Error Control Scheme for High-Speed DVD Systems

Joohyun Lee[†], Jaejin Lee[*], and Taegeun Park[**],

Key Words : DVD, error control coding, EFMPlus code, RSPC, erasure decoding.

ABSTRACT

We present a powerful error control decoder which can be used in all of the commercial DVD systems. The decoder exploits the error information from the modulation decoder in order to increase the error correcting capability. We can identify that the modulation decoder in DVD system can detect errors more than 60% of total errors when burst errors are occurred. In results, for a decoded block, error correcting capability of the proposed scheme is improved up to 25% more than that of the original error control decoder. In addition, the more the burst error length is increased, the better the decoder performance. Also, a pipeline-balanced RSPC decoder with a low hardware complexity is designed to maximize the throughput. The maximum throughput of the RSPC decoder is 740Mbps@100MHz and the number of gate counts is 20.3K for RS (182, 172, 11) decoder and 30.7K for RS (208, 192, 17) decoder, respectively.

1. Introduction

The digital versatile disc (DVD) is an advanced optical recording system with a storage capacity seven times more than the compact disc (CD). The storage capacity can be increased by the redesign of the logical format of the disc with a more powerful error correction code and a high-rate modulation code [1]. Modulation code is used for preventing intersymbol interference (ISI) and that of DVD system is called *EFMPlus code* [1][2]. When the minimum and maximum run-lengths denote d and k, respectively, all codewords of the EFMPlus code satisfy (d=2, k=10)-constraints. In addition, the code rate is 8/16. Error correction codes of data storage systems are mainly used for information data sequence in order to prevent burst errors. *Reed-Solomon Product Code* (RSPC) is the unique code for correcting random and burst errors in DVD systems [4]. When RS (n, k, d_{min}) code with minimum distance, d_{min}, contains k message symbols and $n-k$ parity symbols, the RSPC is composed of RS (182, 172, 11) code, which is called *inner code*, in the row direction and RS (208, 192, 17) code, which is called outer code, in the column direction.

As the reading and writing speeds are increased, the amplitudes of the readback signals of optical recording channel are remarkably decreased. Therefore, a number of errors can be generated by the low signal-to-noise ratio (SNR). Also, reasonable data reading process can be disturbed by the defects of pick-up system, optical lens or disc. However, in DVD systems, the correction of these errors is done by RSPC only. Each row of the inner decoder in RSPC can correct up to 5 byte errors. Since the codeword is recorded on the disc row by row, burst errors may occur in the direction of rows [4]. If the number of rows that have more than 5 byte errors is greater than 16, the errors cannot be corrected because the maximum error correcting capability in each column of the outer decoder is 16 bytes.

In this paper, to increase the error correcting ability of the RSPC, we use the decoding characteristic of the EFMPlus modulation code. The EFMPlus code can easily detect burst errors. Therefore, if we can exploit the detected error information in the RSPC decoder, the maximum number of correctable symbol errors per codeword is 10 bytes at the inner decoder. In addition, for random errors, the performance of the proposed RSPC decoder is also improved.

In Section II, we propose an improved RSPC decoder, which exploits the error detecting characteristic of the EFMPlus code, and discuss the simulation results using

[†] Department of Electronic Engineering, Dongguk University,
E-mail : xmas@dongguk.edu
* school of Electronic Engineering, Soongsil University
** School of Information, Communication, and Electronics Engineering, The Catholic University of Korea

modified RSPC decoder in Section III. In Section IV, we present the proposed RSPC decoder architecture. Finally, conclusion is given in Section V.

2. Decoding Rules of EFMPlus Code and RSPC

2.1 Error Detecting Characteristic of EFMPlus Decoder

The EFMPlus code is consisted of a finite-state sequential machine with 8-bit inputs, 16-bit codewords and four states. Each codeword satisfies (d=2, k=10)-constraints. Each of the four states is characterized by the type of codewords that enter or leave the given state. In detail, codewords in State 1 start with a run-length of at least two and at most nine 'zeros,' and codewords in State 4 start with at most one 'zero.' Obviously, the sets of codewords in State 1 and 4 have no duplicate codewords. The codewords in State 2 satisfy that the most significant bit (MSB) and the thirteenth bit are both 'zero,' and codewords in State 3 satisfy that the addition of these two elements is not equal to 'zero.'

The decoder translates 16-bit codewords into 8-bit source words. The source words are assigned such that decoding of the channel codewords can be uniquely accomplished without knowledge of the encoder state. Sometimes, two source words have the same channel representation. These words cannot be decoded by a sole observation of the 16-bit codeword, and the next state of these codewords is State 2 or 3. This ambiguity can be solved by observing the upcoming codeword. As the sets of codewords leaving State 2 or 3 are chosen such that they can be distinguished by the first and thirteenth bits of each codeword, codewords can be uniquely decoded by observing a 16-bit codeword and the first and thirteenth bits of the upcoming codeword.

The output of modulation decoder for channel outputs can be classified by three cases: (1) correct symbol, (2) undetected error symbol and (3) detected error symbol. The EFMPlus encoding and decoding are accomplished by the look-up table with (d=2, k=10)-constraints. In addition, the maximum numbers of error propagation are two for a codeword error. Therefore, one or two errors can be detected if the codeword doesn't exist in the look-up table of the EFMPlus decoder.

Table I lists the numbers of different and duplicate codewords in the EFMPlus look-up table. From the table, the number of duplicate codewords is more than 90 in each state, and each state is decided by the number of trailing zeros of the codeword entered during the decoding process. If the (d, k)-constraints are not considered, the number of feasible 16-bit sequence is 2^{16}. Therefore, if random errors are occurred, the probability that can detect the errors is very high because the number of the codewords is finite.

Table II shows error detection probabilities in each state of the EFMPlus look-up table when an error occurs at the current or the next codeword. Assume that (d, k)-constraints are satisfied although an error is occurred. In this case, the total number of available codewords with 16-bit lengths is 566. Therefore, the minimum error detecting probability is 0.55 (= (566−254)/566) for the current codeword error because the number of different codewords is less than 255 in each state. When an error exists at the next codeword, the detecting ability for the current codeword's error is more than 26%.

Despite of decoding at State 1, if the first and second bits of the codeword are not 'zeros' by swapping between State 1 and 4, we search the outputs between 88 and 255 at State 4. Similarly, if the first and second bits are all 'zeros' at the decoding location of State 4, we seek the outputs at State 1. In these cases, the number of different codewords between 88 and 255 for each state is exactly 128, and the error detecting probability by swapping between the states is more than 77%.

However, we can identify that the ratio of swapping the states is less than 2.5% for EFMPlus encoding process of RSPC coded data as shown in Fig. 1. Therefore, the errors from state swapping do not affect total error detecting probability.

Based on the results, we simulated error detecting characteristics for several error patterns as shown in Fig. 2. To compare error detection ratios by changing the error rates of EFMPlus decoded symbols, we measured the ratios when the symbol error rates (SERs) are approximately 5.0×10^{-2} and 1.0×10^{-4}, respectively. From the figure, regardless of the error rates, the detectable error ratio for random errors is more than 1/3 of total errors. However, when burst errors are occurred, the minimum detectable error ratio is more than 60%. Furthermore, the ratio is increased according as the length of burst error is increased. Therefore, if we can exploit the error detecting information in the RSPC decoder, the number of correctable symbol errors can be increased.

TABLE I
NUMBERS OF DIFFERENT AND DUPLICATE CODEWORDS

	State 1	State 2	State 3	State 4
# of different codewords	250	254	243	254
# of duplicate codewords	94	90	101	90

TABLE II
ERROR DETECTION PROBABILITIES IN EFMPLUS DECODER

Location of generated error		Error detection probability			
		State 1	State 2	State 3	State 4
Current codeword	No swapping	≥ 0.56	≥ 0.55	≥ 0.57	≥ 0.55
	Swapping	> 0.77	-	-	> 0.77
Next codeword		≥ 0.27	≥ 0.26	≥ 0.29	≥ 0.26

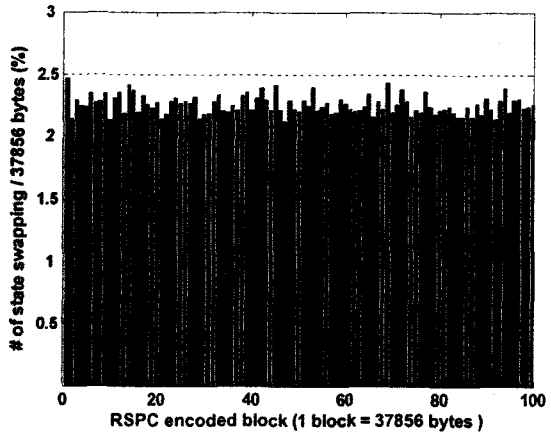


Fig. 1. Ratio of swapping between State 1 and 4. The ratio of swapping the states is less than 2.5% for EFMPlus encoding process.

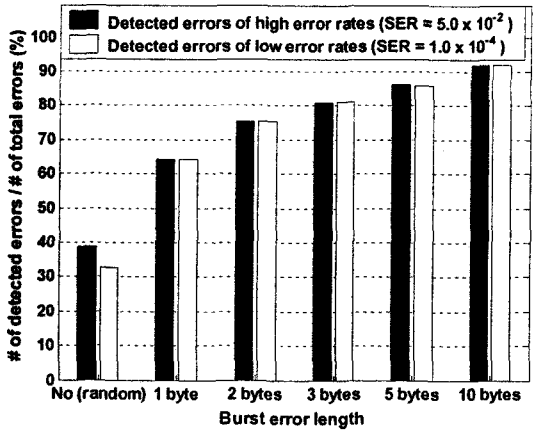


Fig. 2. Error detecting characteristics for several error patterns. For burst errors, the minimum detectable error ratio is more than 60% regardless of the error rates.

2.2 RSPC with Inner Erasure Decoder

Product codes are generated by using two linear block codes arranged in a matrix form [5]. When two linear block codes, one with parameters (n_1, k_1, $d_{\min,1}$) and the other with parameters (n_2, k_2, $d_{\min,2}$), are used in a matrix, the minimum distance of the resulting code is the product of the minimum distances of the component codes as $d_{\min}=d_{\min,1}\,d_{\min,2}$. Therefore, the resulting code is an ($n_1 n_2$, $k_1 k_2$, $d_{\min,1}\,d_{\min,2}$) linear block code. It can be shown that the correcting capability of the code is as follows.

$$t=\left\lfloor\frac{d_{\min,1}d_{\min,2}-1}{2}\right\rfloor=\left\lfloor\frac{n_1n_2-k_1k_2}{2}\right\rfloor$$

where $\lfloor x\rfloor$ means the largest integer not to exceed x. However, in practice, the error correcting capability can be increased by using the erasure information.

The information field of RSPC block used in DVD is composed of 172 bytes × 192 rows. For encoding, first, 16 bytes of the parity on outer code (PO) are attached to each of the 172 columns by the RS (208, 192, 17) code. Next, 10 bytes of the parity on inner code (PI) are attached to each of the 208 rows by the RS (182, 172, 11) code. The RSPC encoded sequences are stored row by row on disc after EFMPlus encoding. Therefore, bursts of errors may occur in the direction of rows.

Decoding of the original RSPC consists of two steps. First, the decoding of the inner RS (182, 172, 11) code is performed. The inner decoder can correct errors up to 5 symbols in each row. Therefore, while most of random and short burst errors may be corrected in the inner decoder, the decoding failure is generated for long burst errors in each row. For these uncorrected inner codewords, we declare erasures which inform the locations of errors to outer RS decoder. In the second step, using these erasures, the decoding of the outer RS (208, 192, 17) code is performed. Since the minimum distance of the outer RS code is 17, the outer decoder can correct errors and erasures up to $\varepsilon+2\nu\le 16$, where ε and ν are the number of erasures and the number of errors, respectively.

When the number of errors that exceeds 5 bytes in each row is more than 16, the errors cannot be corrected because the maximum error correcting ability in each column of the outer decoder is 16 bytes. If the RSPC decoder can use the errors detected in EFMPlus decoder, the inner decoder can also perform the erasure decoding. Therefore, we propose an RSPC with inner erasure decoder using the error information as shown in Fig. 3. This method can be easily applied to commercial DVD systems because it does not need to change the DVD spec [6].

Fig. 4 shows the flowchart of general error and erasure decoding procedure for the proposed scheme as follows.

(1) For each row, we search the number of erasures when the erasure information from EFMPlus decoder is received. If $\varepsilon > d_{\min}-1$ (=10), the decoding of the corresponding row is not performed and its erasure flag is transmitted to outer decoder. Otherwise, the syndrome values $\{s_0, s_1, \ldots, s_{2t-1}\}$ are computed.

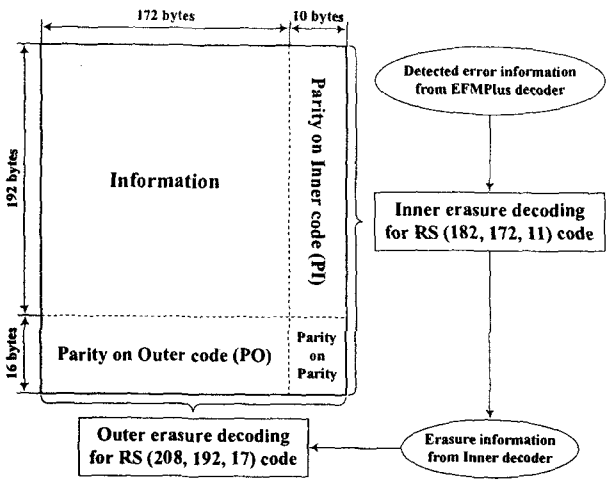


Fig. 3. RSPC structure with inner erasure decoder using error information detected from the EFMPlus decoder.

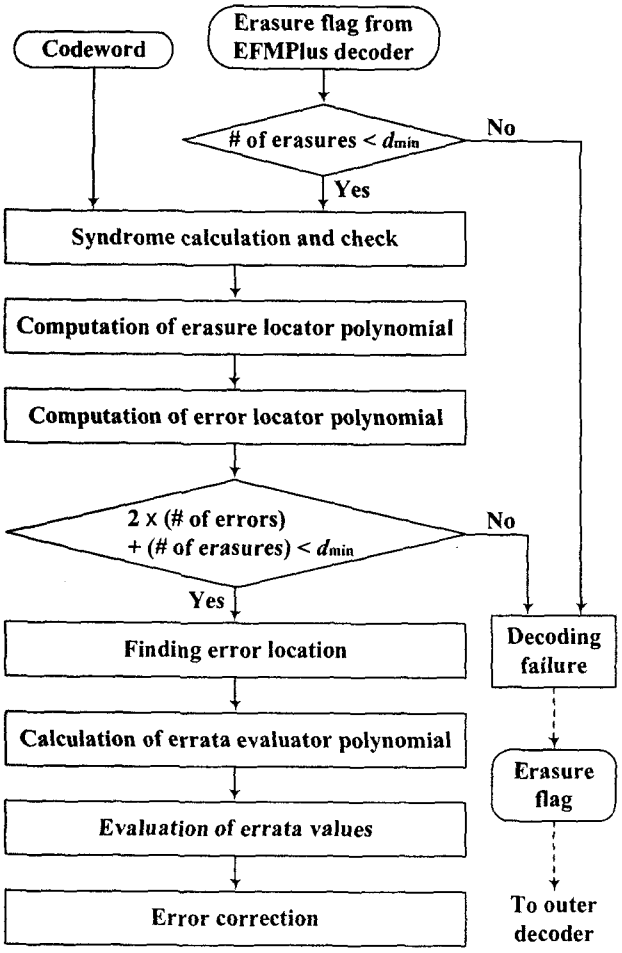


Fig. 4. Flowchart of error and erasure decoding.

(2) If all syndromes have zero values, it means that the received codeword from EFMPlus decoder has no error and no further decoding process is required. Otherwise, we form the polynomial $S(x)$ of all syndromes.

(3) For $S(x) \neq 0$, we compute the erasure locator polynomial using erasures.

(4) If $\varepsilon < d_{\min} - 2$ (=9), the error locator polynomial and the number of errors are computed.

(5) If $\varepsilon + 2\nu > d_{\min} - 1$ (=10), the decoding is failed and the erasure flag is transmitted to outer decoder. Otherwise, we calculate the error and erasure evaluator polynomial, which is called the errata evaluator polynomial, and errors are corrected by evaluating the errata values

(6) If the erasures from EFMPlus decoder exist, the erasures can be corrected up to 10 symbols in inner decoder.

3. Simulation Results

Basically, for a decoding block of original RSPC, the maximum correctable burst error length is 2922 bytes (=182 bytes × 16 rows + 5 bytes × 2 rows) because long burst errors generated in 16 rows can be corrected using the outer erasure decoding. When the erasure decoding is applied in the inner decoder, we can correct up to 10 byte symbols instead of 5 symbols for above 2 rows and therefore the maximum correctable burst error length is 2932 bytes. Table III compares the maximum correctable errors and burst error lengths for original and the proposed RSPC decoders. For the errors generated in a block, the maximum error correcting capability increases approximately 25% more than that of the original RSPC decoder.

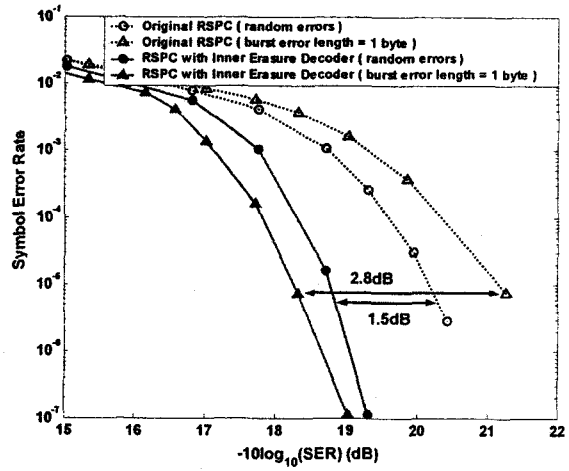
After inserting several error patterns, we compared the symbol error rates of the original RSPC and the proposed method as shown in Fig. 5. The horizontal and vertical axes represent symbol error rates of EFMPlus decoded output and RSPC decoded output, respectively. The symbol error rates are calculated by computer simulation based on RSPC blocks of 3×10^3 ($\approx10^8$ bytes).

In case of random errors, the performance gain of the proposed scheme compared to original RSPC is approximately 1.5 dB at 10^{-5} SER. However, if burst errors are occurred, the proposed decoder is more effective. For example, the proposed decoder achieves the gain of approximately twice when 1, 2 and 3 byte burst errors are inserted. Furthermore, when 5 and 10 byte burst errors are inserted, the performance gains are approximately 5.8 dB and 10.5 dB, respectively.

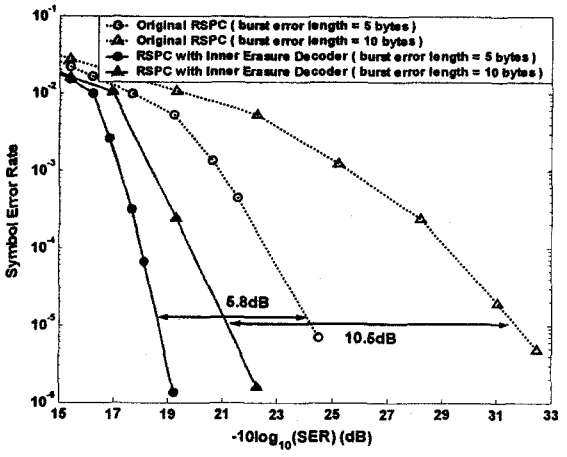
Therefore, we can identify that the error correcting ability of the proposed RSPC decoder is more powerful as the burst error length is increased.

TABLE III
CORRECTABLE ERROR LENGTH AND ERROR CORRECTION CAPABILITY

	Original RSPC decoder	Proposed RSPC decoder
Maximum correctable burst error length	182 bytes × 16 rows + 5 bytes × 2 rows = 2922 bytes	182 bytes × 16 rows + 10 bytes × 2 rows = 2932 bytes
Number of maximum correctable errors	182 bytes × 16 rows + 5 bytes × 192 rows = 3872 bytes	182 bytes × 16 rows + 10 bytes × 192 rows = 4832 bytes



(a) Random errors and burst errors with 1 byte length



(b) Long burst errors with 5 and 10 byte lengths

Fig. 5. Performance comparisons for several error patterns. The more the burst error length is increased, the better the decoder performance.

4. Design of RSPC with Inner Erasure Decoder

Fig. 6 shows the block diagram of the error correcting code and modulation/demodulation blocks in DVD systems. The EFMPlus decoder outputs are fed into the RSPC block to correct the errors. The RSPC is composed of an RS (182, 172, 11) code in row direction and an RS (208, 192, 17) code in column direction. In this section, we explain an efficient RS decoding architecture with a low hardware complexity.

As shown in Fig. 7, the proposed RS architecture consists of three balanced pipeline stages in terms of the number of clocks at each stage. Thus the throughputs of the proposed RSPC decoder become maximized. We divide the decoding process into three steps: (1) calculating the syndrome and erasure location polynomials from the received codewords and erasure location indicators, (2) generating the errata location polynomial and the errata evaluation polynomial, and (3) finally finding the errata locations by Chien search and evaluating the errata values by Forney's algorithm. In the figure, the number with an asterisk stands for the maximum number of clock cycles to complete the corresponding stage.

In the first stage, the syndrome polynomial $S(x)$ is computed with the received codewords. Then, the computation for the erasure location polynomial $\Gamma(x)$ starts right after the syndrome calculation has been completed. We schedule these in a series to share the hardware without any performance degradation since their architectures look very similar. It takes at most 224 clock cycles to complete the first step. The second stage will solve the key equation to generate the errata location polynomial $\Psi(x)$ and the errata evaluation polynomial $\Omega(x)$ based on the inverse-free Berlekamp-Massey (BM) algorithm [7]. We transform the BM algorithm into a symbol-serial structure to balance the pipelines, thus minimizing the hardware costs. $\Psi(x)$ and $\Omega(x)$ can be computed from $S(x)$ and $\Gamma(x)$ generated in the previous stage by 198 clock cycles at worst. Finally, the errata locations and errata values can be found by the Chien search and Forney evaluation procedures in stage three.

If no error occurs, the syndromes are all zero. The syndromes can be computed by Horner's rule in a nested form. At first, the syndrome is calculated from the received codewords and then the erasure polynomial can be computed on the same hardware. The bit-wide erasure location indicators are transformed to the roots for the erasure location polynomial $\Gamma(x)$ and buffered until the point at which they are used. $\Gamma(x)$ can be obtained by multiplying a linear shift of $\Gamma(x)$ with β^j and adding the result to itself recursively, where β's are the roots of the

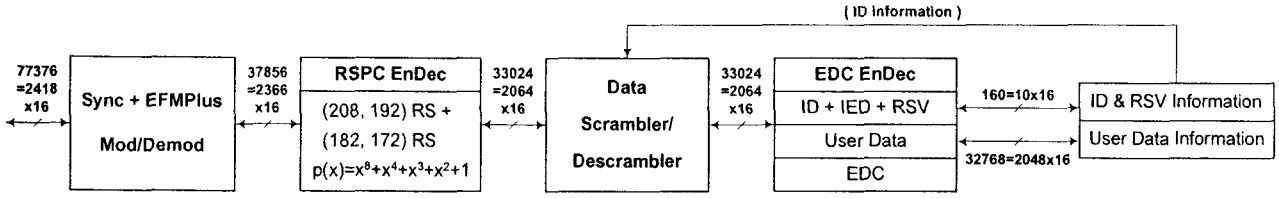


Fig. 6. Modulation/Demodulation and error correcting blocks in DVD systems.

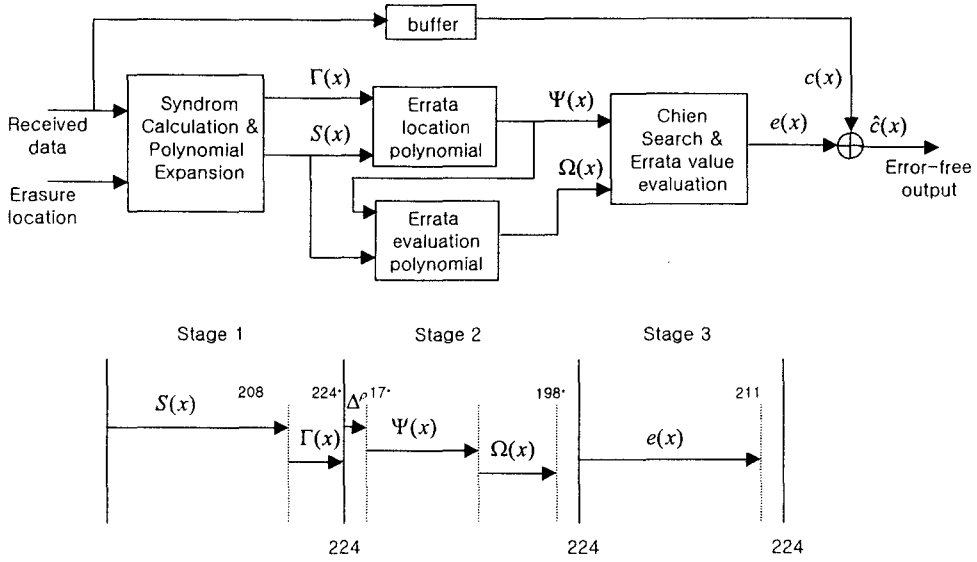


Fig. 7. The proposed three-stage pipeline architecture and the timing for the RS (208, 192, 17) decoder.

erasure location polynomial. In case that $\nu = 0$ and $\rho = 16$, this step takes 224 (208 + 16) clock cycles at worst.

There are two main methods used to solve the key equation: the Berlekamp-Massey (BM) algorithm and the Euclidean algorithm. The BM algorithm is generally known to require a smaller hardware complexity, whereas the Euclidean algorithm has efficiency in terms of regularity [8]. Many researchers have proposed the architectures for the BM algorithm [8][9][11][12]. Liu's architecture needs an FFI (Finite-Field Inverter) for a division operation, which requires larger area and processing time [12]. The inverse-free BM algorithms and architectures [9][10][11] have been proposed but their architectures run in parallel that requires 3ν FFMs. If the erasure correction capability is added, the required resources will increase. Performance-wise, they solve the key equation too fast with too much hardware; nevertheless, it does not increase the decoding performance at all since the performance is limited by the other pipeline stages. The decomposed BM architecture [8] has been proposed to slow down the whole process in a sequential manner with the least number of FFMs. However, the performance of this architecture becomes worse when the RS decoder has a larger $(n - k)$ value, such as (248, 216), since the stage for computing the BM algorithm becomes a bottleneck for decoding. Furthermore, when the erasure correction capability is added, the throughput becomes even worse since the degree of the errata location and evaluation polynomials become higher.

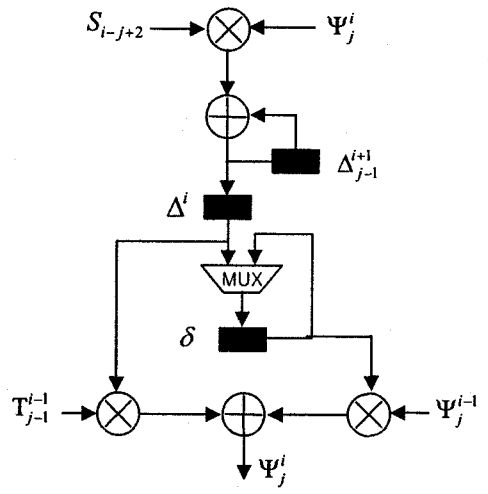
In this paper, we revise the inverse-free BM algorithm to balance the pipeline and maximize the decoding performance. The initial discrepancy Δ^{ρ} needs to be computed from $S(x)$ and $\Gamma(x)$ before solving the key equation. Fig. 8 shows the block diagram for calculating $\Psi(x)$ and $\Omega(x)$. It requires 3 FFMs and some random logics. We use the FFM structure proposed in [13] which is modular and regular. From the architecture point of view, the critical path delay for computing $\Psi(x)$ and $\Omega(x)$ is limited to the time of FFM + α. The result of logic synthesis shows that the FFM used in this design costs 380 gates and takes about 2.4ns using 0.35um technology.

After obtained the key equations, we can locate the roots of $\Psi(x)$ using Chien search method, which evaluates $\Psi(x)$ at all field elements in $GF(2^8)$ and determines whether the current position has an error or

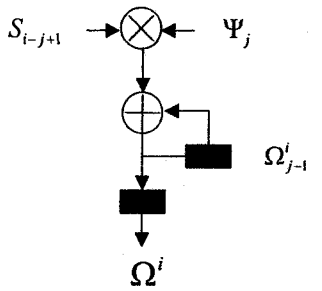
not. In other words, if $\Psi(\alpha^i) = 0$, there is an error at the ith symbol R_i in the received polynomial $R(x)$.

To evaluate the errata values, the Forney algorithm is preferred due to its lower circuit complexity. Fig. 9 shows the errata correction block using the Forney algorithm. The evaluation of $\Omega(\alpha^i)$ is computed in parallel with the Chien search, and therefore an additional few cycles are needed to produce the correct value e_i. The architecture to compute the Chien search and errata value evaluation seems quite straightforward and regular, but requires considerable gate counts mostly caused by FFMs. We can optimize the gate counts for each FFM using the fact that one of the inputs of FFM is a constant. The syndrome calculation block explained earlier has a similar structure, however it is difficult to optimize further because the FFMs in this block are shared with the polynomial expansion block for $\Gamma(x)$ that needs both inputs of FFM as variables. According to the Forney algorithm, a finite-field inversion is required to compute the errata values. We can implement this with either an LUT (Look-Up Table) ROM or an algorithm-based FFI circuit. The LUT-based approach seems to be more practical since the latency for a finite-field inversion circuit is hardly acceptable. Finally, the correcting term e_i is added to the delayed input data to produce the error-free codewords.

The proposed three-stage pipelined RSPC decoder is designed with Verilog HDL in a ModelSim environment and synthesized by Synopsis using a Hynix 0.35um standard cell library. The static timing analysis tells us that the control block for the BM algorithm causes the critical path to limit the decoding performance. The maximum throughput of the proposed decoder is about 740Mbps@100MHz, which is fast enough for DVD applications. The number of gate counts is 20.3K for RS (182, 172, 11) decoder and 30.7K for RS (208, 192, 17) decoder as shown in Table IV.



(a) Errata location polynomial Ψ(x)



(b) Errata evaluation polynomial Ω(x)

Fig. 8. Blocks for the errata location polynomial Ψ(x) and the errata evaluation polynomial Ω(x).

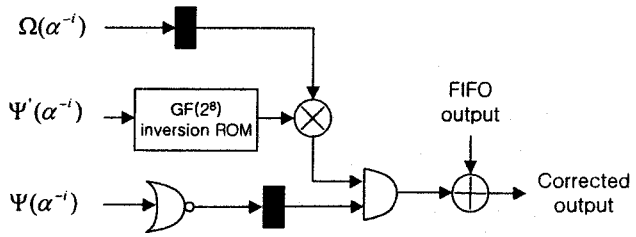


Fig. 9. Forney's algorithm and errata correction block.

TABLE IV. GATE COUNTS OF THE FUNCTIONAL BLOCKS IN THE PROPOSED RSPC DECODER.

RS decoder	(182, 172, 11)	(208, 192, 17)
Total gate counts	20.3K	30.7K

5. Conclusion

For DVD systems, as the reading and writing speeds are increased or data read process may be disturbed by some defects of disc, a number of errors can be generated. However, in commercial DVD systems, the correction of these errors is done by RSPC decoder only. Therefore, in read process, the errors may not be corrected because the error correcting abilities in each row and column of the RSPC are limited.

To increase the error correcting ability of DVD systems, we have proposed an RSPC with the inner erasure decoder which uses the decoding information of the EFMPlus code. We can identify that the EFMPlus decoder can detect errors more than 1/3 for random errors. Furthermore, the detected errors are more than 60% of total errors when burst errors are occurred. In results, for a block, the error correcting capability of the proposed scheme is improved up to 25% more than that of the original RSPC decoder. In addition, the more the burst error length is increased, the better the decoder performance.

The RSPC decoder is designed with a three-stage pipelined architecture, which is carefully balanced to maximize the decoding throughput. The BM algorithm

used to solve the key equation is transformed into a symbol-serial structure and its architecture requires less hardware while maintaining the decoding performance. The architecture operates the RS decoding, limited to the time of only one FFM + α. After synthesis, using the Hynix 0.35um standard cell library, the maximum throughput is 740Mbps@100MHz.

Above all, its apparent merit is that the proposed method can be applied to all of the commercial DVD systems without changing the DVD spec.

Acknowledgment

We are grateful to IC Design Education Center that provides us with a design environment.

References

[1] K. A. S. Immink, "EFMPlus: the coding format of the multimedia compact disc," *IEEE Trans. on Consumer Electronics*, vol. 41, no. 3, Aug. 1995.

[2] K. A. S. Immink and H. Ogawa, "Method encoding binary data," US Patent 4,501,000, Feb. 1985.

[3] K. A. S. Immink, "Method of converting a series of m-bit information words to a modulated signal, method of producing a record carrier, coding device, decoding device, recording device, reading device, signal, as well as record carrier," US Patent 5,696,505, Dec. 1997.

[4] W. Coene, H. Pozidis, M. van Dijk, J. Kahlman, R. van Woudenberg and B. Stek, "Channel coding and signal processing for optical recording systems beyond DVD," *IEEE Trans. on Magnetics*, vol. 37, no. 2, pp. 682-688, March 2001.

[5] B. Sklar, *Digital Communications 2/E: Fundamentals and Applications*, Prentice Hall, 2001.

[6] "DVD specifications for recordable disc for general part 1: physical specifications," Version 2.0, May 2000.

[7] T. Truong, J. Jeng and K. Hung, "Inversionless decoding of both errors and erasures of Reed-Solomon code," *IEEE Trans. on Communications*, vol. 46, pp. 973-976, 1998.

[8] H. Chang, C. Shung, and C. Lee, "A Reed-Solomon Product-Code (RS-PC) decoder chip for DVD applications," *IEEE J. of Solid-State Circuits*, vol.36, no.2, pp.229-238, 2001.

[9] I. Reed, M. Shieh, and T. Truong, "VLSI design of inverse-free Berlekamp-Massey algorithm," *IEE Proc.-E*, vol.138, no.5, pp.295-298, 1991.

[10] T. Truong, J. Jeng, and K. Hung, "Inversionless decoding of both errors and erasures of Reed-Solomon code," *IEEE Trans. on Communications*, vol.46, no.8, pp.973-976, 1998.

[11] D. Sarwate and N. Shanbhag, "High-speed architectures for Reed-Solomon decoders," *IEEE Trans. on VLSI*, vol.9, no.5, pp.641-655, 2001.

[12] K. Liu, "Architecture for VLSI design of Reed-Solomon decoders," *IEEE Trans. on Computers*, vol.C-33, pp.178-189, 1984.

[13] E. Mastrovito, "VLSI design for multiplication over finite fields $GF(2^m)$," *Proc. 6th Int'l Conf. Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes (AAECC-6)*, pp.297-309, 1988.