Error control codes used in DVD

Part Ⅰ：

The Digital Versatile Disc (DVD): System Requirements and Channel Coding

by 洪斌峰

PartⅡ：

Reed-Solomon Pruduct Code on DVD

by 林敬雄

1. **Introduction**

- The single-layer DVD disc can hold 4.7 Gbytes, which amounts to seven times more data than an audio CD. Applying multiplayer technology, a storage capacity of up to 17 Gbytes was obtain.

- The major part of the capacity increase is achieved by the use of optics, shorter laser wavelength, and larger numerical aperture, which reduces the spot diameter by a factor of 1.65.

- The storage capacity of the DVD is also increased by a complete redesign of the logical format of the disc, including a more powerful Reed-Solomon product code (RS-PC) and recording code (EFMPluse)

2.Physical Aspects

- In order to allowing full backward compatibility, mechanical specifications such as outer diameter and center hole diameter of the DVD are equal to those of the CD.

- However, in order to get a higher storage capacity than possible with a disc of 1.2mm thickness, the thickness was halved to 0.6mm. This mechanical instabilities is solved by back-to-back bonding of two 0.6mm discs.

- A second method to increase the capacity by a factor of two is by using a dual-layer disc, where the layers are on one side of the disc. The principle of operation of the dual-layer disc is shown in Fig.1.

- By bonding two dual discs, the structure depicted in Fig.2 is obtained. The sandwich dual-layer disc has a capacity of 18 Gbytes.

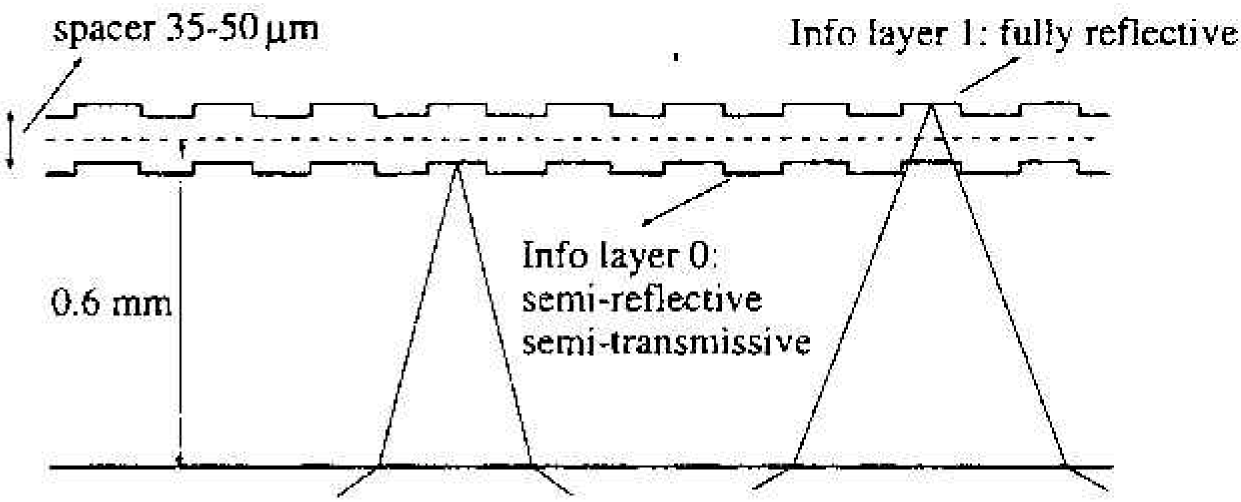


Fig 1. Dual-layer reading principle diagram

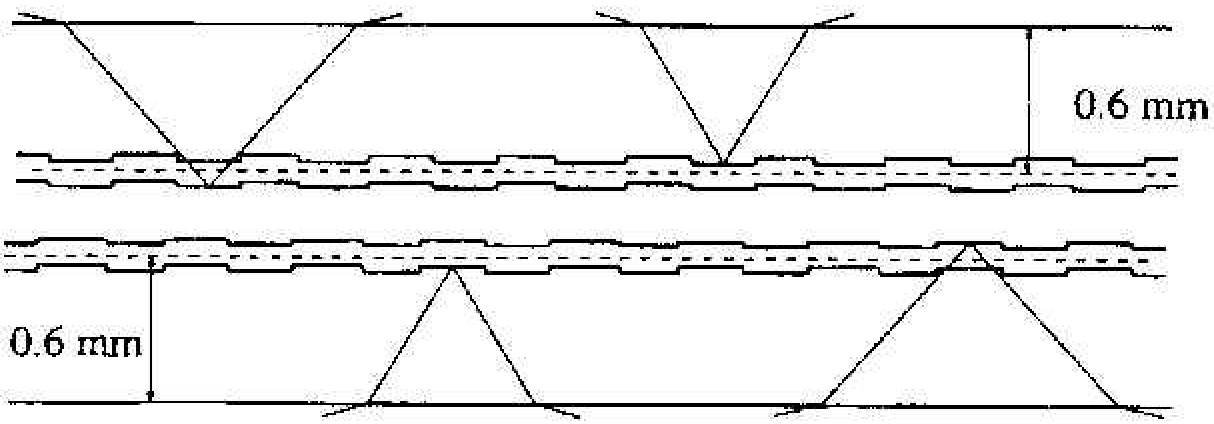


Fig 2. Double dual-layer reading principle diagram

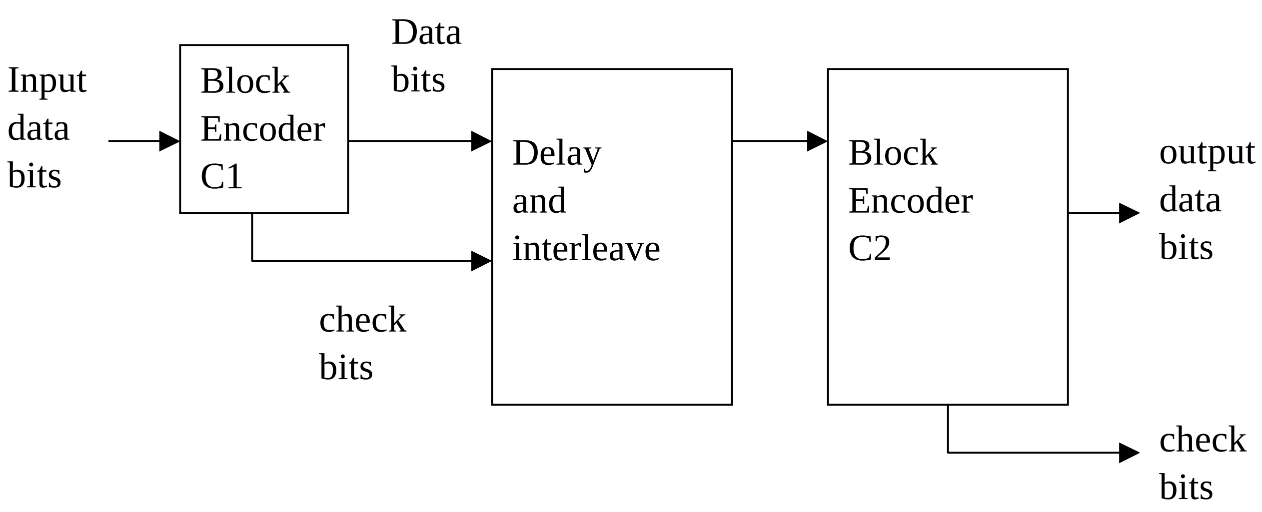
3.Electronic Aspects

- Under DVD format rules (RS-PC + EFMPlus), sectors of 2048 user bytes are translated into 2418 bytes (2048 user + 52 sync + 16 header + 302 error code), which are in turn translated into 16 x 2418 channel bits. Thus one user bit is translated into 4836/2048 = 2.36 channel bits.
- In a conventional CD (CIRC + EMF), an audio bit is translated into 588/192 = 3.05 channel bits. And we can conclude that the format efficiency of the DVD is improved by 32%.

4.Error Correction Code：RS-PC

- The errors found in both the CD and DVD systems are a combination of a random and burst character. For DVD a much more powerful error correction is required for several reasons. **First**, the increased physical density implies that physical imperfections affect proportionally more bits. **Second**, as DVD is a true multimedia disc, its data integrity must be comparable to that of computer data.
- CIRC use a combination of two RS codes denoted by C1 and C2. In CIRC, C1 is [32,28] code and C2 is [28,24] code. Thus is rate of CIRC is 24/32.

A cross-interleaved block encoder



In RS-PC, the C1 and C2 codes are significantly longer than in CIRC, namely [208,192] and [182,172]. As a result, the rate of RS-PC is much higher than that of CIRC, namely 172*192/(182*208) = 0.872.

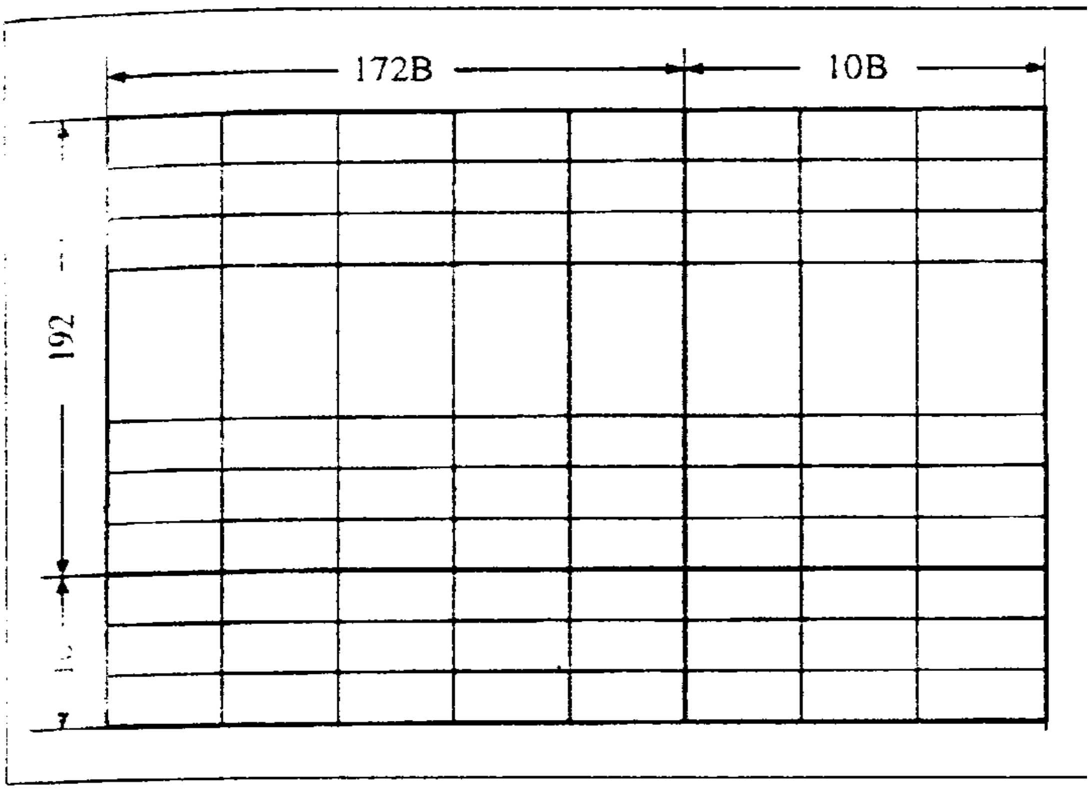


Figure 5. Block diagram of RS-PC encoder.

- The maximum correctable burst length is approximately 500 bytes for CIRC, while it is 2200 bytes for RS-PC. RS-PC is capable of reducing a random input error rate of $2x10^{-2}$ to a data error rate of 10^{-15} which is a factor of 10 better than in CD.

5.EFM Recording Code

- Under EFM rules, the data bits are translated eight at a time into fourteen channel bits, with minimum runlength parameter d=2 and a maximum runlength parameter k=10 (this means at least 2 and at most 10 successive zeros between successive ones). Part of the EFM coding table is presented in Table 1. And it should be noted that the codewords are described in NRZI notation.

Data	*Code*	*Data*	*Code*
100	01000100100010	112	10010010000010
101	00000000100010	113	00100000100010
102	01000000100010	114	01000010000010
103	00100100100010	115	00000010000010
104	01001001000010	116	00010001000010
105	10000001000010	117	00100001000010
106	10010001000010	118	01001000000010
107	10001001000010	119	00001001001000
108	01000001000010	120	10010000000010
109	00000001000010	121	10001000000010
110	00010001000010	122	01000000000010
111	00100001000010	123	00001000000010

Table 1: Part of EFM coding table.

- It is easily seen that at least two bits, called merging bits, are required to ensure that the runlength conditions continue to satisfied when the codewords are cascaded.
- A third merging bit is necessary to give sufficient freedom for effective suppression of low-frequency content. (There are two reasons why EFM suppression low frequency content. **First**, the servo systems for track following and focusing are controlled by low-frequency signal, so that low-frequency components of the information signal could interfere with the servo system. **Second**, fingerprints on the disc will cause low frequency disturbance.). Our measure of the low-frequency content is the running digital sum (RDS); this is the difference between the totals of pit and land lengths accumulated from the beginning of the disc.

● The following figure illustrate how to determine merging bits.

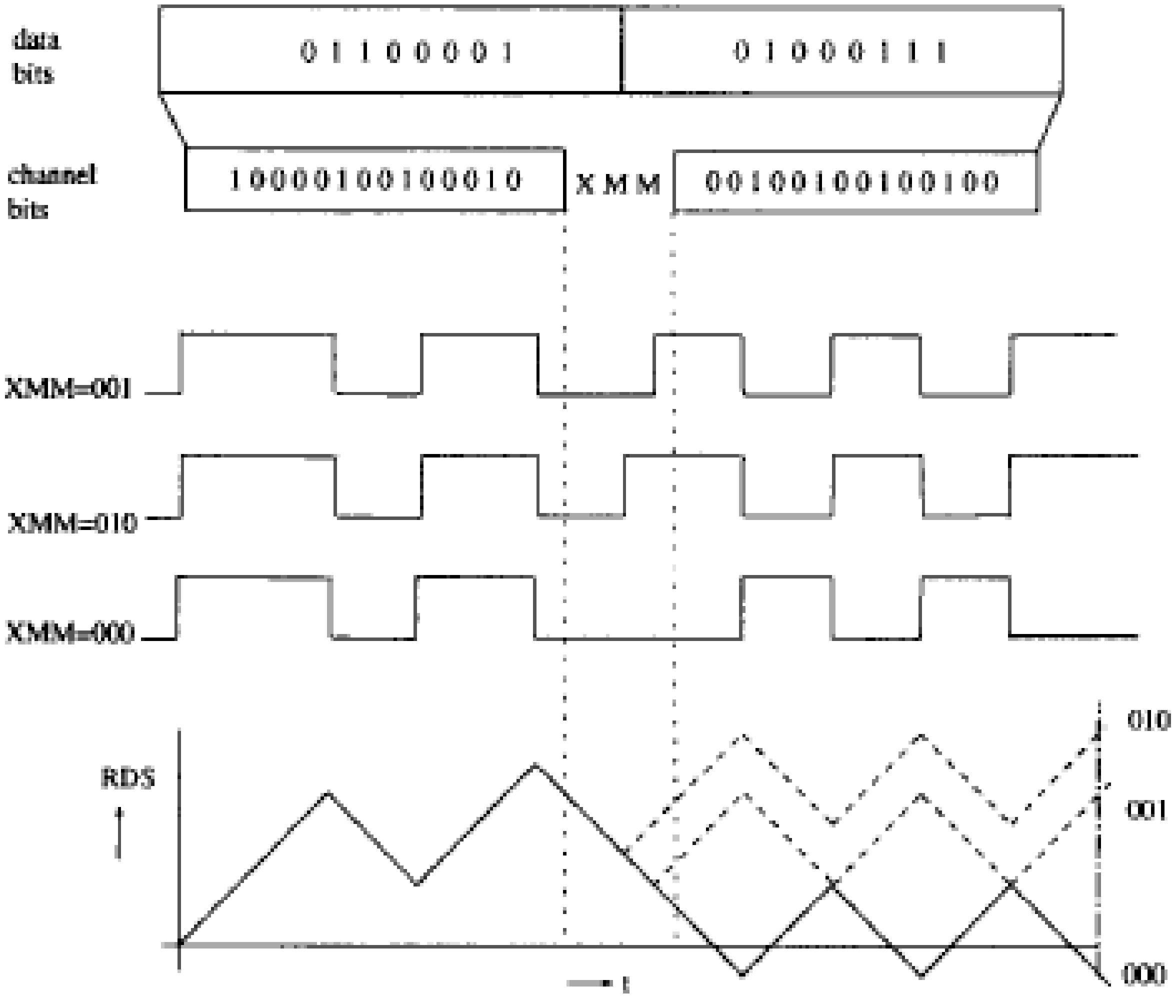
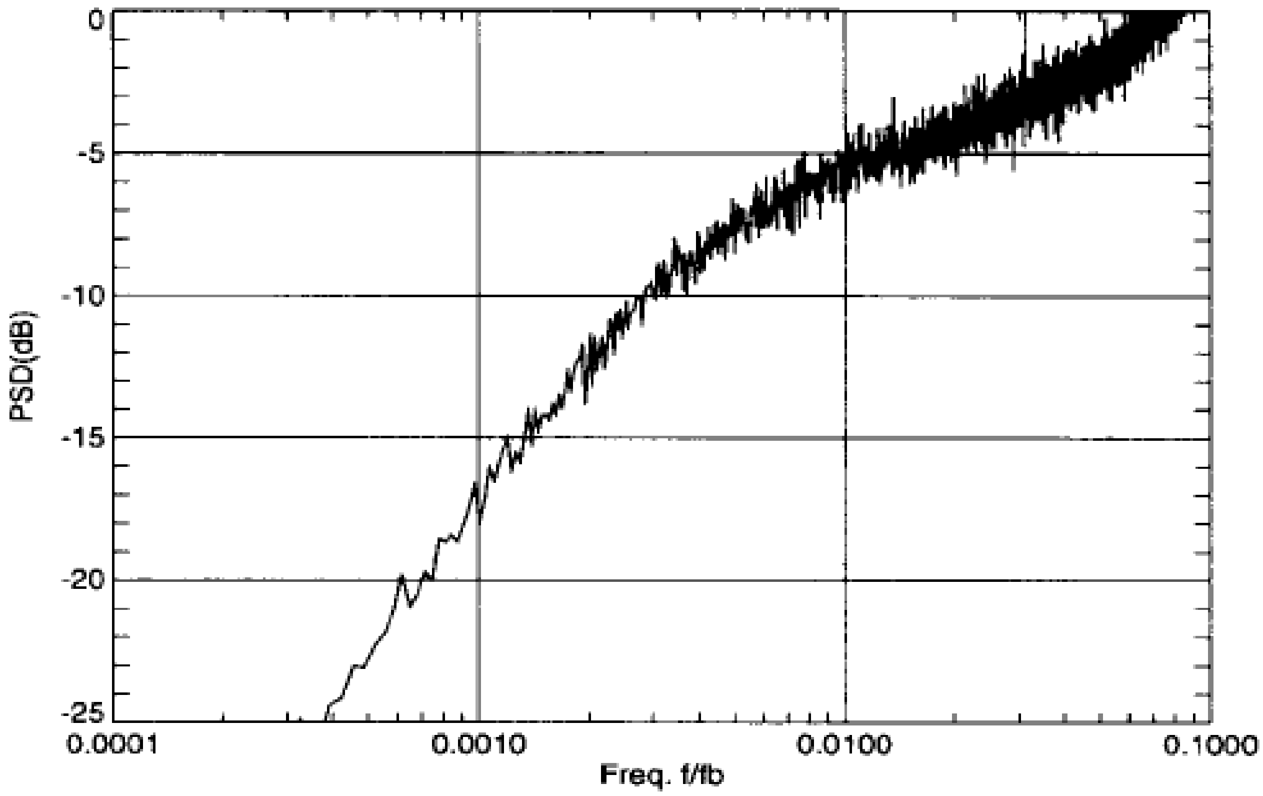


Figure 1: Strategy for minimizing the running digital sum (RDS). Eight user bits are translated into 14 channel bits. The 14 bits are merged by means of 3 merging bits in such a way that the runlength conditions continue to be satisfied. The proviso that there should be at least 2 'zeros' between 'ones' requires a 'zero' at the first merging bit position. In this case there are thus three alternatives for the merging bits: '000', '010', and '001'. The encoder chooses the alternative that gives the lowest absolute value of the RDS at the end of a new codeword, i.e. '000' in this case.

● Spectrum of conventional EFM



6.Description of the EFMPlus Encoder

- In EFM, 8 data bits are mapped to 17 channel bits (14 EFM bits + 3 merging bits). EFMPlus maps 8 data bits to 16 channel bits which increase 6%(1/17) recording efficiency. Under EFMPlus rules the runlength conditions are still satisfied after codewords cascaded. However the cost is a larger coding table and worse low-frequency property.

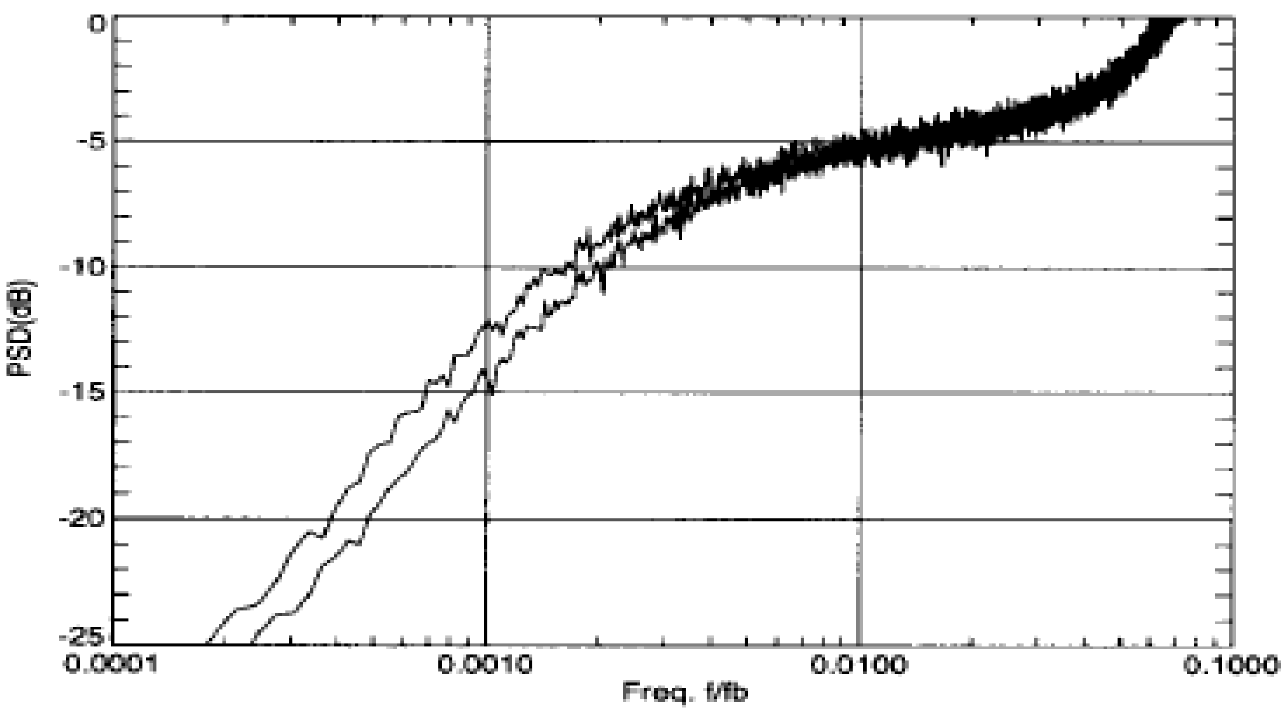
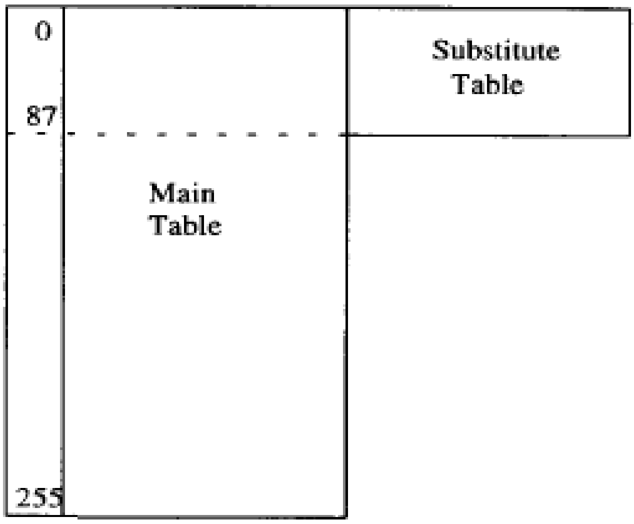


Figure 4: Spectrum of EFMPlus. The upper curve shows the spectral density without look-ahead while the lower curve shows the spectral density using two bytes look-ahead.



Block diagram of EFMPlus encoder

- In EFM, DC Control is done by adding merging bit. In EFMPlus, there are 88 out of 256 source codewords which each source codeword (8 bits) maps to two channel codewords (16 bits). Thus, the DC Control can be done by choosing the codeword which minimize the RDS from two available selections.

- The principle of operation of the encoder can be represented by a finite-state sequential machine with an 8-bit input, a 16-bit output and four states which are functions of the time. A word in our context is a 16-bit sequence that obeys the prescribed constrains (d=2,k=10). The states and word sets are characterized as follows.

1. Words entering State 1 end with {0,1} trailing zero
2. Words entering State 2 and 3 end with {2,…,5} trailing zeros
3. Words entering State 4 end with {6,…,9} trailing zeros

- The words leaving the states are chosen in such a way that the concatenation of words entering a state and those leaving that state obey the channel constraints (d=2, k=10). In order to distinguish the word leaving State 2 and 3, the word leaving State 2 have been selected such that the first (msb) bit, x1, and the thirteenth bit, x13, are both equal zero.

Words leaving State 3 are characterized by the fact that the 2-tuple x1x13 does not equal ‘00’.Let the entry t_{ij} of the 4 x 4 transition matrix T denote the number of words leaving state I that go to state j. Then we obtain

$$T = \begin{bmatrix} 138 & 96 & 96 & 22 \\ 145 & 90 & 90 & 27 \\ 132 & 102 & 102 & 15 \\ 164 & 113 & 113 & 25 \end{bmatrix}$$

The fan-outs of States 1,2,3,4 are 352, 352, 351, 415. Thus from each states at least 351 words are leaving.

A. Encoder operation

The codeword transmit by the encoder at time instant t, x_t, and the next state, S_{t+1}, are function of source word b_t and current state S_t.

$$X_t=h(b_t,s_t)$$

$$S_{t+1}=g(b_t,S_t)$$

For example：

i	$h(i,1), g(i,1)$	$h(i,2), g(i,2)$	$h(i,3), g(i,3)$	$h(i,4), g(i,4)$
0	0010000000001001, 1	0100000100100000, 2	0010000000001001, 1	0100000100100000, 2
1	0010000000010010, 1	0010000000010010, 1	1000000100100000, 3	1000000100100000, 3
2	0010000100100000, 2	0010000100100000, 2	1000000000010010, 1	1000000000010010, 1
3	0010000001001000, 2	0100010010000000, 4	0010000001001000, 2	0100010010000000, 4
4	0010000010010000, 2	0010000010010000, 2	1000000100100000, 2	1000000100100000, 2
5	0010000000100100, 2	0010000000100100, 2	1001001000000000, 4	1001001000000000, 4
6	0010000000100100, 3	0010000000100100, 3	1000100100000000, 4	1000100100000000, 4
7	0010000001001000, 3	0100000000010010, 1	0010000001001000, 3	0100000000010010, 1
8	0010000010010000, 3	0010000010010000, 3	1000010010000000, 4	1000010010000000, 4

B. Decoder operation

Basically, knowledge of the encoder state at the receiver site is necessary to re-constitute the source word. $b_t=h^{-1}(x_t,S_t)$

On a few occasions, as we can observe in the table, two source words have the same channel representation. For example, source words ‘3’ and ‘7’ generate the same channel representation when the encoder is in State 1. This ambiguity can be remedied by observing that the code was constructed in such a way that if the same two words do leave a given state, one of them goes to state 2 and the other to State 3. In other words,

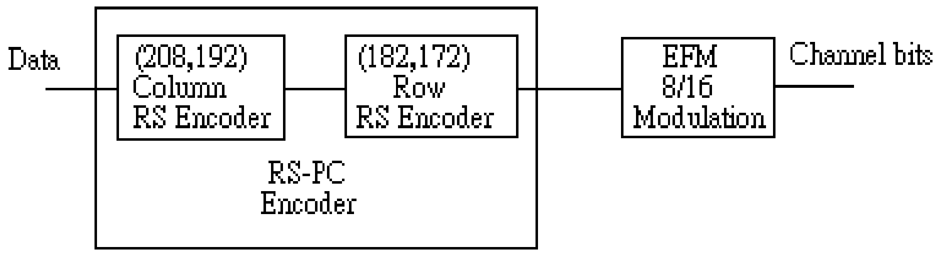
$$b_t=h^{-1}(x_t, x_{t+1,1}, x_{t+1,13})$$

7.Conclusion

- The DVD is a new optical recording medium with a storage capacity seven times higher than the conventional CD. A significant part of the capacity increase has been achieved by the use of optics, shorter laser wavelength, and larger numerical aperture, which reduces the spot diameter by a factor of 1.65.
- In DVD, the conventional CIRC and EFM codes have been replaced by a more powerful error correction, RS-PC, and recording code, EFMPlus. As a result, the format efficiency of the DVD relative to audio CD is improved by 32%. Even though consuming a significantly lower data overhead, the error correction code can cope with longer bursts and more random errors.

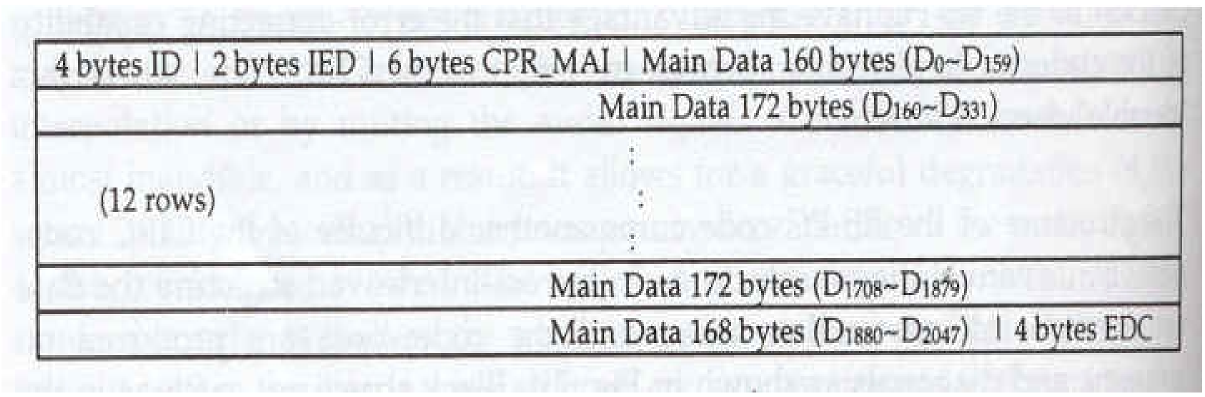
Part II Error Correction in DVD

Encoder Flowchart

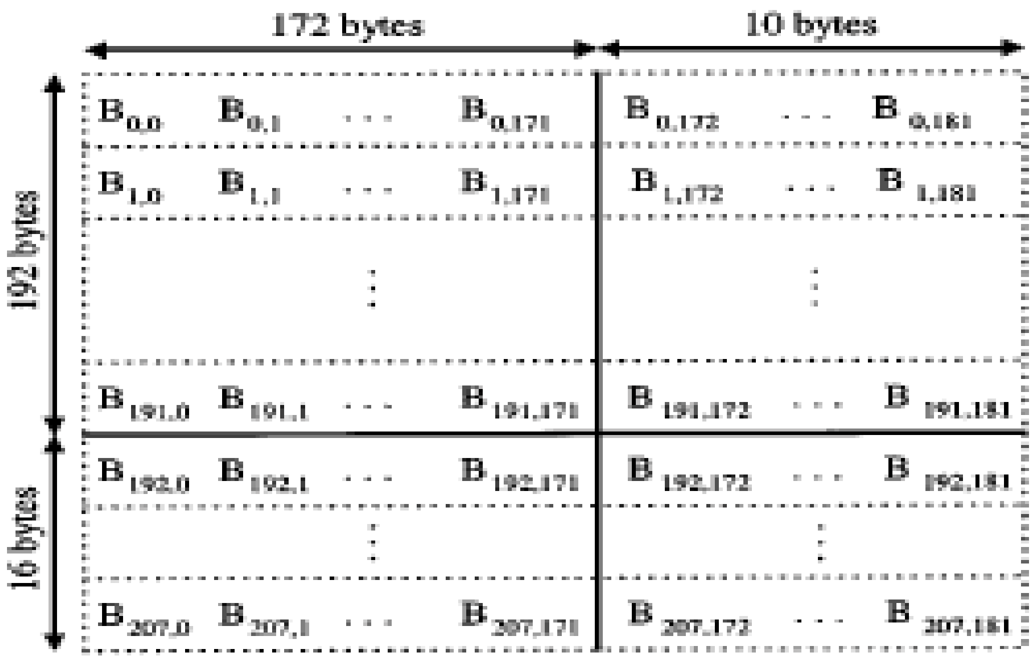


Data Sector Configuration

◆ The 2064 byte sector is, for purposes of error correction, organized into 12 separate rows, each with 172 bytes. The first row starts with the 12 Byte sector header (ID, IEC, Reserved bytes), followed by the remaining data bytes. The following 10 rows contain only data. The final row is punctuated with a 4 Byte field (EDC).



- To combat bursty errors characteristic of CD-ROM, 16 sectors are further interleaved together, forming a block of 192 rows (16 sectors * 12 rows/sector = 192 rows). Error correction bytes are concatenated to the data block in a 2-dimensional fashion (hence the term "product" in the phrase "Reed-Soloman product codes").
- Specifically: at the end of each row, 10 bytes of RS data are added, hence the RS(182,172,11) vector. At the end of the block, 16 rows of RS data are added (hence the RS(208,192,17) vector). Therefore out of 37,856 total bytes (182*208) for the interleaved block of data, only 33,024 bytes (192*172) or roughly 87% is payload.



RS-PC Encoder

- To generate the PO and PI, the following encoding procedure is performed :

  First, in column j (for j=0 to 171) attach 16 bytes(for i=192 to 207) of outer-code RS(208,192)

$$
\begin{aligned}
\mathrm{r}_{\mathrm{j}}(x) &= \sum_{i=192}^{207} b_{i,j}\, x^{207-i} \\
&= q_j(x) \bullet x^{16} \quad \bmod \quad g_{PO}(x),
\end{aligned}
$$

where

$$
q_j(x) = \sum_{i=0}^{191} b_{i,j}\, x^{191-i}, \quad g_{PO}(x) = \prod_{k=0}^{15} (x + \alpha^k)
$$

- Next, in row i(i=0 to 207), attach 10 bytes (j=172 to 181) of inner-code RS(182,172)

$$
r_i(x) = \sum_{j=172}^{181} b_{i,j}\, x^{181-j} = q_i(x) \bullet x^{10} \quad \bmod \quad g_{PI}(x)
$$

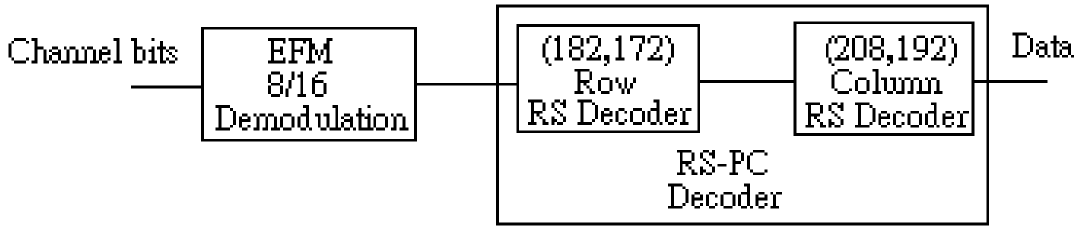
where

$$
q_i(x) = \sum_{j=0}^{171} b_{i,j}\, x^{171-j}, \quad g_{PI}(x) = \prod_{k=0}^{9} (x + \alpha^k),
$$

Here α represents the primitive root of the primitive polynomial,

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1$$

Decoder Flowchart



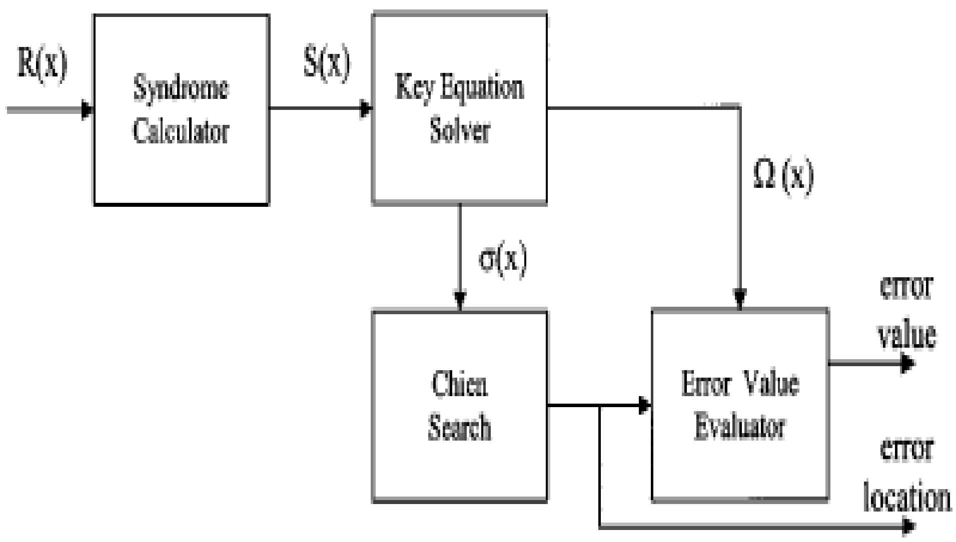
There are six major decoder stage on DVD:

- ✧ Stage 1→SYNC/ID detection, 8/16 Demodulation,
- ✧ ***Stage 2→Error detection and correction*** : If the check bits (EDC) don't match the finger- print of the unscrambled data, the Reed Soloman bytes (IEC) are used to attempt error correction of the corrupted data.
- ✧ Stage 3→Descramble/Decrypt
- ✧ ***Stage 4→EDC Check*** : The fingerprint of the unscrambled data is checked against the EDC code to verify whether the data was correctly descrambled.
- ✧ Stage 5→Track buffer
- ✧ Stage 6→Transfer to MPEG system decoder.

RS-PC Decoder

◆ The most popular RS decoder today can be summarized into four steps:

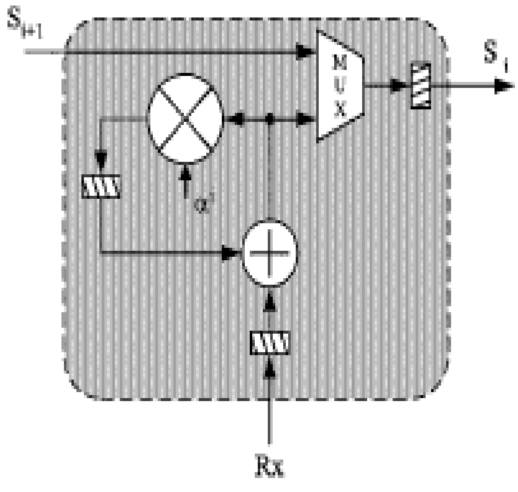
1) calculating the syndrome polynomial S(x)
2) Computing the error locator polynomial σ (x)
   And the error evaluator polynomial S(x) σ (x)= Ω (x) mod x^{2t}
3) Finding the error locations using Chien Search, it's equivalent to find out β_i, such that σ (β_i)=0.
4) Computing error values using Forney's algorithm and correcting error.



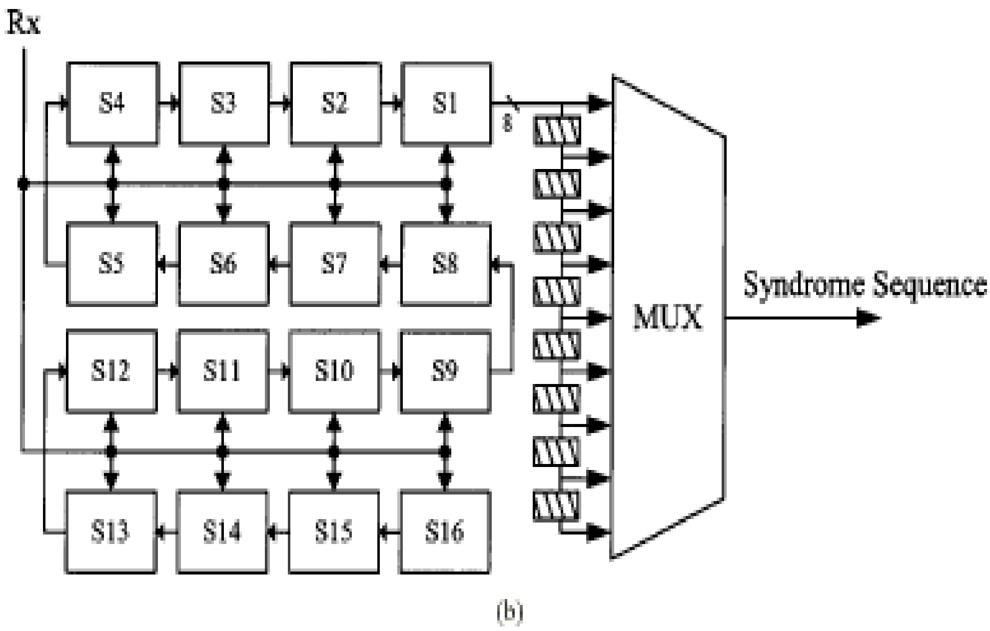
A. Syndrome Calculator

$$S(x)=\sum_{i=1}^{2t} S_i x^{i-1}, \quad S_i = R(\alpha^i),$$
$$where \quad R(x)=R_0 + R_1 x + R_2 x^2 + \ldots + R_{N-1} x^{N-1}$$

◆ As shown in Fig. , at each cycle, the partial sybdrome is multiplied with α^i and acculmulated with the received symbol.



(a)



(b)

B. *Key Equation Solver*

- The block to solve the Key equation occupies the largest area of RS decoder.
- The techniques frequently used to solve the key equation include the Berlekamp-Massey algorithm, the Euclidean algorithm, and the continue-fraction algorithm.
- Berlekamp-Massey algorithm is generally considered to be the one with the least hardware complexity, but still require an FFI(finte field inverter) to implement the division operation.
- Inversionless Berlekamp-Massey algorithm is proposed by Burton at 1971 and is widely be used in VLSI implementation.

◆ Berlekamp-Massey algorithm

$$initial\ \ condition\ :\quad C^{(0)}(D)=1\quad B^{(0)}(D)=1\quad L^{(0)}=0$$

$recursively,$

$$\Delta^{(k+1)}=\sum_{j=0}^{L^{(k)}} C_j^{(k)} S_{m_0+k-j}$$

and let

$$C^{(k+1)}(D)=C^{(k)}(D)-\Delta^{(k+1)}B^{(k)}(D)D$$

$$B^{(k+1)}(D)=\begin{cases} DB^{(k)}(D) & if\ \ \Delta^{(k+1)}=0\quad or\quad if\ \ 2L^{(k)}>k \\ C^{(k)}(D)/\Delta^{(k+1)} & if\ \ \Delta^{(k+1)}\neq 0\quad and\quad 2L^{(k)}\leq k \end{cases}$$

◆ Inverse-free Berlekamp-Massey algorithm

$$Initial\ :\quad \mu^{(0)}(D)=1\quad \lambda^{(0)}(D)=1\quad l^{(0)}=0$$

and

$$\gamma^{(k)}=1\quad if\ \ k\leq 0$$

Re *cursively* ,

$$\delta^{(k+1)}=\sum_{j=0}^{l^{(k)}} \mu_j^{(k)} S_{m_0+k-j}$$

$$\mu^{(k+1)}(D)=\gamma^{(k)}\mu^{(k)}(D)-\delta^{(k+1)}\lambda^{(k)}(D)D$$

$$\lambda^{(k+1)}(D)=\begin{cases} D\lambda^{(k)}(D) & if\ \ \delta^{(k+1)}=0\quad or\quad if\ \ 2l^{(k)}>k \\ \mu^{(k)}(D) & if\ \ \delta^{(k+1)}\neq 0\quad and\quad 2l^{(k)}\leq k \end{cases}$$

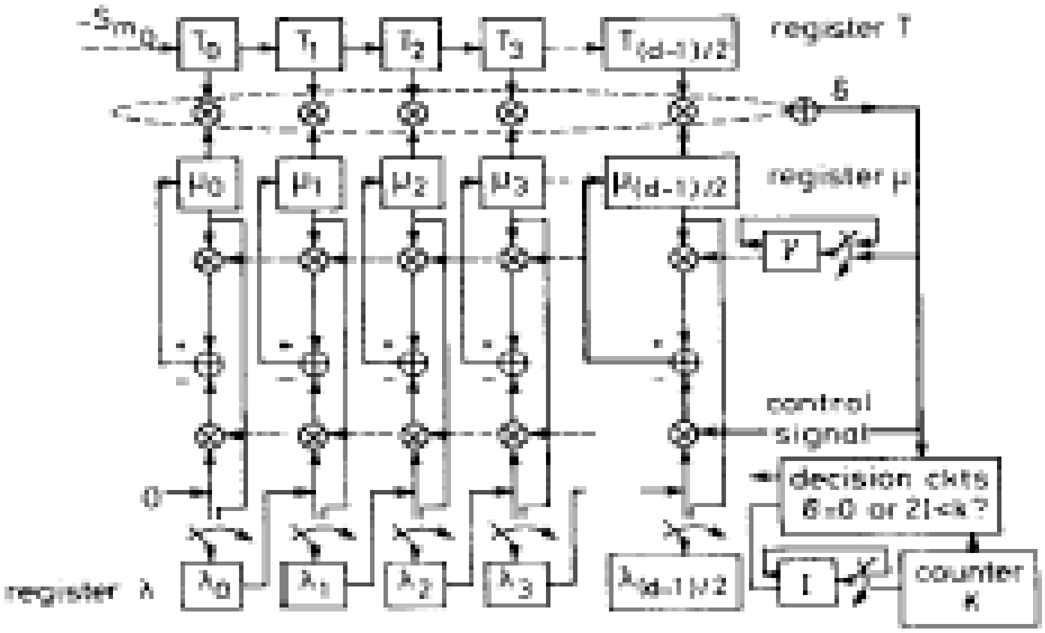
$$and\quad \gamma^{(k+1)}=\begin{cases} \gamma^{(k)} & if\ \ \delta^{(k+1)}=0\quad or\quad if\ \ 2l^{(k)}>k \\ \delta^{(k+1)} & if\ \ \delta^{(k+1)}\neq 0\quad and\quad 2l^{(k)}\leq k \end{cases}$$

◆ Principle of these two algorithm

$$\mu^{(k)}(D) = \prod_{i=-1}^{k-1} \gamma^{(i)} C^{(k)}(D)$$
$$\lambda^{(k)}(D) = \gamma^{(k)} B^{(k)}(D)$$
$$l^{(k)} = L^{(k)}$$

◆ The degrees of μ(D) and C(D) are the same at each stage but the C(D) always has its lowest order coefficient equal to 1 whereas the lowest order coefficient of μ(D) can be any nonzero field element.

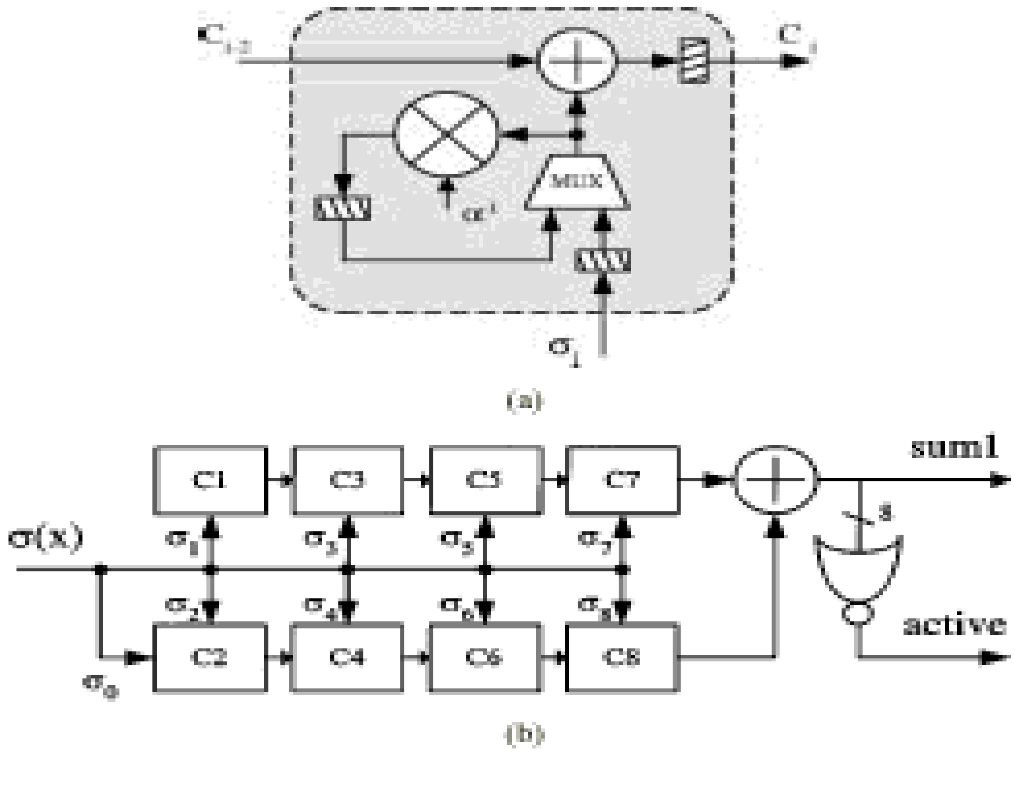
◆ One more storage register is required to store the coeeficient of μ(x), and another register needed to store γ(x).



C. *Chien Search*

- Chien Search is uesd to check whether the error locator σ (x) equals zero or not while $x = \alpha^{-n}$.
- If $\sigma(\alpha^{-n}) = 0$, it means there is an error at R_n, where the received polynomial is defined as

$$R(x) = R_0 + R_1 x + R_2 x^2 + \ldots + R_{N-1} x^{N-1}$$



(a)

(b)

D. Error Value Evaluator

◆ In the Forney algorithm, the error value becomes

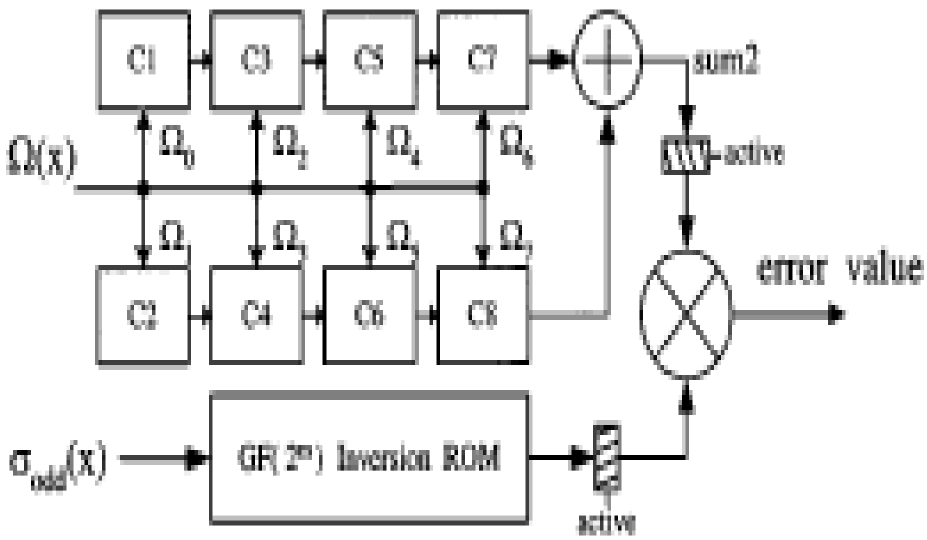
$$e_l = \frac{\Omega(\beta_l^{-1})}{\sigma'(\beta_l^{-1})}$$

◆ In the $GF(2^m)$,

$$\begin{aligned}\sigma'(x) &= \left(\sum_{k=0}^{t} \sigma_k x^k\right)' \\ &= \sum_{k=1}^{t} k \cdot \sigma_k x^{k-1} \\ &= \sigma_1 + \sigma_3 x^2 + \ldots\ldots + \sigma_{t_{odd}} x^{t_{odd}-1} \\ &\equiv \frac{1}{x}\sigma_{odd}(x)\end{aligned}$$

So we rewrite the Forney's algorithm as

$$e_l = \frac{\Omega(\beta_l^{-1}) \cdot \beta_l^{-1}}{\sigma_{odd}(\beta_l^{-1})}$$



Performance of RS-PC

- These two RS inner- and outer- codes form a direct-product code.
- The outer-code RS(208,192) has a minimum distance 17 and is able to correct 8 bytes of errors.
- The inner code RS(182,172) has a minimum distance 11 and is able to correct up to 5 bytes of errors
- The direct-product code is a two-dimential (208x182,192x172) code whose minimum distance is 17x11=187 bytes and is able to correct (17x11-1)/2=93 byte random error
- The burst-error-correcting capabilities of the RS-PC code satisfies:

  $b \geqq \max\{208 \times 5, 182 \times 8\} = 1465$ bytes=11720 bits

Appendix

PRML/Viterbi Decoder

- Characteristics of the optical channel are affected by the characteristics of an optical pick-up devise
- For bit-by-bit detection, the signal level (+,-) must be made out to revert to the original data stream.
- On DVD, by the readout crosstalk from the adjacent tracks, the read out noise signal component increases.
- PRML(Partial Response Maxium Likelihood): Partial Response means essentially alowing more than one clock period for the signal changing from its lowest level to the highest level.
- The viterbi decoder has an ability of error correction. Due to the EFM code having a run length limited characteristic, several states and branches can be deleted.

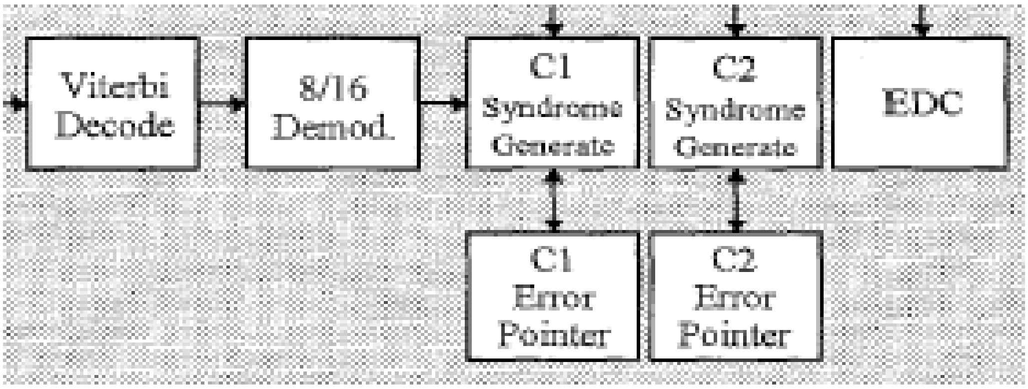


Figure 1

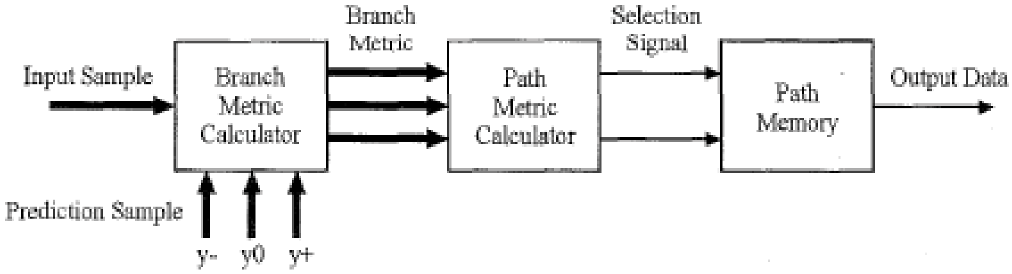
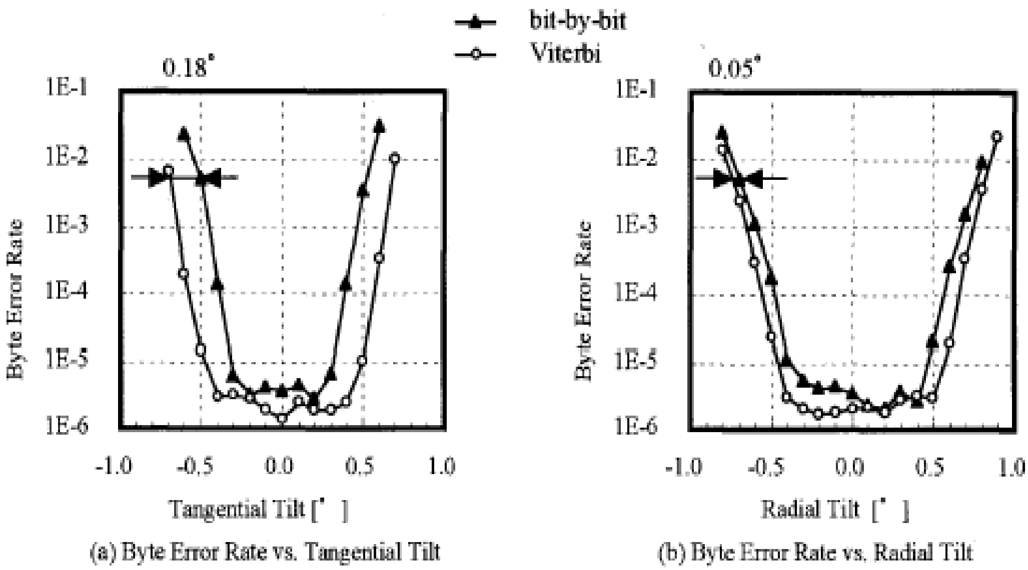


Figure 2



◆ The Viterbi decoder always showed better error rates than the bit-by-bit decoder did.

Reference

[1] K. A. S. Immink, “The digital versatile disc (DVD) : system requirements and channel coding”, SMPTE Journal, pp.483-489, August 1996.

[2] K. A. S. Immink, “ EFMPlus：The Coding Format of the Multimedia Compact Disc,” IEEE Trans. Consumer Electr., CE-41:491-497, Aug. 1995.

[3] Irving S.Reed and Xuemin Chen, Error-Control Coding for Data Networks, Kluwer Academic Publishers, 1999.

[4] H. Chang, C. Bernard Shung, C. Lee, “A Reed-Solomon Product-Code (RS-PC) Decoder Chip for DVD Applications,” IEEE Solid-State Circuit, vol. 36, No. 2, pp. 229-238, Feb 2001.

[5] I. Reed, M. Shih, and T. Trung, “VLSI design of inverse-free Berlekamp-Massey algorithm,” Pro. IEE, pt. E, vol. 138, pp. 295-298, Sept 1991.

[6] S. Choi, J. Kong, B. Chung, Y. Kim, “Viterbi Detector Architechture for High Speed Optical Storage,” IEEE TECON, 1997.

[7] H. Hayashi, H. Kobayashi, M. Umezawa, S.Hosaka, H. Hirano, “DVD Player using a Viterbi Decoding Circuit,” IEEE Transs Cosumer Electronics, Vol. 44, No. 2, May 1998.