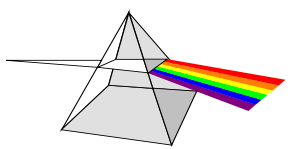
Error Correction

- Sources of Errors
- Cyclic Redundancy Check Code
- Error-Correction Codes
- Interleaving
- Reed-Solomen Codes
- Cross-Interleave Reed-Solomon Code



Introduction

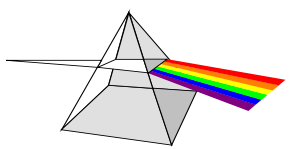
Potential for the Error Correction Techniques

- Digital audio can be coded for error-correction and concealment.
- Relax the manufacturing tolerances for mass media as the compact disk.

Roles of the Error Correction Techniques

- A key technique in the evolving from analog audio to digital audio.
- An obligation for high data densities in audio storage media.
- Can approach the computer industry standard which specifying an error rate of 10^{-12}.

The Evolution of Digital Audio Technique can be Measured by the Prequisite Adnances in Error Correction



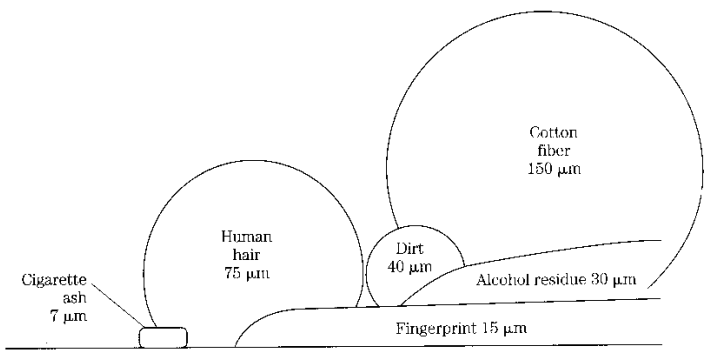
1. Sources of Errors

Sources

- Magnetic tap
  - – Dust, scratches, fingerprints, tap stretching or abuse, impure oxide or blinder, irregular tape slitting, and physical editing.
- Optical Media
  - – Pit asymmetry, bubbles or defects in subtrate, and coating defects.
- Transmitted Data
  - – multipath interference, atmospheric conditions, and other interfering signals.

Error Classification

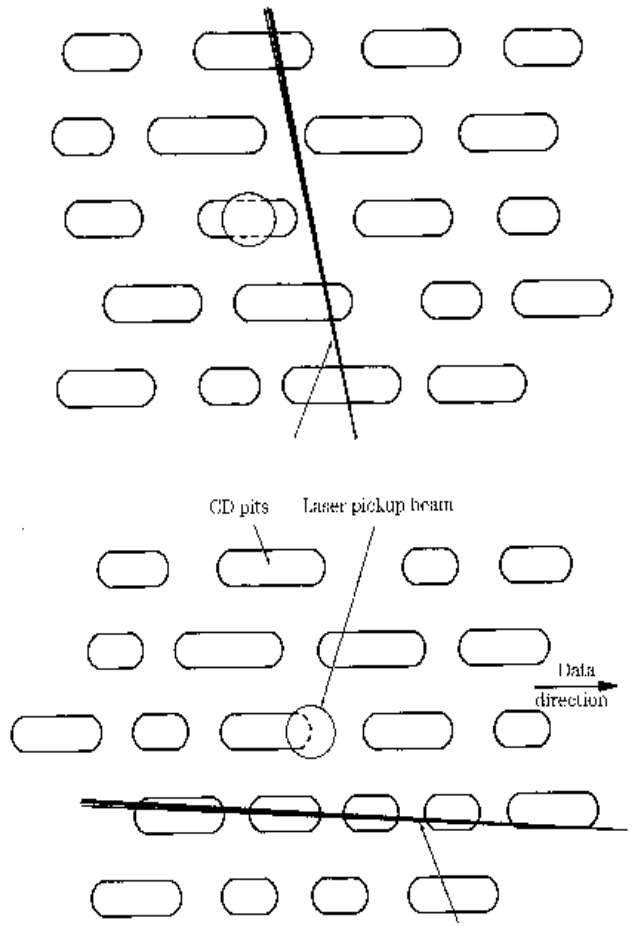
- Rabdom-bit errors
- Burst errors



1. Sources of Errors (c.1)

Parameters

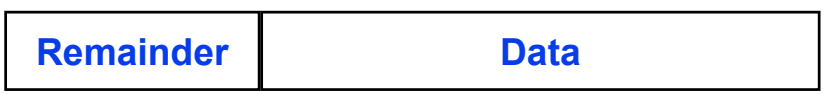
- Bit Error Rate (BER)
  - An optical disk system can contain error-correction algorithms able to handle a BER of 10^{-5} to 10^{-4}.
- Block Error Rate (BLER)
  - Measures thee number of blocks or frames of data per second that have at least one occurrence of uncorrected data.
- Burst Error Length (BERL)
  - Counts the number of consecutive blocks in error.
  - Over 80% of the errors in an optical disk might be burst errors.

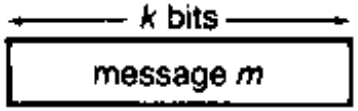


2. Cyclic Redundancy Check Code

Encoding

- Polynomial form m(X)
- Divided by an generation polynomial
  - $X^{n-k}m(X) = q(X)g(X) + r(X)$
- The transmitted Code
  - $V(X) = r(X) + X^{n-k}m(X)$

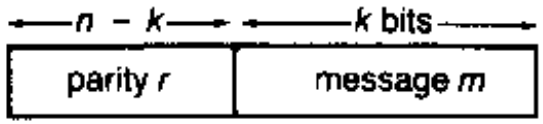


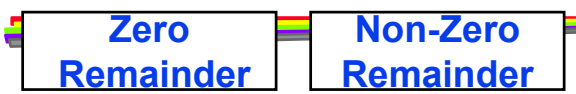


$\mathbf{m} = (m_0, m_1, m_2 \cdots m_{k-1})$
in polynomial form,
$\mathbf{m}(X) = m_0 + m_1 X + m_2 X^2 + \cdots + m_{k-1} X^{k-1}$
multiplying $\mathbf{m}(X)$ by X^{n-k}
$X^{n-k}\mathbf{m}(X) = m_0 X^{n-k} + m_1 X^{n-k+1} + \cdots + m_{k-1} X^{n-1}$
dividing $X^{n-k}\mathbf{m}(X)$ by $\mathbf{g}(X)$, the generation polynomial,
$X^{n-k}\mathbf{m}(X) = \mathbf{q}(X)\,\mathbf{g}(X) + \mathbf{r}(X)$
where $\mathbf{q}(X)$ and $\mathbf{r}(X)$ are quotient and remainder respectively,
where $\mathbf{r}(X) = r_0 + r_1 X + r_2 X^2 + \cdots + r_{n-k-1} X^{n-k-1}$
arranging previous equation and adding $\mathbf{r}(X)$,
$\mathbf{r}(X) + X^{n-k}\mathbf{m}(X) = \mathbf{q}(X)\,\mathbf{g}(X) + \mathbf{r}(X) + \mathbf{r}(X)$
however $\mathbf{r}(X) + \mathbf{r}(X) = 0$ thus,
$\mathbf{r}(X) + X^{n-k}\mathbf{m}(X) = \mathbf{q}(X)\,\mathbf{g}(X)$
thus $\mathbf{r}(X) + X^{n-k}\mathbf{m}(X)$ is a multiple of $\mathbf{g}(X)$.
$\mathbf{r}(X) + X^{n-k}\mathbf{m}(X)$ is the transmitted code polynomial $\mathbf{v}(X)$:

$$\mathbf{v}(X) = \mathbf{r}(X) + X^{n-k}\mathbf{m}(X) = r_0 + r_1 X + r_2 X^2 + \cdots + r_{n-k-1} X^{n-k-1} + m_0 X^{n-k} + m_1 X^{n-k+1} + \cdots m_{k-1} X^{n-1}$$

this corresponds to the transmitted code word:
$(r_0, r_1, r_2, \cdots r_{n-k-1}, m_0, m_1, m_2 \cdots m_{k-1})$





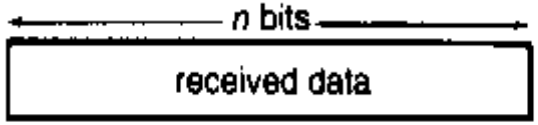
2. Cyclic Redundancy Check Code (c.1)

Decoding

- Get the polynomial form u(X)
- Obtain the remainder
  - u(X) = p(X)g(X) + s(X)
  - s(x) is the syndrome
- The s(X) can be suitable selected for correcting u(X)

Question

- The selection of g(x) for generating the suitable error pattern for s(X).



$\mathbf{u} = (u_1, u_2, u_3, u_4, \cdots u_{n-1})$
in polynomial form:
$\mathbf{u}(X) = u_0 + u_1 X + u_2 X^2 + \cdots + u_{n-1} X^{n-1}$
where $u_0, u_1, u_2, \cdots u_{n-k-1}$, are parity check bits and u_{n-k}, $\cdots u_{n-1}$, are information bits. The syndrome **s** is calculated by taking the mod 2 sum of the received parity bits and the parity bits formed from the received information. Thus, syndrome $\mathbf{s}(X)$ is equal to remainder of $\mathbf{u}(X)$ divided by $\mathbf{g}(X)$:

$$\mathbf{u}(X) = \mathbf{p}(X)\mathbf{g}(X) + \mathbf{s}(X)$$

a nonzero value for **s** detects an error. The difference between received (**u**) and transmitted (**v**) information is an error pattern **e**. From **e**, we can recover **v**, by using the syndrome for error correction

$$\mathbf{u}(X) = \mathbf{v}(X) + \mathbf{e}(X)$$

since $\mathbf{v}(X) = \mathbf{m}(X)\,\mathbf{g}(X)$,

$$\mathbf{u}(X) = \mathbf{m}(X)\,\mathbf{g}(X) + \mathbf{e}(X) = \mathbf{p}(X)\,\mathbf{g}(X) + \mathbf{s}(X)$$

thus $\mathbf{e}(X) = [\mathbf{p}(X) + \mathbf{m}(X)]\,\mathbf{g}(X) + \mathbf{s}(X)$

Thus when the error pattern is divided by the generation polynomial, the remainder is the syndrome, which can be used to correct errors. Note that the generation polynomial was chosen so that the error polynomial consists of an error pattern not divisible by **g**. The above derivations utilize the properties of modulo 2 arithmetic.

2. Cyclic Redundancy Check Code (c.2)

An Example

Given a message m = (1001) to be encoded, message polynomial $m(x) = x^3 + 1$. Multiplying by x^{n-k}, $x^3 m(x) = x^6 + x^3$.

0	0	0	1	0	0	1	m(x)

$x^3 + 1$

1	0	0	1	0	0	0	$x^3 m(x)$

$x^6 + x^3$

Given a generation polynomial $g(x) = x^3 + x^2 + 1$ division by g(x) is performed:

$$
\begin{array}{r|l}
x^3 + x^2 + 1 & x^3 + x^2 + x + 1 \\
\hline
 & x^6 \quad\quad\quad x^3 \\
 & x^6 + x^5 \quad + x^3 \\
\hline
 & \quad x^5 \\
 & \quad x^5 + x^4 \quad\quad + x^2 \\
\hline
 & \quad\quad x^4 \quad\quad + x^2 \\
 & \quad\quad x^4 + x^3 \quad\quad + x \\
\hline
 & \quad\quad\quad x^3 + x^2 + x \\
 & \quad\quad\quad x^3 + x^2 \quad\quad + 1 \\
\hline
 & \quad\quad\quad\quad\quad x + 1
\end{array}
$$

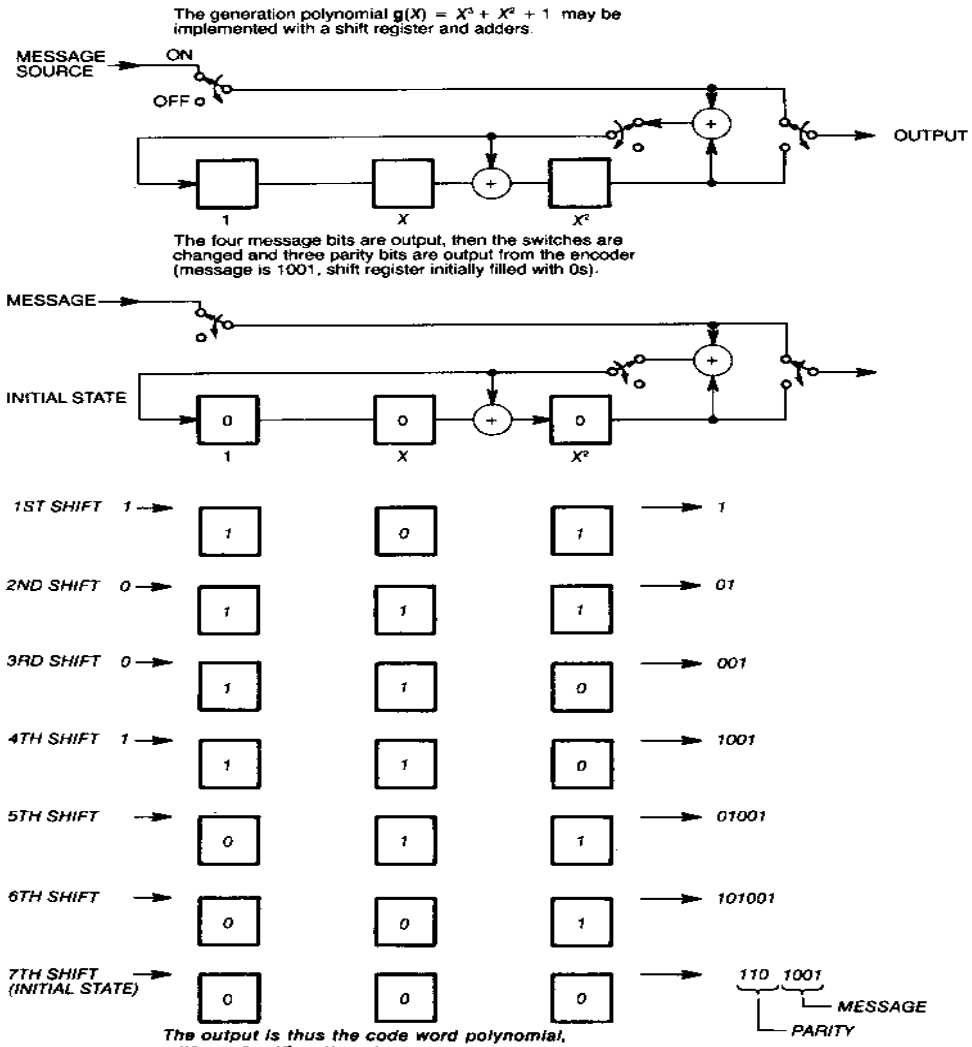
Remainder = r(x)

The code word polynomial

$(x) = x^3 m(x) + r(x)$

$= x^6 + x^3 + x + 1$

1	0	0	1	0	1	1



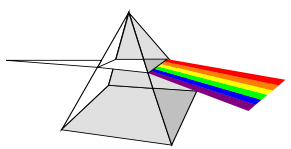
2. Cyclic Redundancy Check Code (c.3)

Error-Detection Analysis

- Given a k-bit data word with m (m=n-k) bits of CRCC, a code word of n bits is formed

1. The burst errors less than or equal to m bits are always detectable.
2. Detection probability of burst errors of m+1 bits is $1-2^{-m+1}$
3. Detection probability of burst errors longer than m+1 bits is $1-2^{-m}$.
4. Random errors up to three consecutive bits long can be detected.

- CRCC is quite reliable.
  - ex. 16 parity bits are generated for error detection $1 - 2^{-16} = 99.99\%$
- CRCC is typically used as error pointer to identify the number and extent of errors prior to other error-correction process.



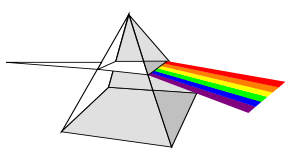
3. Error-Correction Codes

Block Codes

- Use algebraic methods
- The data are coded from the message coded from the a data block.

Convolutional Codes

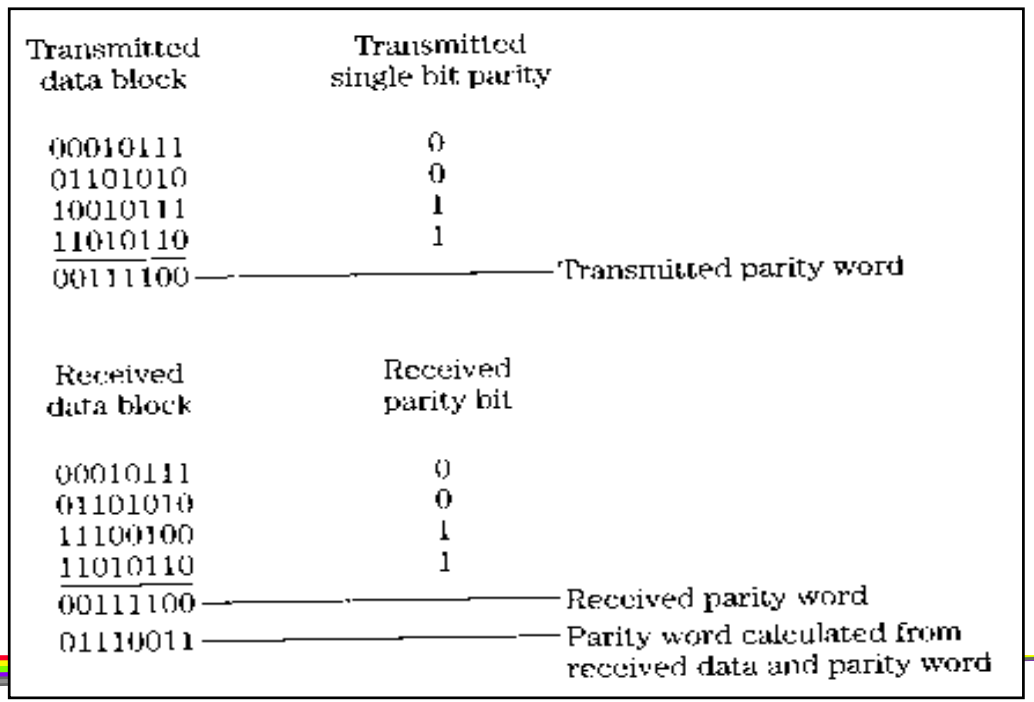
- Use probabilistic methods.
- The data are coded from the message present in the encoder at that time as well as previous message data.

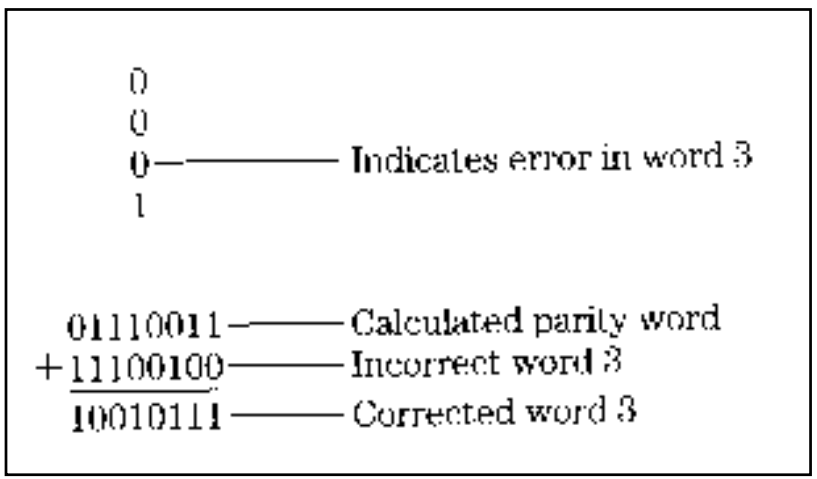


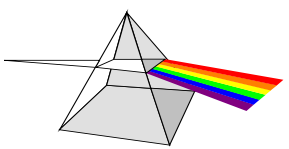
3. Error-Correction Codes-- Block Codes

Concepts

- Assemble a number of data words to form a block.
- Generate one or more parity words and append them to the block.
- Can be conceived as a binary message consolidated into a block with row and column parity.







3. Error-Correction Codes-- Block Codes (c.1)

Received data and two parity words

W_1	10
W_2	30
W_3	20
W_4	25
W_5	30
W_6	15
P	$130 = W_1 + W_2 + W_3 + W_4 + W_5 + W_6$
Q	$440 = 6W_1 + 5W_2 + 4W_3 + 3W_4 + 2W_5 + W_6$

Syndrome $S_1 = W_1 + W_2 + W_3 + W_4 + W_5 + W_6 - P = 10 + 30 + 20 + 25 + 30 + 15 - 130 = 0$

$S_2 = 6W_1 + 5W_2 + 4W_3 + 3W_4 + 2W_5 + W_6 - Q = 60 + 150 + 80 + 75 + 60 + 15 - 440 = 0$

(n,k) Block Codes

- *A message of k symbols is used to generate a larger n-bit symbol.*
- *If m parity blocks are included, the misimum distance is m+1.*
- *Detecting d number of errors requires a distance greater than or equal to (d+1)*
- *Correcting all combination of e errors requires a distance greater than or equal to (2e+1).*

Received data and two parity words

W_1	10
W_2	30
W_3	20
W_4	25
W_5	10
W_6	15
P	130
Q	440

$S_1 = -20$

$S_2 = -40$

Algebraically we see that

If $6S_1 = S_2$ then W_1 is erroneous
If $5S_1 = S_2$ then W_2 is erroneous
If $4S_1 = S_2$ then W_3 is erroneous
If $3S_1 = S_2$ then W_4 is erroneous
If $2S_1 = S_2$ then W_5 is erroneous
If $S_1 = S_2$ then W_6 is erroneous
If $S_1 \neq 0$ and $S_2 = 0$ then P is erroneous
If $S_1 = 0$ and $S_2 \neq 0$ then Q is erroneous

In this case $2S_1 = S_2$, W5 is erroneous, thus (as in single erasure case):

$S_1 = 10 + 30 + 20 + 25 + 0 + 15 - 130 = -30$

$W_5 = W_5' - S$
$= 0 - (-30)$
$= 30$ Corrected

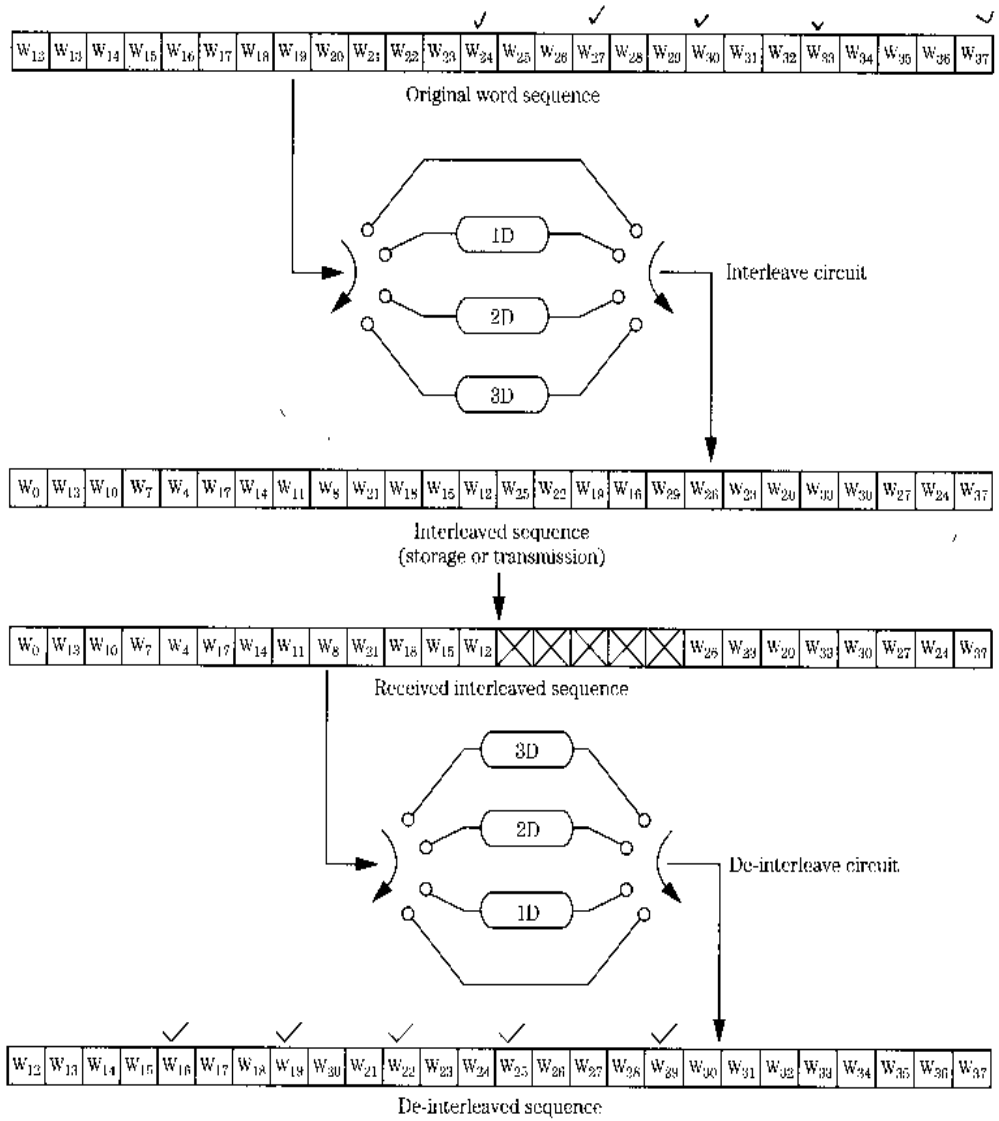
4. Interleaving

Observation

- Burst error losses both the data and redundancy bits.

Interleaving

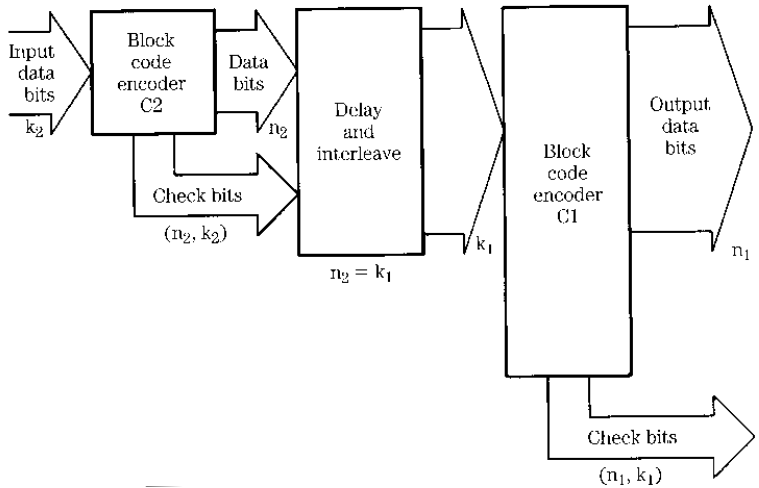
- Disperses data.
- Without interleaving, the amount of redundancy would be dictated by the size of the largest correctable burst error.
- Greatly increases burst error correctability.

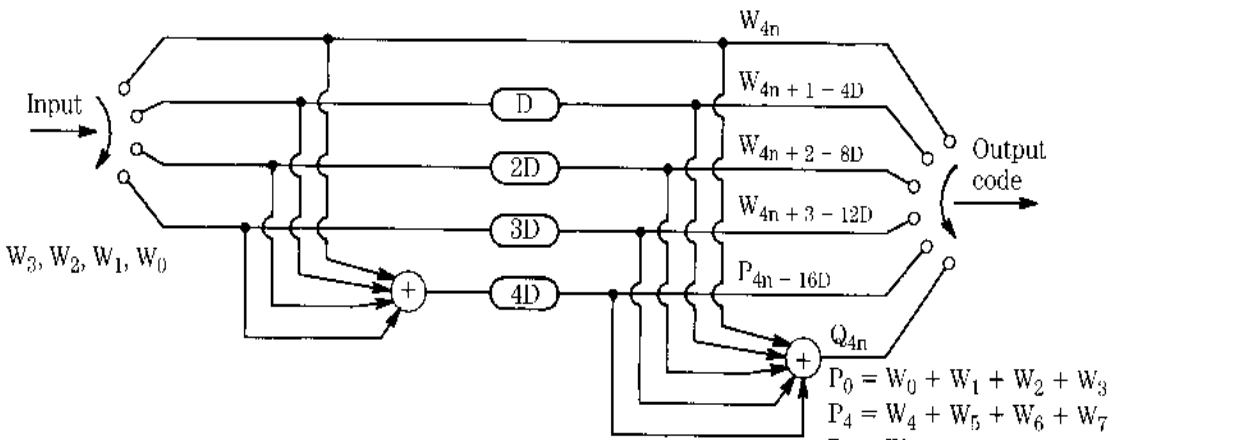


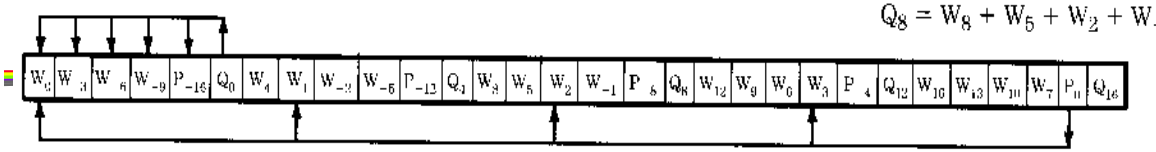
4. Interleaving

Cross-Interleaving

- *Interleaving might be inadequate when burst errors are accompanied by random errors.*
- *Although the burst us scattered, the random errors add additional errors in a given word, perhaps overloading the correction algorithm.*







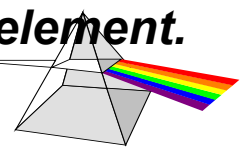
5. Reed-Solomen Codes

Overview

- Devised by Irving Reed and Gustave Solomon 1960 in MIT's Lincoln Lab.
- RS codes are cyclic codes that are multiple-error correcting codes.
- Use polynomials derived using finite field mathematics known as Galois Fields.

Galois Fields

- Named in hornor of the extraordinary and teormented mathematical genius Evariste Galois.
- Comprise a finite number of elements with special properties.
- Either multiplication or addition can be used to combine elements in the field.
- Such fields generally only exist when the number of elements is a prime number or a power of a prime number.
- There exists at least one element called a primitive such that every other element can be expressed as a power of this element.



5. Reed-Solomen Codes (c.1)

RS Codes Features

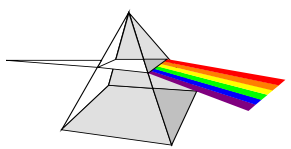
- Data are formed into symbols that are members of the Galois Field used by the code.
- The size of the Galois Field determines the number of symbols in the code, is based on the number of bits comprising a symbol.
  - Ex. 8-bit symbols are commonly used.
  - The code thus contains 2^8-1 or 255 eight-bit symbols.
  - A primitive polynomial often used in GF(2^8) systems is

    $x^8 + x^4 + x^3 + x^2 + 1$

An example

- Consider GF(2^3) with primitive element α and is the solution to the equation
  - $F(x) = x^3 + x + 1 = 0$

    $\alpha^3 + \alpha + 1 = 0$



5. Reed-Solomen Codes (c.2)

- All 3-bit symbols can be expressed as the powers of the primary elements.

$$
\begin{aligned}
0 &= 000 \\
1 &= 001 \\
\alpha &= 010 \\
\alpha^2 &= 100 \\
\alpha^3 = \alpha + 1 &= 011 \\
\alpha^4 = \alpha \cdot \alpha^3 = \alpha(\alpha + 1) = \alpha^2 + \alpha &= 110 \\
\alpha^5 = \alpha^2 + \alpha + 1 &= 111 \\
\alpha^6 = \alpha \cdot \alpha^5 = \alpha(\alpha^2 + \alpha + 1) = \alpha^3 + \alpha^2 + \alpha & \\
= \alpha + 1 + \alpha^2 + \alpha = \alpha^2 + 1 &= 101 \\
\alpha^7 = \alpha(\alpha^2 + 1) = \alpha^2 + \alpha = \alpha + 1 + \alpha = 1 &= 001 = 1
\end{aligned}
$$

5. Reed-Solomen Codes (c.3)

The RS code

- Suppose that A, B, C, D are data symbols and P and Q are parity symbols.
- The RS code will satisfy the following equations.

$$A+B+C+E+P+Q=0$$

$$\alpha^7 A+\alpha^6 B+\alpha^5 C+\alpha^4 D+\alpha^3 E+\alpha^2 P+\alpha Q=0$$

- Solving these equations yields

$$P = \alpha^6 A + \alpha B + \alpha^2 C + \alpha^5 D + \alpha^3 E$$

$$Q = \alpha^2 A + \alpha^3 B + \alpha^6 C + \alpha^4 D + \alpha^1 E$$

Element: / Bits:	0 000	a 010	a^2 100	a^3 011	a^4 110	a^5 111	a^6 101	$a^7=1$ 001
0 000	0 000	0 000	0 000	0 000	0 000	0 000	0 000	0 000
a 010	0 000	a^2 100	a^3 011	a^4 110	a^5 111	a^6 101	1 001	a 010
a^2 100	0 000	a^3 011	a^4 110	a^5 111	a^6 101	1 001	a 010	a^2 100
a^3 011	0 000	a^4 110	a^5 111	a^6 101	1 001	a 010	a^2 100	a^3 011
a^4 110	0 000	a^5 111	a^6 101	1 001	a 010	a^2 100	a^3 011	a^4 110
a^5 111	0 000	a^6 101	1 001	a 010	a^2 100	a^3 011	a^4 110	a^5 111
a^6 101	0 000	1 001	a 010	a^2 100	a^3 011	a^4 110	a^5 111	a^6 101
$a^7=1$ 001	0 000	a 010	a^2 100	a^3 011	a^4 110	a^5 111	a^6 101	1 001

5. Reed-Solomen Codes (c.4)

Checking Example

Input data	A	101	$a^6A = 111$	$a^2A = 010$
	B	100	$aB = 011$	$a^3B = 111$
	C	010	$a^2C = 011$	$a^6C = 001$
	D	100	$a^5D = 001$	$a^4D = 101$
	E	111	$a^3E = 010$	$aE = 101$
			100	100
Check symbols	P	100	← 100	
	Q	100		← 100

Codeword	A	101	$a^7A = 101$	
	B	100	$a^6B = 010$	
	C	010	$a^5C = 101$	
	D	100	$a^4D = 101$	
	E	111	$a^3E = 010$	
	P	100	$a^2P = 110$	
	Q	100	$aQ = 011$	
	$S_0 =$	000	$S_1 = 000$	← Both syndrom

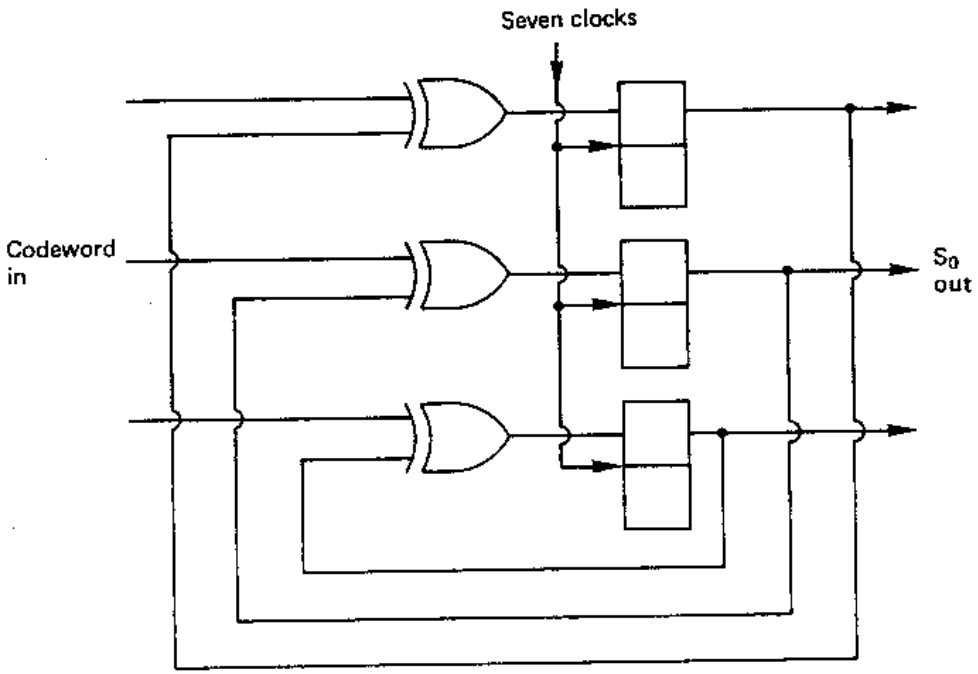
7	A	101	$a^7A = 101$	$\frac{S_1}{S_0} = \frac{a^4}{1} = a^4$
6	B	100	$a^6B = 010$	
5	C	010	$a^5C = 101$	
4	D′	101	$a^4D' = 011$	← $k = 4$
3	E	111	$a^3E = 010$	
2	P	100	$a^2P = 110$	$D' + S_0 = 101 + 001$
1	Q	100	$aQ = 011$	$D = 100$
	$S_0 =$	001	$S_1 = 110$	

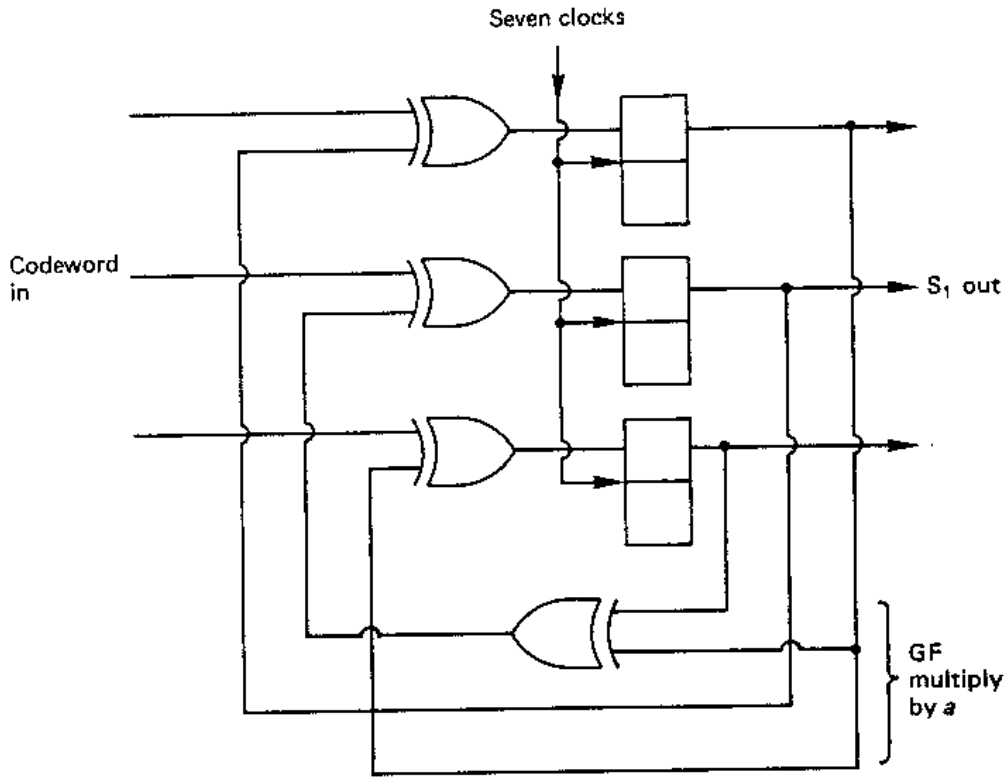
7	A	101	$a^7A = 101$	$\frac{S_1}{S_0} = \frac{1}{a^2} = \frac{1}{a^2} \times \frac{a^5}{a^5} = a^5$
6	B	100	$a^6B = 010$	
5	C′	110	$a^5C = 100$	← $k = 5$
4	D	100	$a^4D = 101$	
3	E	111	$a^3E = 010$	
2	P	100	$a^2P = 110$	$C' + S_0 = 110 + 100$
1	Q	100	$aQ = 011$	$C = 010$
	$S_0 =$	100	$S_1 = 001$	

7	A′	111	$a^7A = 111$	$\frac{S_1}{S_0} = \frac{a}{a} = 001 = a^7$
6	B	100	$a^6B = 010$	
5	C	010	$a^5C = 101$	
4	D	100	$a^4D = 101$	← $k = 7$
3	E	111	$a^3E = 010$	
2	P	100	$a^2P = 110$	$A' + S_0 = 111 + 010$
1	Q	100	$aQ = 011$	$A = 101$
	$S_0 =$	010	$S_1 = 010$	

5. Reed-Solomen Codes (c.5)

Implementation Circuits for S_0 and S_1





6. Cross-Interleave Reed-Solomon Code

A double error correction cross-interleave RS code.

- C2 is a (28, 24) code.
- C1 is a (32, 28) code.
- The encoder inputs 28 symbols and 32 symbols.
- A primitive polynomial often used in GF(2^8) systems is

  $$x^8 + x^4 + x^3 + x^2 + 1$$

- Minimum distance is five.
- CIRC might provide correction of up to 3874 bits. corresponding to an 2.5 mm defect.
- Good concealment can extend to 13282 bits corresponding to an 8.7 mm defect.
- Marginal cpncealment can extend to approximately 15500 bits.
- The CD standard sets a maximum 220 BLER errors.