless and Optical Communications Conference (WOCC 2010), Shanghai, China, 14–15 May 2010, pp. 1–5. [[CrossRef](#)]
13. An, X.; Liang, Y.; Zhang, W. High-Efficient Reed–Solomon Decoder Based on Deep Learning. *IEEE Int. Symp. Circuits Syst. (ISCAS)* **2020**, 1–5. [[CrossRef](#)]
14. Li, X.; Zhang, W.; Liu, Y. Efficient architecture for algebraic soft-decision decoding of Reed–Solomon codes. *IET Commun.* **2015**, *9*, 10–16. [[CrossRef](#)]