

Chương 3

XỬ LÝ LỖI TRONG MẠNG TELEPROCESSING

- PHÁT HIỆN VÀ SỬA LỖI
- CÁC TÍNH CHẤT CHUNG CỦA MÃ PHÁT HIỆN VÀ SỬA LỖI
- MÃ ĐA THỨC
- MÃ VÒNG
- MÃ CHẬP

3.1. PHÁT HIỆN VÀ SỬA LỖI

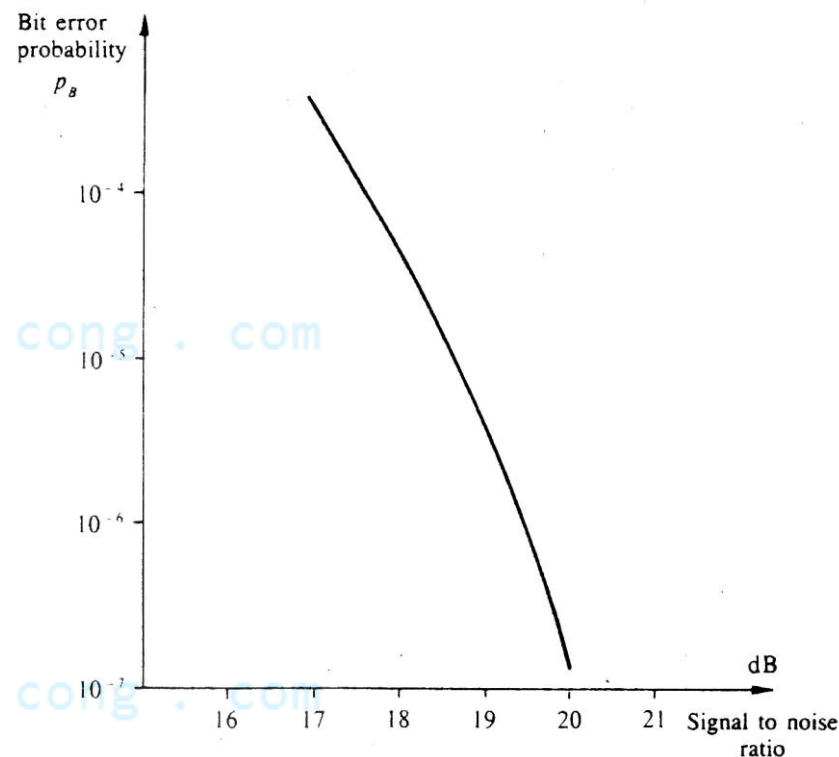
- Các nguyên nhân gây lỗi
- Các loại lỗi và tỷ lệ lỗi trên các đường truyền
- Phát hiện lỗi bằng ECHO
- Phát hiện và sửa lỗi bằng truyền lặp
- Phát hiện lỗi bằng mã Parity dọc
- Sửa lỗi bằng mã Parity dọc và ngang
- Sửa lỗi bằng cách truyền lại
- Phát hiện bản tin mất
- Độ dài bản tin tối ưu
- Các lỗi ở vị trí bắt đầu và kết thúc khung
- Kiểm tra Likelihood
- Chẩn đoán lỗi.

3.1.1. Các nguyên nhân gây lỗi

- Định nghĩa: *Lỗi là tất các sai sót có thể xảy ra trong quá trình hoạt động của mạng, ngoài ý muốn của người thiết kế cũng như nhà cung cấp dịch vụ.*
- Do Thiết bị
 - + DTE
 - + DCE
- Do Đường truyền
 - + Méo phi tuyến
 - + Nhiễu
- Do Giao thức điều khiển
- Ý tưởng khắc phục lỗi

3.1.2. Các loại lỗi và tỷ lệ lỗi trên các đường truyền

- Lỗi truyền dẫn sinh ra do nhiễu nhiệt được mô hình hóa là AWGN tỷ lệ nghịch với S/N
- Với độ rộng băng tần cho trước, khi tăng số mức sẽ dẫn đến tăng P_B , hay nói cách khác P_B sẽ tăng khi tốc độ truyền càng cao mà công suất trung bình không đổi



Lỗi sinh ra do nhiễu nhiệt thường là lỗi đơn, hoặc nhiều lần là vài bit lân cận

Histogram của các tỷ lệ lỗi truyền dẫn trên mạng chuyển mạch

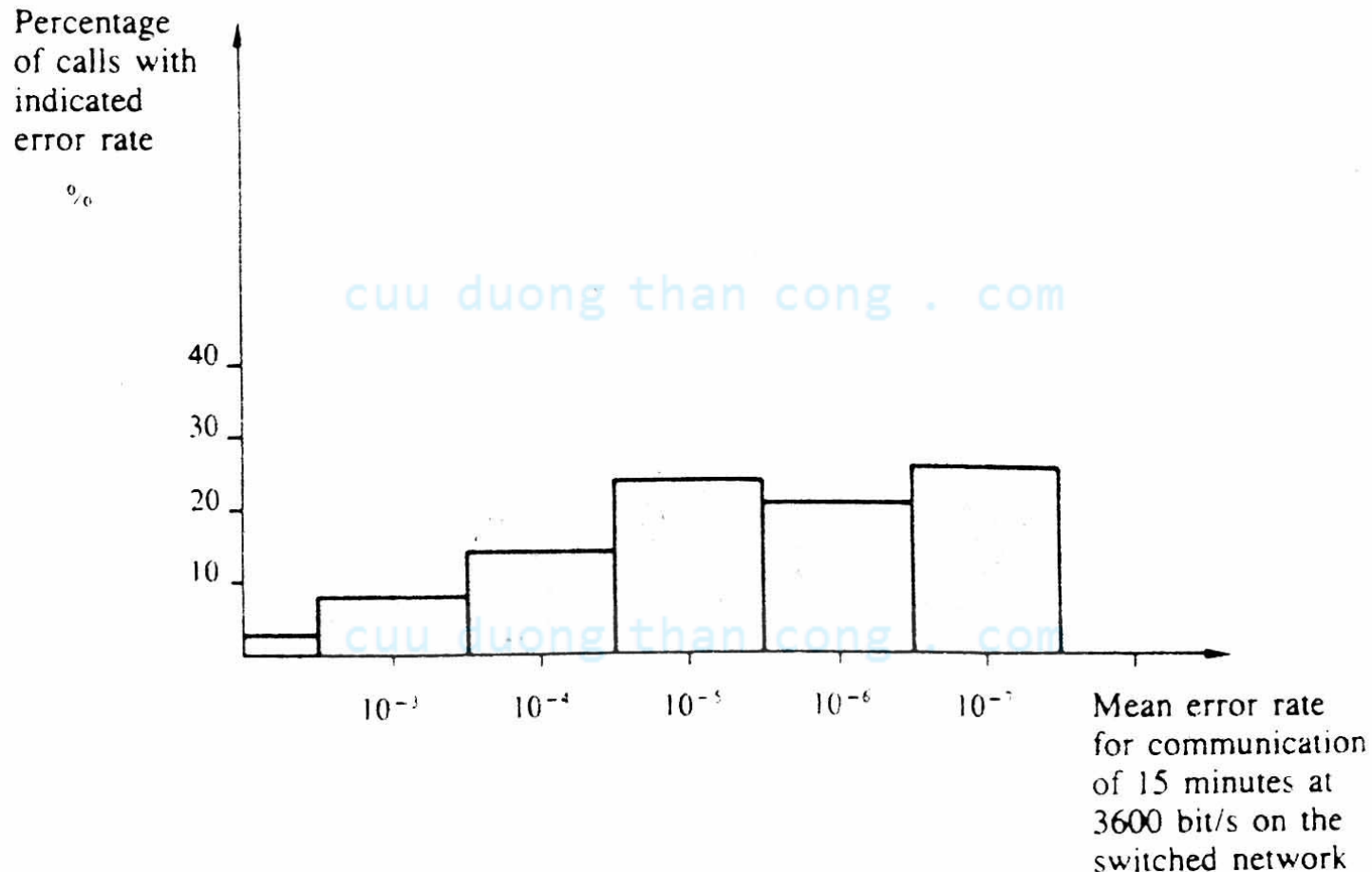


Figure 3.2 Histogram of error rates for transmission on a switched network line

Tỷ lệ lỗi trên đường truyền mạng chuyển mạch

- Lỗi sinh ra do nhiễu xung, chủ yếu do hệ thống chuyển mạch, thường ảnh hưởng trong một thời khoảng từ 10ms đến 50ms, và ở tốc độ 2400 bit/s sẽ gây ra lỗi từ 10 bit đến 100 bit

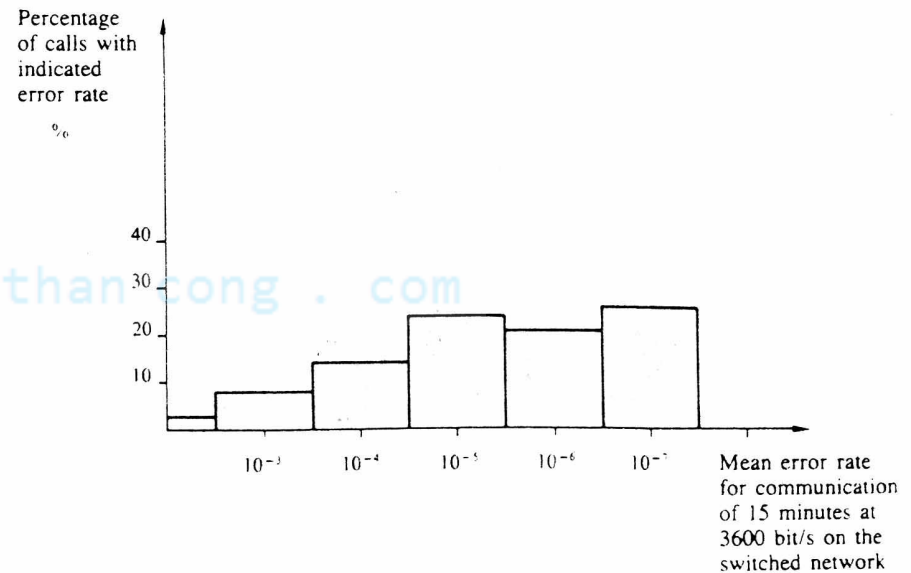


Figure 3.2 Histogram of error rates for transmission on a switched network line

Được đánh giá bằng tỷ lệ lỗi khối P_M .

CCITT đưa ra định mức $p_B = 10^{-3}$ cho đường truyền analog và $p_B = 10^{-6}$ cho đường truyền số PCM

3.1.3. Phát hiện lỗi bằng ECHO

- Bản tin phát đi sẽ được lưu giữ để so sánh với bản tin ECHO, nếu giống nhau nghĩa là bên thu nhận đúng

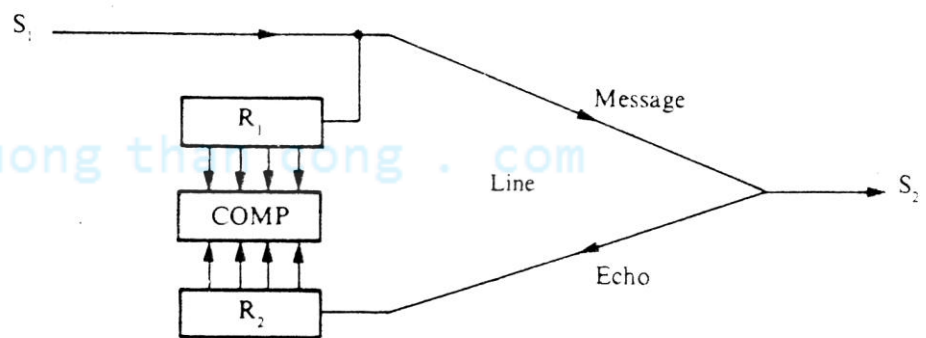


Figure 3.3 Error detection by echo

Tuy nhiên có thể xảy ra trường hợp bên thu đã nhận đúng nhưng bản tin ECHO bị sai do suy giảm nhiều

3.1.4. Phát hiện và sửa lỗi bằng truyền lặp

- Bên thu quyết định việc truyền lại bằng cách so sánh hai bản tin
- Có thể xảy ra trường hợp có một bản tin nhận đúng nhưng cứ phải truyền lại

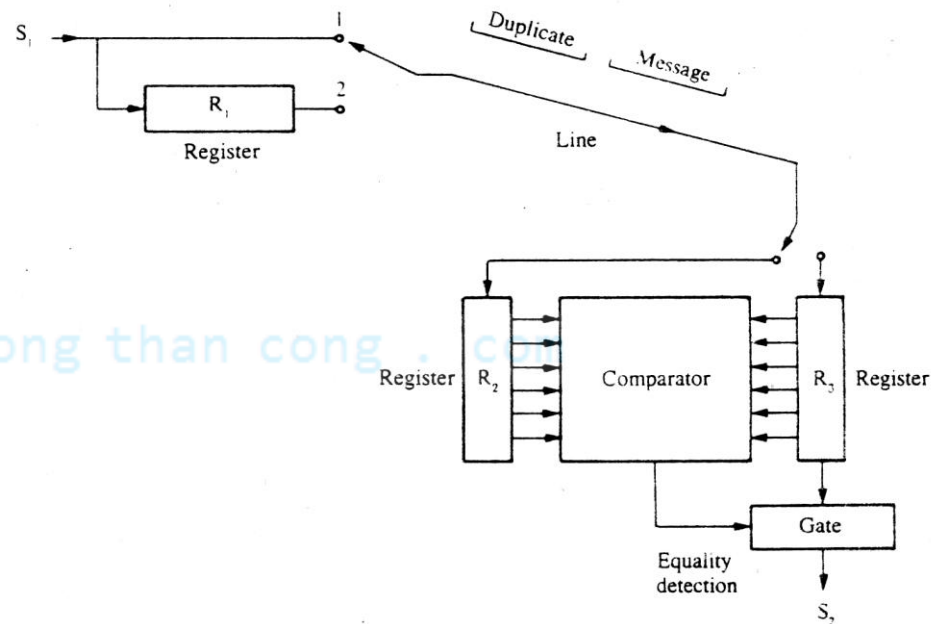


Figure 3.4 Error detection by repetition of messages

Xác suất lỗi khối của bản tin m -bit là $P_M = 1 - (1 - P_B)^m \cong mP_B$. Do xác suất không lỗi ở m -bit bất kỳ là $(1 - P_B)^m$.

Tính xác suất lỗi khối và lỗi không phát hiện được

Xác suất lỗi khối của bản tin m -bit là $P_M = 1 - (1 - P_B)^m \cong mP_B$. Do xác suất không lỗi ở m -bit bất kỳ là $(1 - P_B)^m$.

Một lỗi sẽ không phát hiện được nếu nó lặp lại trong bản tin sao, xác suất xảy ra là P_B^2 . Như vậy xác suất nhận được bản tin mà không có lỗi không phát hiện được là $(1 - P_B^2)^m$.

Từ đó tính được xác suất lỗi không phát hiện được là $P_M = 1 - (1 - P_B^2)^m \cong mP_B^2$.

Giả sử $p_B = 10^{-5}$, với bản tin 100-bit thì $p_M = 10^{-3}$, nhưng $p_N = 10^{-8}$ trong trường hợp truyền lặp.

Thu theo đa số

- Nếu có hai bản tin giống nhau thì nó được chọn, chỉ truyền lại khi cả ba bản tin đều khác nhau.
- Nhược điểm chủ yếu là thời gian truyền tăng gấp ba lần

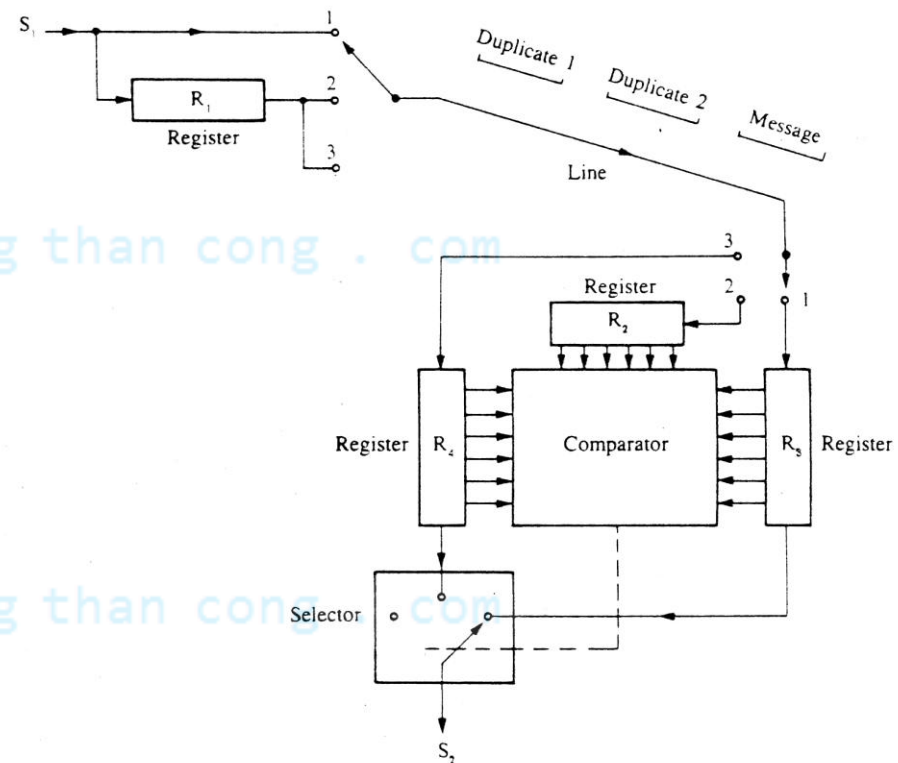


Figure 3.5 Error detection and correction by repetition with voting

Sử dụng scrambler

- Bản tin được gửi đi trên line 1, làm trễ N-bit nhờ bộ ghi dịch và cộng modulo 2 với bản tin đã scrambling sau đó gửi đi trên line 2.
- Ở đầu thu bản tin được làm trễ qua bộ ghi dịch N-bit như ở máy phát và scrambling để cộng với bản tin qua line 2.

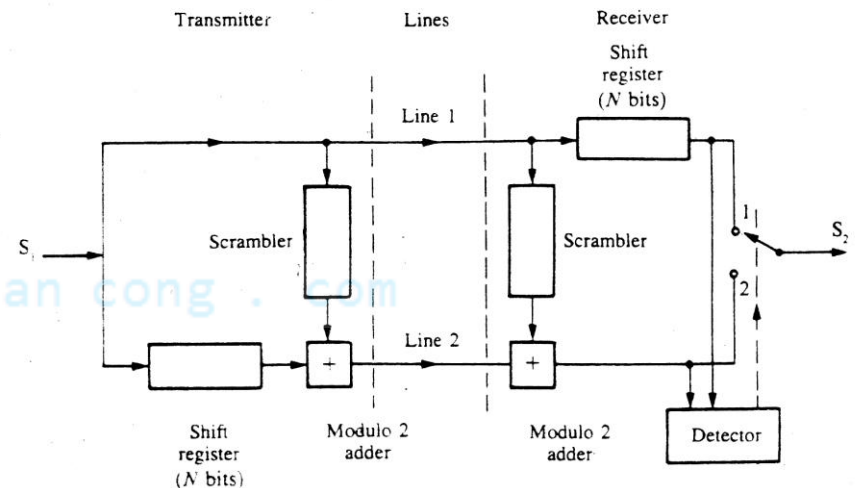


Figure 3.6 Error correction system with simple repetition

Kết quả so sánh cho phép phát hiện các bit lỗi đơn và sửa lỗi

Bộ scrambler

- Bao gồm các bộ ghi dịch và bộ cộng mắc nối tiếp, nhằm mục đích giảm sự tương quan giữa bản tin và bản sao được làm trễ N-bit của nó

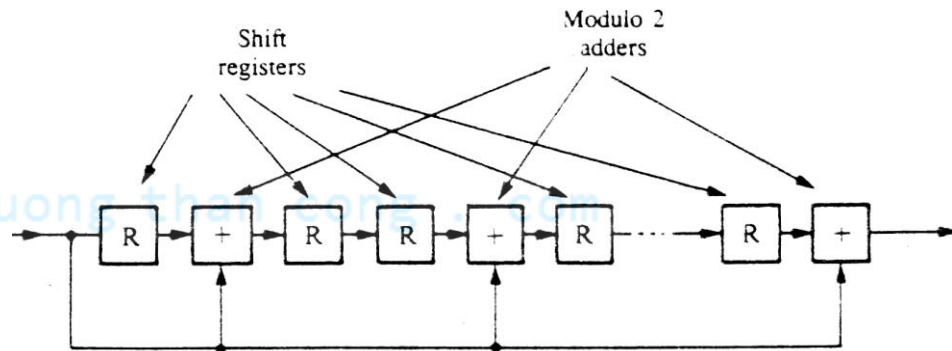


Figure 3.7 Scrambler

Giả sử có lỗi ở Line 1 ?

Giả sử có lỗi ở Line 2 ?

Giải pháp kết hợp

- Sử dụng chung đường truyền để gửi bản tin và bản sao của nó

So sánh hai sơ đồ ?

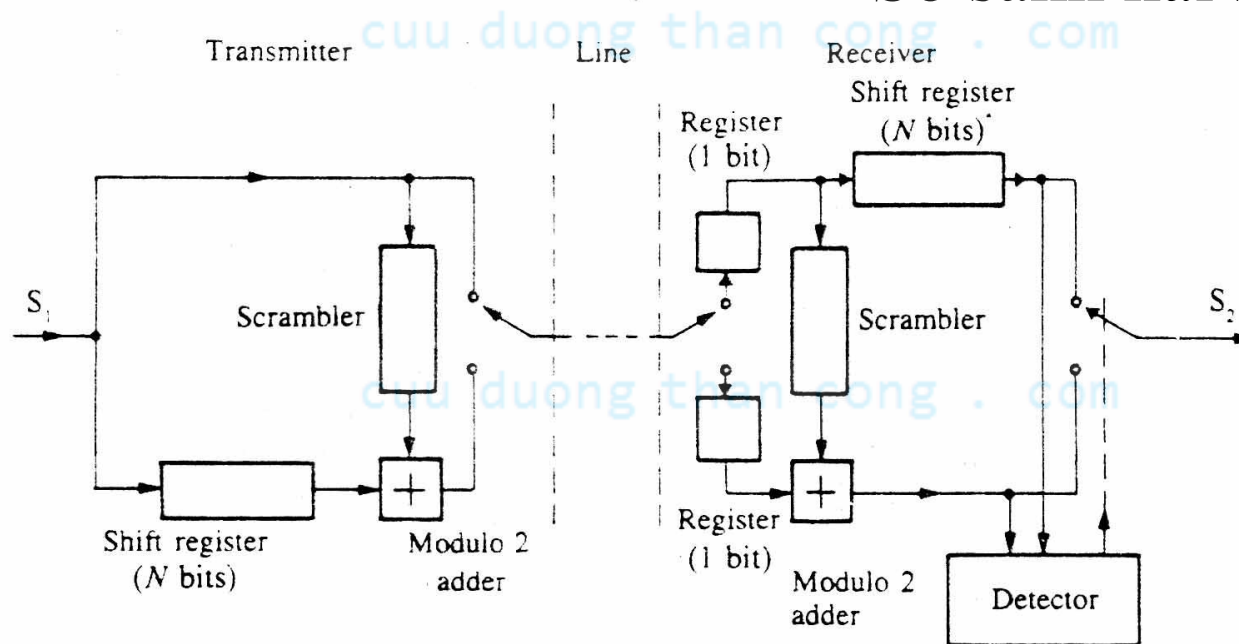


Figure 3.8 Error correction system with simple repetition and a single line

3.1.5. Phát hiện lỗi bằng mã Parity dọc

- Bản tin mã hóa được chia thành các khối m-bit, mã hóa bằng các khối n-bit

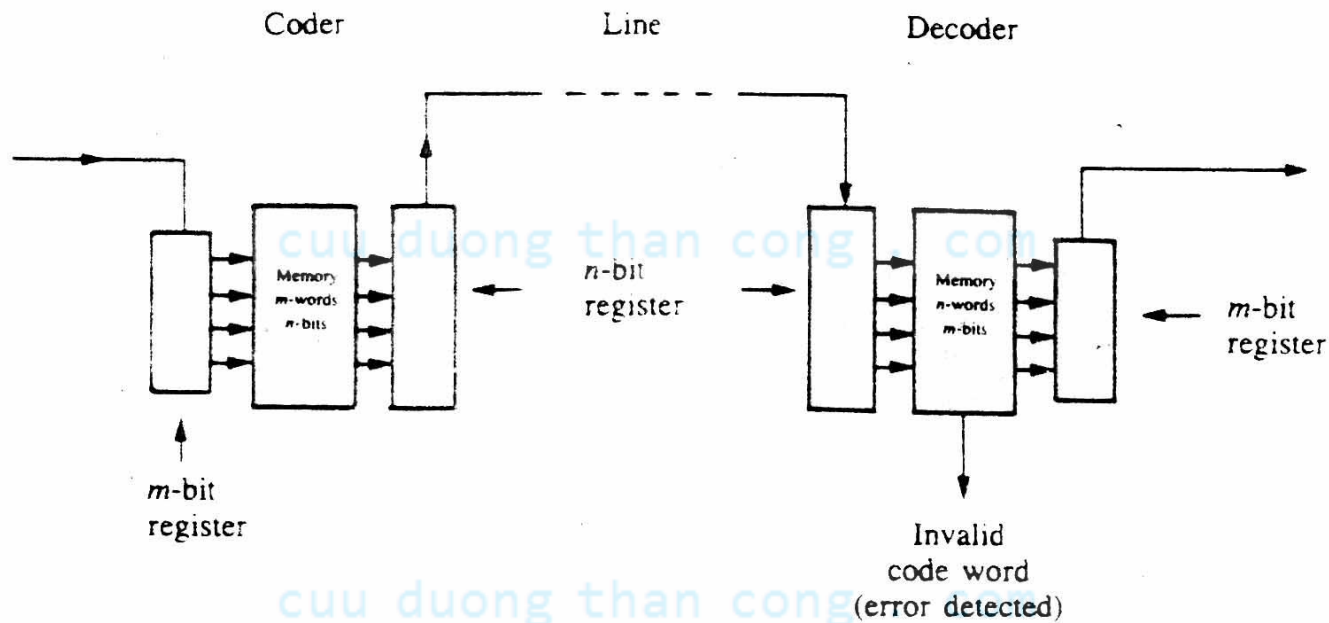


Figure 3.9 The principle of coding and decoding for error detection

Việc lựa chọn r -bit parity nhằm dễ dàng phát hiện lỗi ở đầu thu, $n = m + r$

Nguyên lý truyền dẫn với mã phát hiện lỗi

Source message

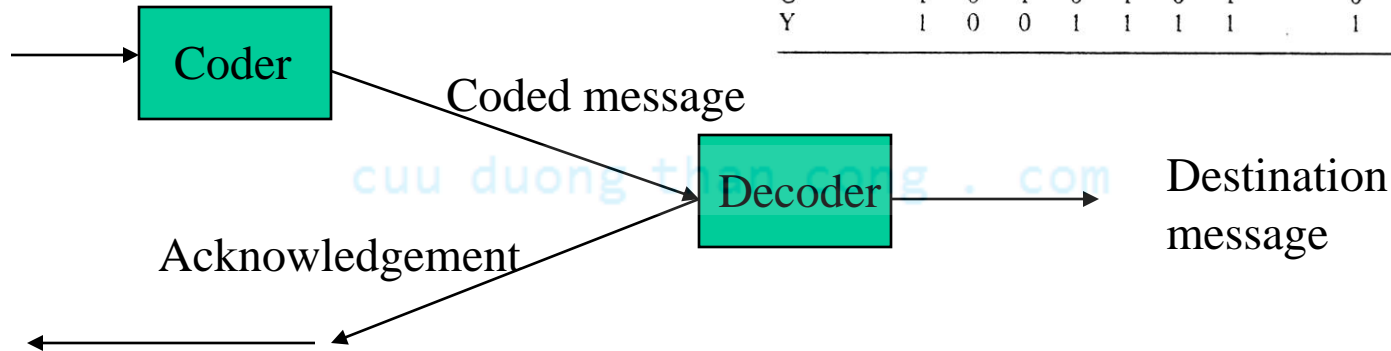


Table 3.1 Parity checking with the ASCII code.

| Symbol | ASCII character | | | | | | | Parity bit |
|--------|-----------------|---|---|---|---|---|---|------------|
| C | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 3 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| U | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| Y | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

Table 3.2 Word error and undetected error probabilities for 8-bit words with parity checking.

| | Bit error probability p_B | | | |
|---------------------------------------|--------------------------------|---------------------|---------------------|----------------------|
| | 10^{-1} | 10^{-3} | 10^{-5} | 10^{-7} |
| Word error probability p_M | $5.7 \cdot 10^{-1}$ | $8 \cdot 10^{-3}$ | $8 \cdot 10^{-5}$ | $8 \cdot 10^{-7}$ |
| Undetected error probability p_N | $1.4 \cdot 10^{-2}$ | $2.8 \cdot 10^{-5}$ | $2.8 \cdot 10^{-9}$ | $2.8 \cdot 10^{-13}$ |

- Với 1 bit dư, sẽ không phát hiện được các lỗi chẵn
- $p_N = ?$

3.1.6. Sửa lỗi bằng mã Parity dọc và ngang

- Cho phép phát hiện vị trí lỗi đơn

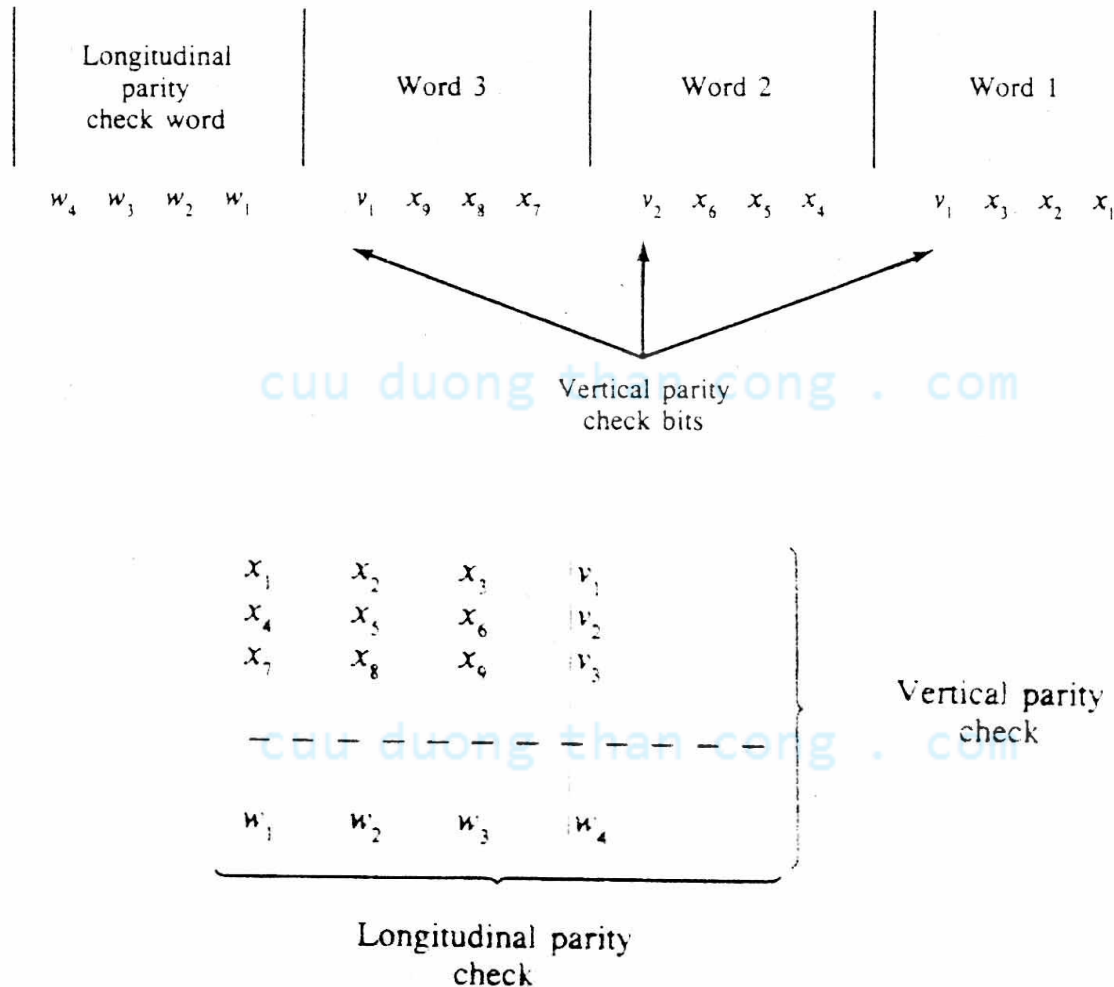


Figure 3.12 Two-dimensional representation of the method using longitudinal and vertical parity checks

Khả năng sửa lỗi

- Nếu các bit lỗi xuất hiện trên cùng một cột thì không sửa được
- Tính p_M và p_N ?

| | 1 | 2 | 3 | 4 | 5 | | 1 | 2 | 3 | 4 | 5 | |
|-------|---|---|---|---|---|--|-------|---|---|---|---|---------|
| 1 | 1 | 1 | 0 | 0 | 0 | | 1 | 1 | 0 | 0 | 0 | |
| 2 | 1 | 1 | 1 | 0 | 1 | | 1 | 1 | 0 | 0 | 1 | ← Error |
| 3 | 0 | 1 | 1 | 1 | 1 | | 0 | 1 | 1 | 1 | 1 | |
| 4 | 1 | 1 | 1 | 1 | 0 | | 1 | 1 | 1 | 1 | 0 | |
| <hr/> | | | | | | | <hr/> | | | | | |
| 5 | 1 | 0 | 1 | 0 | 0 | | 1 | 0 | 1 | 0 | 0 | |
| | | | | | | | | | ↑ Error | | | |

(A) Error free transmission

(B) Transmission with an error in row 2 column 3

Figure 3.13 Locating an error with longitudinal and vertical parity checks

| | 1 | 2 | 3 | 4 | 5 | |
|-------|---|---|---|---|---|---------|
| 1 | 1 | 1 | 0 | 0 | 0 | |
| 2 | 1 | 1 | 0 | 0 | 1 | ← Error |
| 3 | 0 | 1 | 0 | 1 | 1 | ← Error |
| 4 | 1 | 1 | 1 | 1 | 0 | |
| <hr/> | | | | | | |
| 5 | 1 | 0 | 1 | 0 | 0 | |
| | | | ↑ Undetected error | | | |

Figure 3.14 Detection of two errors by longitudinal and vertical parity checks

3.1.7. Sửa lỗi bằng cách truyền lại

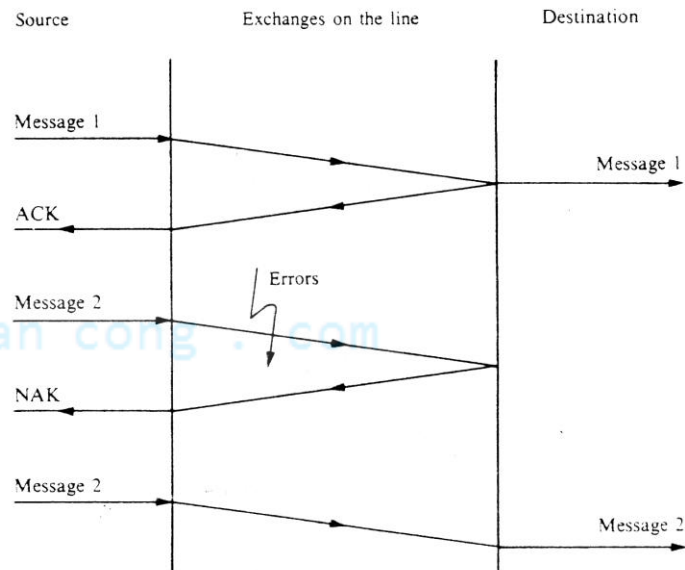


Figure 3.15 Error correction by retransmission

3.1.8. Phát hiện bản tin mất

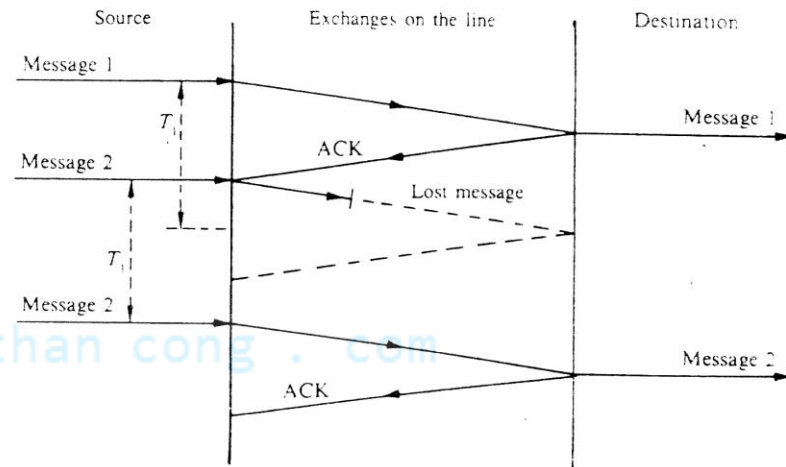


Figure 3.16 Detection of lost messages using a timer

Trường hợp mất bản tin phúc đáp

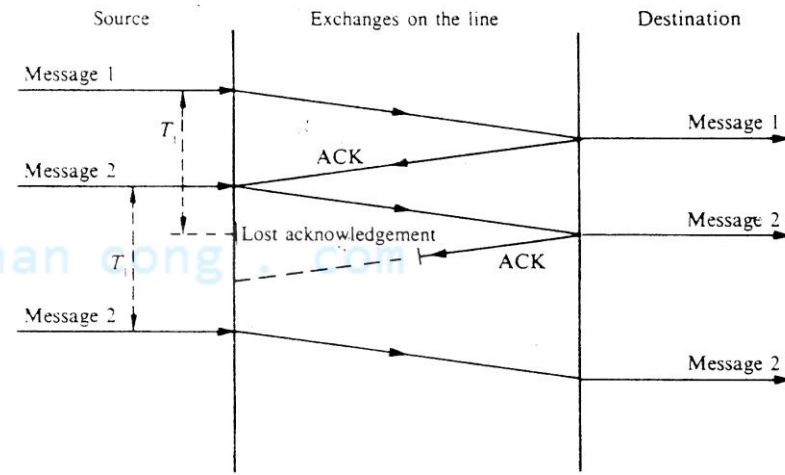


Figure 3.17 Duplication of messages produced by the loss of an acknowledgement

3.1.9. Độ dài bản tin tối ưu

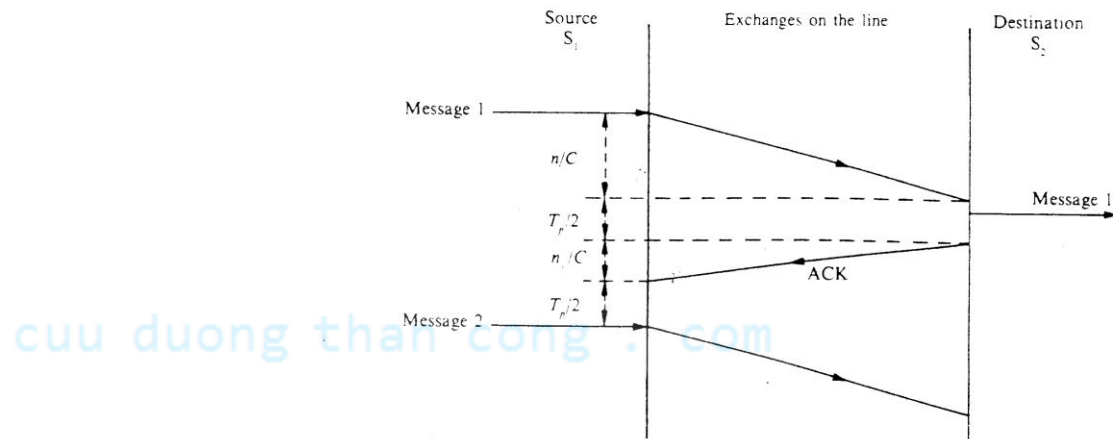


Figure 3.18 Delay between two successive messages without errors with a *send and wait* procedure

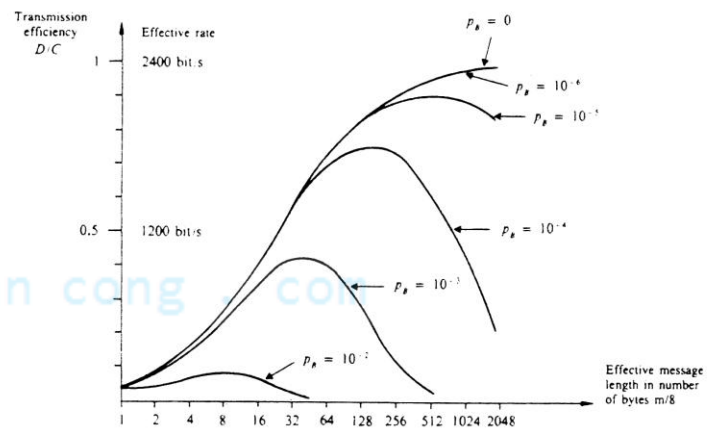


Figure 3.19 Transmission efficiency versus bit error probability and message length with a send and wait procedure

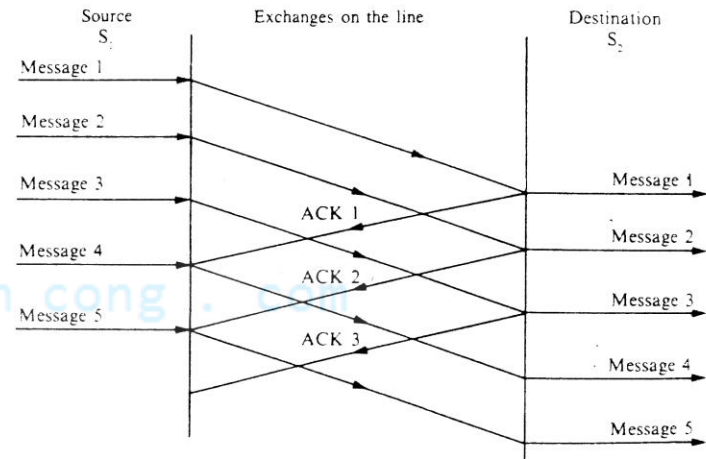


Figure 3.20 Message with anticipation of acknowledgements. *Simultaneous bidirectional transmission*

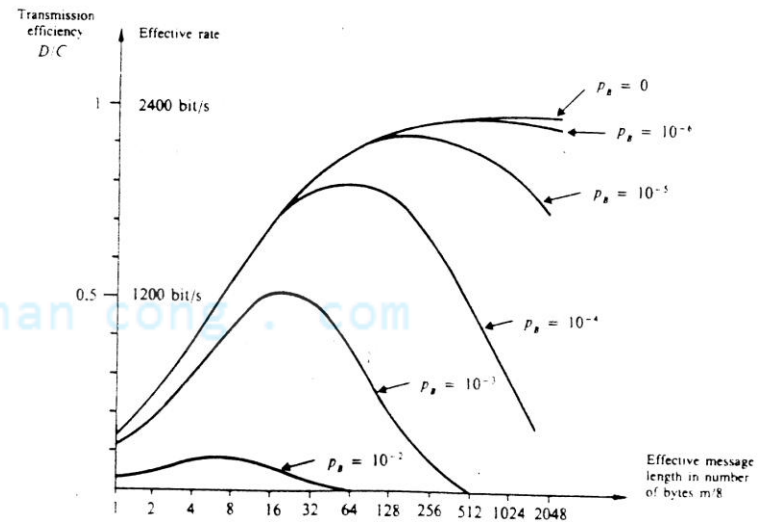


Figure 3.21 Transmission efficiency as a function of bit error probability and message length with an anticipation and simple reject procedure

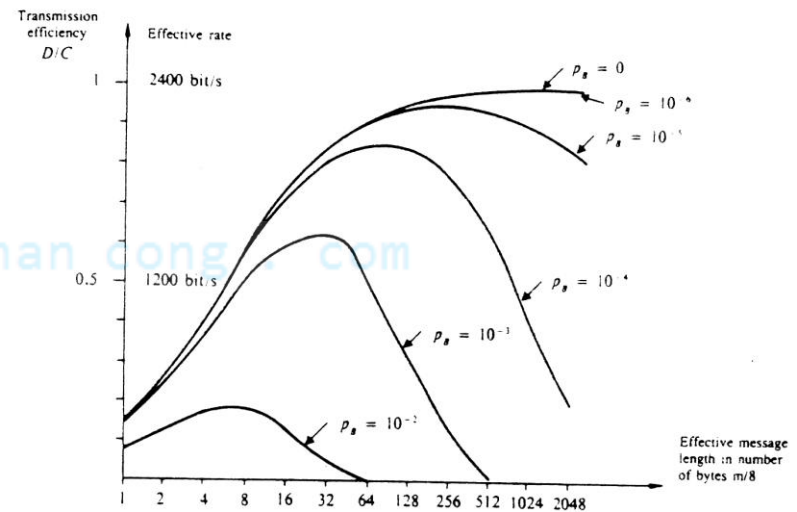


Figure 3.22 Transmission efficiency as a function of bit error probability and message length, with an anticipation and selective reject procedure

3.1.10. Các lỗi ở vị trí bắt đầu và kết thúc khung

- Trong trường hợp các bản tin được truyền đi trong các khung, thông thường các bit bắt đầu khung hay bị mất và ở phần kết thúc khung hay xuất hiện các bit thừa
- Các lỗi này làm cho việc nhận khung bị sai sót và dẫn đến mất gói
- Cần có các giao thức chuẩn để xử lý khung chính xác, hạn chế lỗi bằng các phương pháp kiểm tra khung

3.1.11. Kiểm tra Likelihood

- Bản thân các hệ thống phát hiện lỗi mặc dù có thể đạt được hiệu quả cao nhưng không thể bảo đảm vận hành không lỗi (error-free operation), cần có các thiết bị hỗ trợ được thiết kế theo phương pháp kiểm tra Likelihood
- Ở mức thấp có thể kiểm tra kích thước khung (là bội của 8 trong SDLC), kích thước gói hoặc một số trường kiểm tra và dùng thủ tục cấm để từ chối các khung, gói sai chuẩn.
- Ở mức cao hơn có thể kiểm tra dựa vào thứ tự các kiểu khung hay các kiểu gói bắt buộc phải xuất hiện theo giao thức điều khiển đường truyền để khởi động các thủ tục hồi phục

3.1.12. Chẩn đoán lỗi.

- Nhằm phát hiện trực tiếp các sai sót của phần cứng (breakdowns) hoặc phần mềm (faults).
- Khác với kiểu phát hiện và sửa lỗi (trực tiếp) của hệ thống chỉ phát hiện gián tiếp các sai sót dựa vào sự lặp lại một kiểu lỗi.
- Có nhiều kiểu chẩn đoán dựa vào các phương pháp khác nhau:
 - + Kiểm tra một chức năng cụ thể
 - + Kiểm tra tone trên đường dài
 - + Kiểm tra mức DC trên đường truyền nội hạt.
 - + watchdog.

Các chiến lược truyền tin

- ARQ (Automatic Repeat Queue) sử dụng mã phát hiện lỗi.
 - Stop & Wait
 - Pull Back N
 - Selective Repeat (Continuous)
- FEC (Forward Error Correct) sử dụng mã sửa lỗi.

3.2. CÁC TÍNH CHẤT CHUNG CỦA MÃ PHÁT HIỆN VÀ SỬA LỖI

- Khái quát
- Phân loại mã
- Khả năng phát hiện và sửa lỗi của mã khối.
Khoảng cách Hamming
- Mã khối tuyến tính hệ thống
- Nguyên lý mã hóa với mã tuyến tính hệ thống
- Nguyên lý phát hiện lỗi với mã khối tuyến tính hệ thống
- Nguyên lý sửa lỗi với mã khối tuyến tính hệ thống
- Mã Hamming

3.2.1. Khái quát

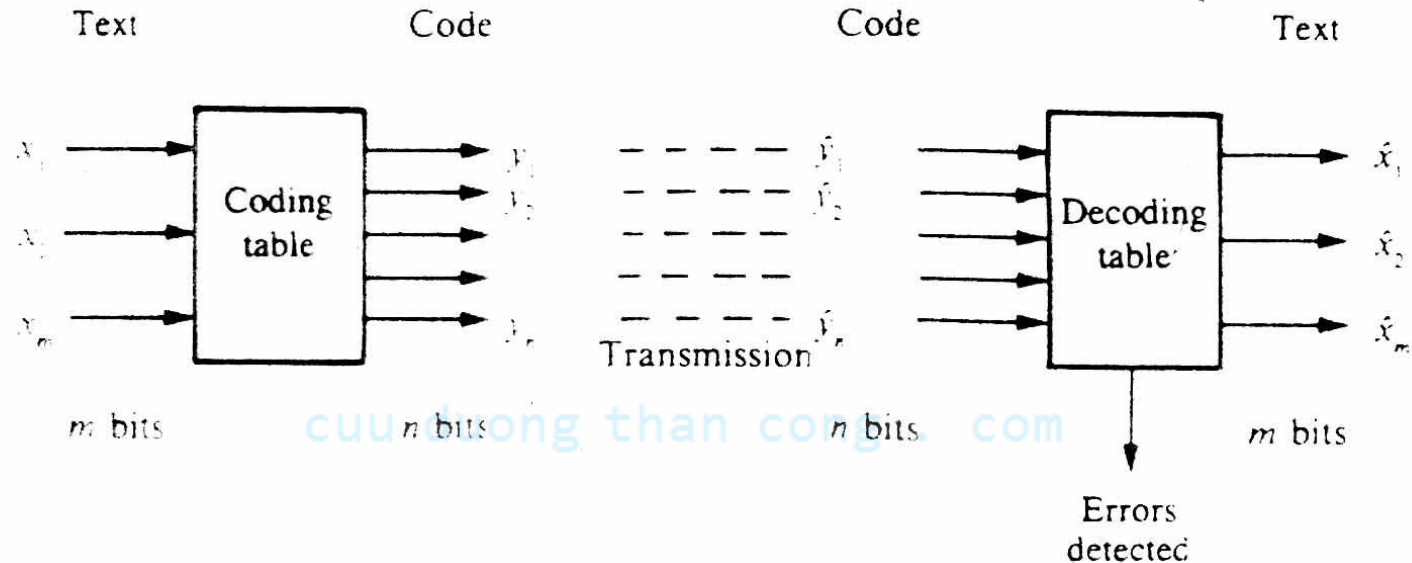


Figure 3.23 The principle of coding and decoding

- Thêm vào các bit dư để kiểm tra lỗi, phát hiện và sửa lỗi.
- Chấp nhận trả giá về thời gian truyền (giảm tốc độ) để đạt được chất lượng truyền dẫn cao.

3.2.2. Phân loại mã

- Mã khối và tính chất hệ thống
- Mã chập hoặc recurrent.

cuu duong than cong . com

cuu duong than cong . com

Cơ sở toán học

- Không gian vector V_n trên trường số nhị phân
 - Xét các n-tuples có cả thảy 2^n bộ khác nhau
 - Hai toán tử cộng modulo 2 và nhân modulo 2 được định nghĩa theo bảng sau :

| A | B | $A+B$ | $A.B$ | | $A+B+B=A$ | | $A+B+A=B$ | | |
|---|---|-------|-------|--|-----------|--|-----------|--|--|
| 0 | 0 | 0 | 0 | | 0 | | 0 | | |
| 0 | 1 | 1 | 0 | | 0 | | 1 | | |
| 1 | 0 | 1 | 0 | | 1 | | 0 | | |
| 1 | 1 | 0 | 1 | | 1 | | 1 | | |

Không gian con S của V_n

- Không gian con S của V_n :

- Chứa vector không $v_0 \in S$

- Khép kín $v_i, v_j \in S \Rightarrow v_i \oplus v_j \in S$

Thí dụ V_4 :

| | | | |
|------|------|------|------|
| 0000 | 0001 | 0010 | 0011 |
| 0100 | 0101 | 0110 | 0111 |
| 1000 | 1001 | 1010 | 1011 |
| 1100 | 1101 | 1110 | 1111 |

| v_i | S1 | Hà | Tình | |
|-------------------|------|------|------|--|
| V_0 | 0000 | 0000 | 0000 | |
| v_1 | 0100 | 0001 | 0010 | |
| v_2 | 1000 | 0101 | 0100 | |
| $v_3 = v_1 + v_2$ | 1100 | 0100 | 0110 | |

cuu duong than cong . com

cuu duong than cong . com

Mã khối tuyến tính

- Định nghĩa : mã khối tuyến tính là không gian con của không gian V_n . Được đặc trưng bởi (n,k) với n là kích thước vector từ mã và k là kích thước vector mang tin, tỷ số mã $R = k/n$.
- Mã hoá và giải mã
- Ma trận sinh
- Tính chất hệ thống

Phát hiện và sửa lỗi

- Ma trận kiểm tra
- Bảng triệu chứng (syndrome)

[cuu duong than cong . com](http://cuuduongthancong.com)

[cuu duong than cong . com](http://cuuduongthancong.com)

Bảng syndrome của mã (6,3)

| e/Vi | V0 | V1 | | V7 | Syndro me |
|--------|--------|--------|-----|--------|--------------|
| 000000 | 000000 | 110100 | ... | 000111 | 000 |
| 100000 | 100000 | 010100 | | 100111 | 100 |
| 010000 | 010000 | 100100 | | 010111 | 010 |
| 001000 | 001000 | 111100 | | 001111 | 001 |
| 000100 | 000100 | 110000 | | 000011 | 110 |
| 000010 | 000010 | 110110 | | 000101 | 011 |
| 000001 | 000001 | 110101 | | 000110 | 101 |
| 010010 | 010010 | 100110 | | 010101 | 001 |
| 101001 | 101001 | 011101 | | 101110 | 000 |

Các vector lỗi có triệu chứng bằng không

| e/V _i | V ₀ | V ₁ | V ₂ | V ₃ | | V ₇ | Syndro me |
|------------------|----------------|----------------|----------------|----------------|--|----------------|--------------|
| 110100 | 110100 | 000000 | 011010 | | | 110011 | 000 |
| 011010 | 011010 | 101110 | 000000 | | | 011101 | 000 |
| 101110 | 101110 | | | | | | 000 |
| 101001 | 101001 | 011101 | 110100 | | | 101110 | 000 |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Khả năng phát hiện và sửa lỗi của mã khối tuyến tính

- Hàm trọng Hamming
- Khoảng cách Hamming
- Khả năng phát hiện lỗi tối đa
- Khả năng sửa lỗi
- Khả năng phát hiện và sửa lỗi đồng thời
- Phân bố Hamming
- Xác suất lỗi không phát hiện được

Hàm trọng Hamming

- Hàm trọng Hamming $W(V_i)$ của vector từ mã V_i là số digit khác không của V_i .
- Thí dụ:

| | m_i | V_i | $W(V_i)$ |
|-------------------------|-------|---------|----------|
| V_0 | 000 | 000 000 | |
| V_1 | 100 | 110 100 | 3 |
| V_2 | 010 | 011 010 | 3 |
| $V_3 = V_1 + V_2$ | 110 | 101 110 | 4 |
| V_4 | 001 | 101 001 | 3 |
| $V_5 = V_1 + V_4$ | 101 | 011 101 | 4 |
| $V_6 = V_2 + V_4$ | 011 | 110 011 | 4 |
| $V_7 = V_1 + V_2 + V_4$ | 111 | 000 111 | 3 |

cuu duong than cong . com

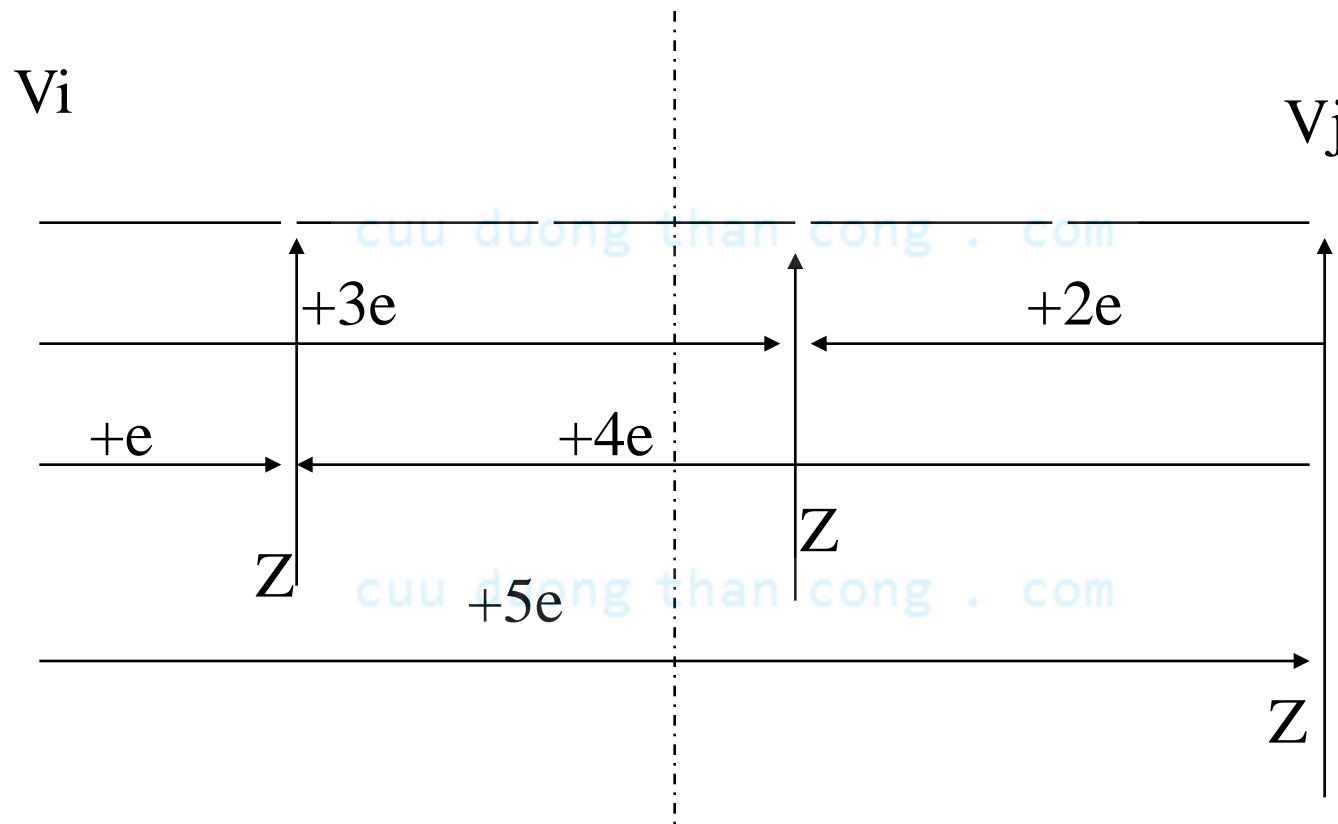
cuu duong than cong . com

Khoảng cách Hamming

- Khoảng cách Hamming giữa 2 vector từ mã V_i và V_j , ký hiệu $d(V_i, V_j)$ bằng số digit khác nhau giữa V_i và V_j .
- Nhận xét:
 - $d(V_i, V_j) = W(V_i + V_j)$.
 - $d_{\min} = W_{\min}$.

$$\begin{aligned}
 d_{\min} &= \min_{d(V_i, V_j); V_i, V_j \in (n, k)} \\
 &= \min_{W(V_i); V_i \in (n, k)} \exists W_{\min}
 \end{aligned}$$

Nguyên lý Maximum Likelihood



3.2.3. Khả năng phát hiện và sửa lỗi của mã khối. Khoảng cách Hamming

- Khả năng phát hiện lỗi tối đa: $e = d_{\min} - 1$
- Khả năng sửa lỗi tối đa: $t = \text{int}[(d_{\min} - 1)/2]$
- Khả năng phát hiện α và sửa lỗi β đồng thời phải thỏa mã điều kiện: $\alpha + \beta + 1 \leq d_{\min}$.

Với $d_{\min} = 7$

| α | β |
|----------|---------|
| 6 | 0 |
| 5 | 1 |
| 4 | 2 |
| 3 | 3 |

Phân bố Hamming

- A_j là số vector từ mã có $W(v_i) = j$
- $\{A_j\}$ với $j = 1, 2, \dots, n$ là phân bố Hamming của mã khối (n, k)
- Ví dụ: mã $(6, 3)$ có phân bố Hamming như sau: $A_1=0, A_2=0, A_3=4, A_4=3, A_5=0, A_6=0$
- $\{0, 0, 4, 3, 0, 0\}$

Các vec tơ lỗi không phát hiện được
Cho xác suất lỗi bit của kênh là P_B .

| e | $P(e)$ | | |
|--------|---|---------------------------|--|
| 110100 | $P_B \cdot P_B \cdot (1 - P_B) \cdot P_B \cdot (1 - P_B) \cdot (1 - P_B)$ | $P_B^3 \cdot (1 - P_B)^3$ | |
| 011010 | $(1 - P_B) \cdot P_B \cdot P_B \cdot (1 - P_B) \cdot P_B \cdot (1 - P_B)$ | $P_B^3 \cdot (1 - P_B)^3$ | |
| 101110 | $P_B \cdot (1 - P_B) \cdot P_B \cdot P_B \cdot P_B \cdot (1 - P_B)$ | $P_B^4 \cdot (1 - P_B)^2$ | |
| 101001 | | $P_B^3 \cdot (1 - P_B)^3$ | |
| 011101 | | $P_B^4 \cdot (1 - P_B)^2$ | |
| 110011 | | $P_B^4 \cdot (1 - P_B)^2$ | |
| 000111 | | $P_B^3 \cdot (1 - P_B)^3$ | |
| | $P_{nd} = 4 \cdot P_B^3 (1 - P_B)^3 + 3 \cdot P_B^4 (1 - P_B)^2$ | | |

Công thức tính xác suất lỗi không phát hiện được

$$P_{nd} = \sum_{j=1}^n A_j P_B^j (1 - P_B)^{n-j}$$

cuu duong than cong . com

3.2.4. Mã khối tuyến tính hệ thống

- Mã (n, k) được đặc trưng bằng:
 - + Ma trận sinh G
 - + Ma trận kiểm tra H
- Các thông số n, k, R, P_{nd} .

3.2.5. Nguyên lý mã hóa với mã tuyến tính hệ thống

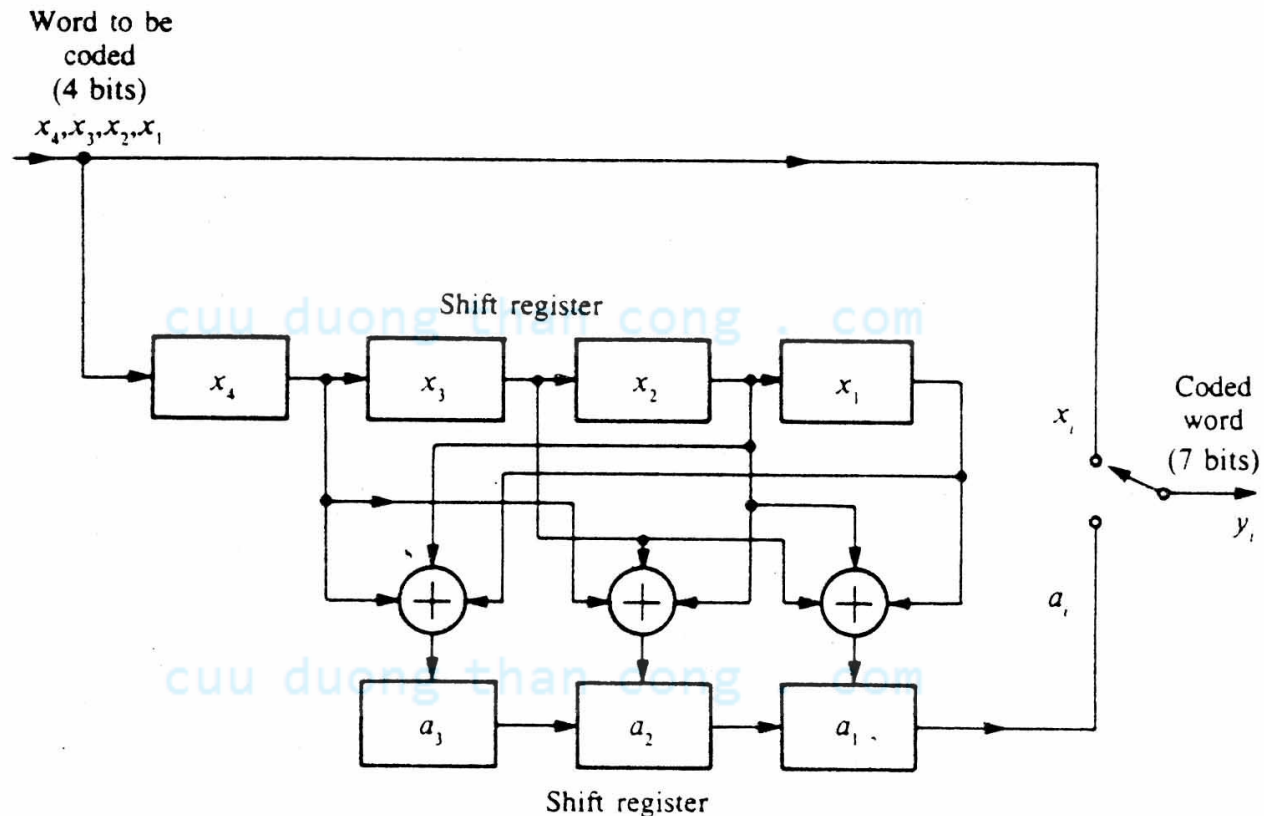


Figure 3.24 Implementation of a coder for a systematic linear code

3.2.6. Nguyên lý phát hiện lỗi với mã khối tuyến tính hệ thống

cuu duong than cong . com

cuu duong than cong . com

3.2.7. Nguyên lý sửa lỗi với mã khối tuyến tính hệ thống

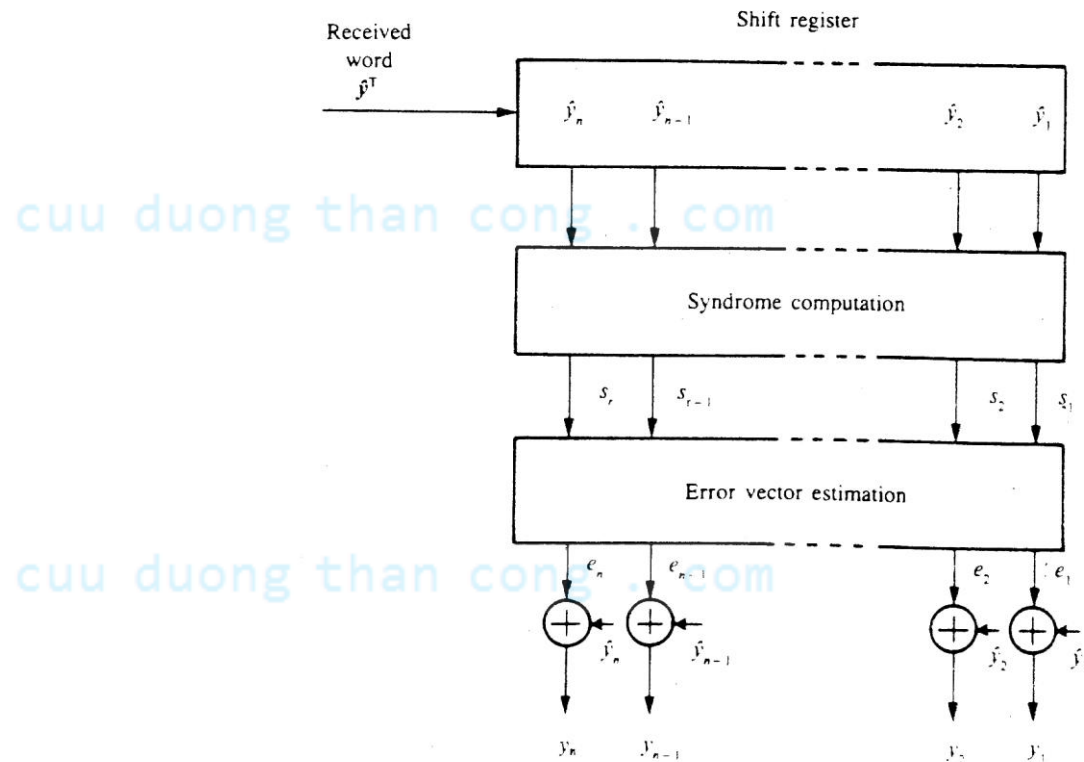


Figure 3.25 Error correction system

- Bảng mã (7,4)

Table 3.4 The words of the (7, 4) code defined by (3.58) and their weights.

| Code words | | | | | | | Weights of code words |
|------------|-------|-------|-------|-------|-------|-------|--------------------------|
| x_1 | x_2 | x_3 | x_4 | a_1 | a_2 | a_3 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 3 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 4 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 3 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 3 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 4 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 3 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 4 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 4 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 3 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 4 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 4 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 3 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 4 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 |

Triệu chứng lỗi

Table 3.5 Syndromes of the error configurations corresponding to a single transmission error for the (7,4) code defined by (3.58).

| Error vectors | | | | | | | Syndromes | | |
|---------------|-------|-------|-------|-------|-------|-------|-----------|-------|-------|
| e_1 | e_2 | e_3 | e_4 | e_5 | e_6 | e_7 | s_1 | s_2 | s_3 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

Sơ đồ sửa lỗi

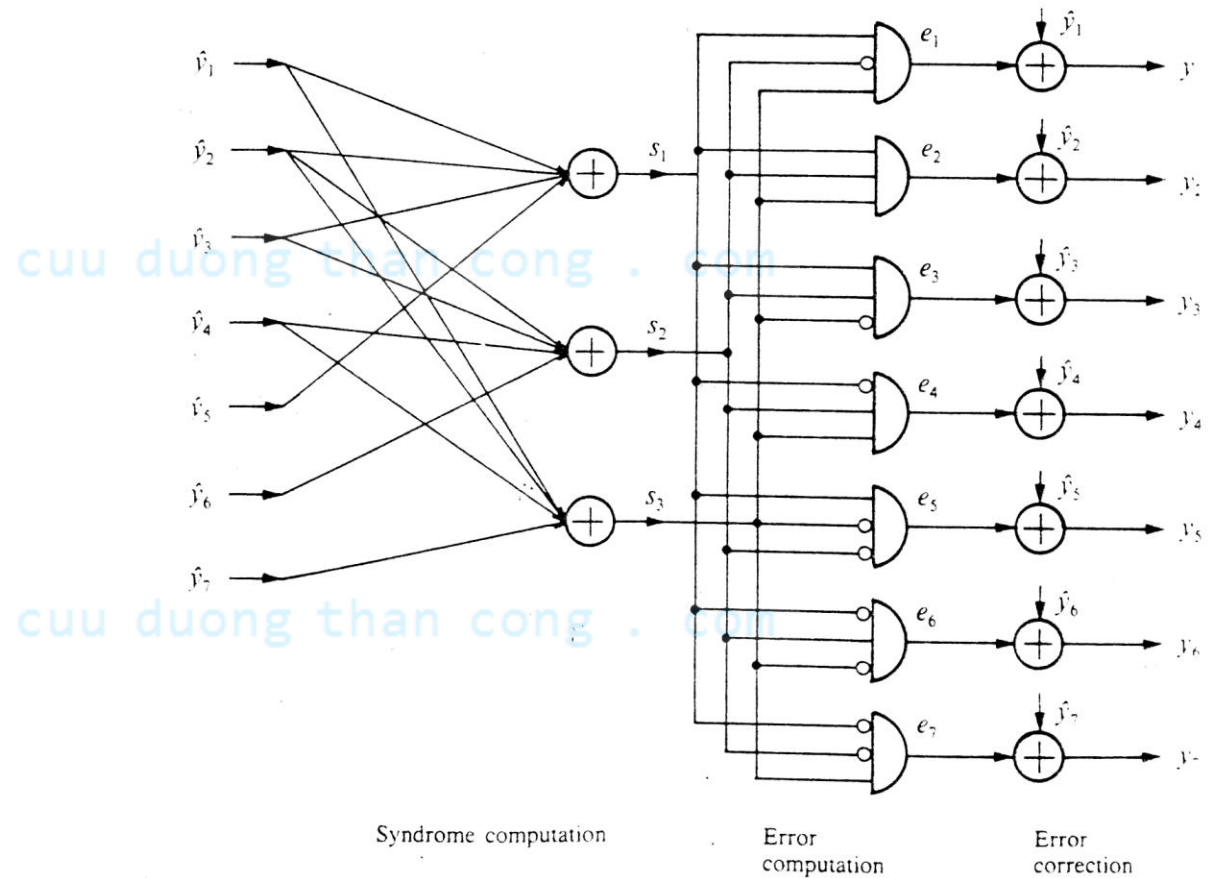


Figure 3.26 Implementation of the single error corrector for the (7,4) code defined by (3.58)

3.2.8. Mã Hamming

- $n = 2^r - 1$
- $m = n - r$
- $d_{\min} = 3$
- $t = 1$

Table 3.6 Number of information and redundancy bits for standard Hamming codes.

| Length of code word (number of bits) (n) | 3 | 7 | 15 | 31 | 63 | 127 |
|--|---|---|----|----|----|-----|
| Number of information bits (m) | 1 | 4 | 11 | 26 | 57 | 120 |
| Number of redundancy bits (r) | 2 | 3 | 4 | 5 | 6 | 7 |

Table 3.7 Computation of parity bits with a Hamming code.

| Code word | y_1 a_0 | y_2 a_1 | y_3 x_1 | y_4 a_2 | y_5 x_2 | y_6 x_3 | y_7 x_4 | y_8 a_3 | y_9 x_5 | y_{10} x_6 | y_{11} x_7 | y_{12} x_8 | y_{13} x_9 | y_{14} x_{10} | y_{15} x_{11} | |
|--------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-------------------|-------------------|-------------------|-------------------|----------------------|----------------------|-------|
| Parity a_0 | x | | x | | x | | x | | x | | x | | x | | x | 2^0 |
| Parity a_1 | | x | x | | | x | x | | | x | x | | | x | x | 2^1 |
| Parity a_2 | | | | x | x | x | x | | | | | x | x | x | x | 2^2 |
| Parity a_3 | | | | | | | | x | x | x | x | x | x | x | x | 2^3 |

Mã (15,11)

- Qui luật tính các bit parity

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 3.27 Generating matrix of the (15, 11) Hamming code

Table 3.8 Locating an error with a Hamming code.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Information word x^T | | | | | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| Transmitted code word y^T | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Error e^T | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| Received word y^T | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Parities calculated for the received word | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| | 1 | 0 | | 1 | | | | 0 | | | | | | | |

5 → The bit in error is in position 5

Nguyên lý sửa lỗi

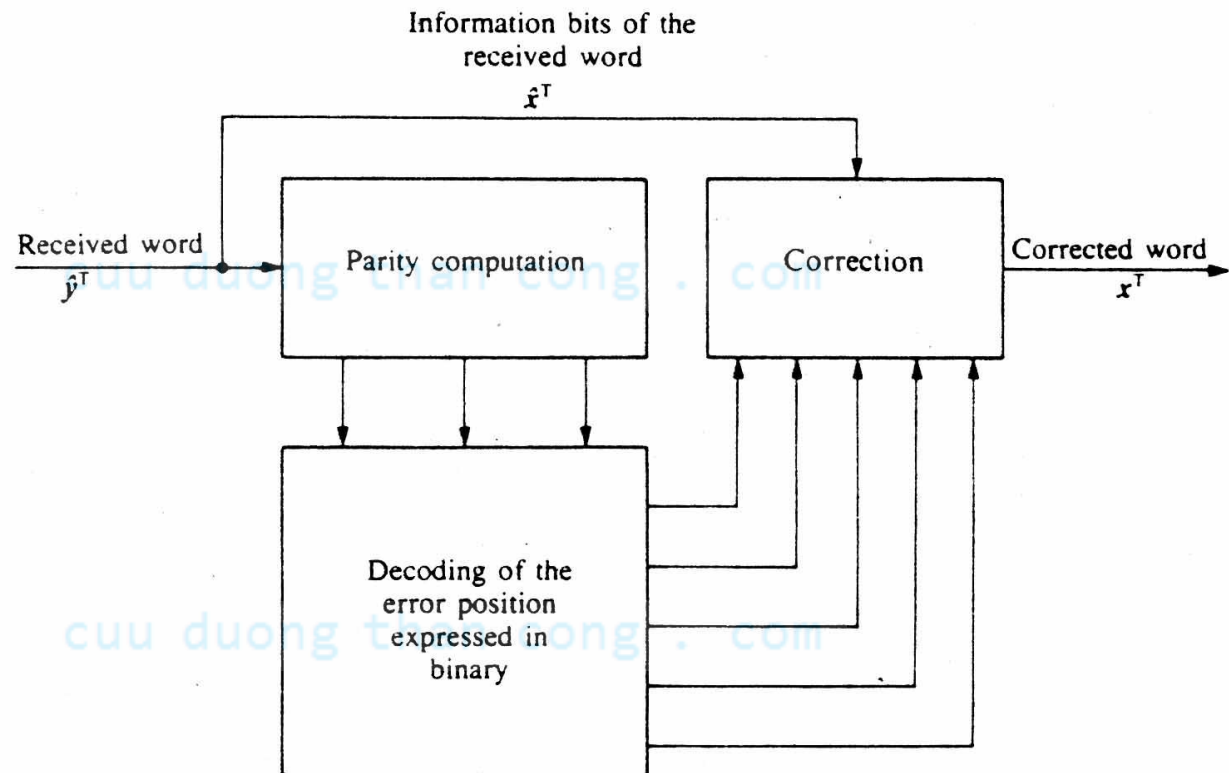


Figure 3.28 The principle of a Hamming code error corrector

Ma trận sinh dạng chuẩn²

$$H = \left[\begin{array}{cccccccccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array} \right]$$

Matrix I_{11} Matrix P

Figure 3.29 Standard form of the generating matrix of the (15, 11) Hamming code

3.3. MÃ ĐA THỨC

- Định nghĩa các mã đa thức
- Mã hóa và giải mã bằng đa thức
- Thực hiện phần mềm mã hóa và giải mã bằng đa thức
- Các tính chất chủ yếu của mã đa thức
- Các mã đa thức chuẩn

3.3.1.Định nghĩa các mã đa thức

cuu duong than cong . com

cuu duong than cong . com

3.3.2. Mã hóa và giải mã bằng đa thức

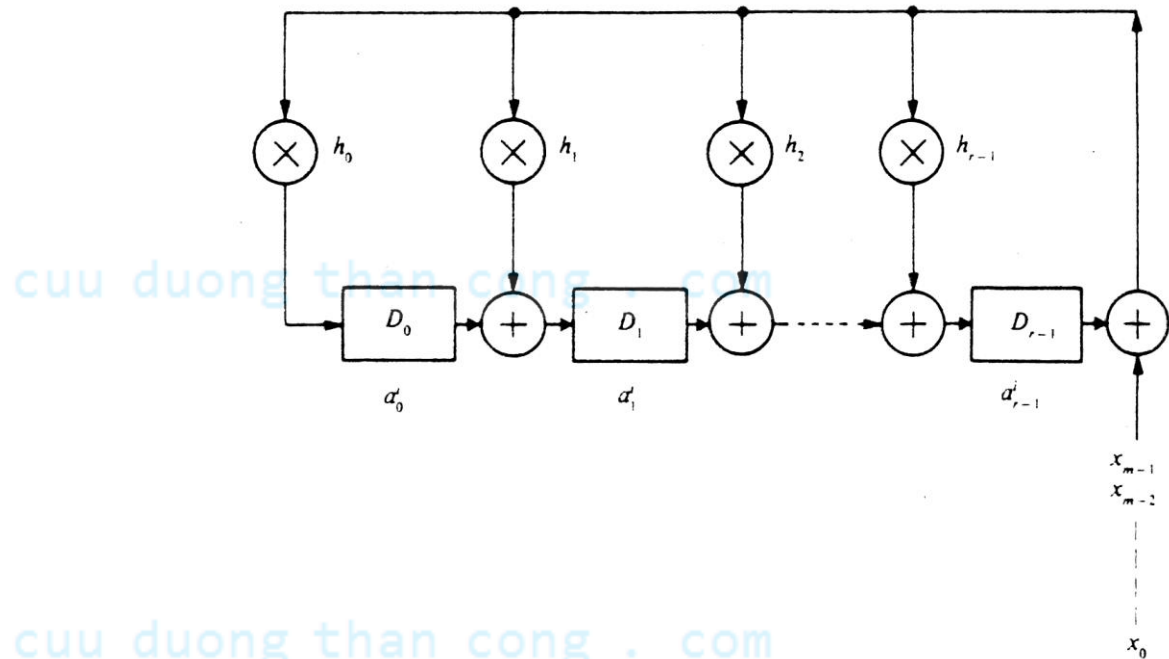


Figure 3.30 Polynomial division circuit

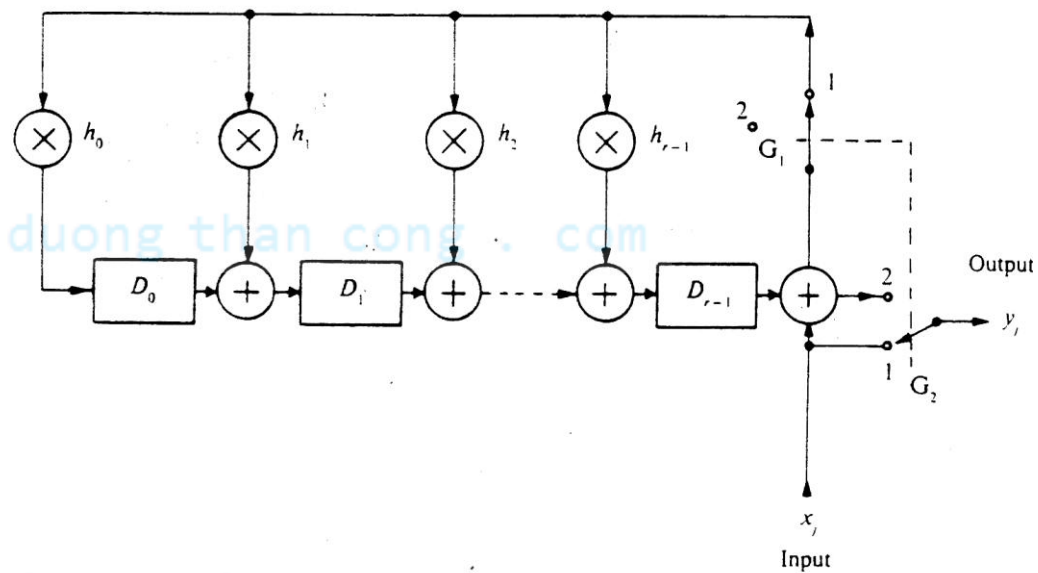


Figure 3.31 Implementation of a polynomial coder

3.3.3. Thực hiện phần mềm mã hóa và giải mã bằng đa thức

Table 3.9 Bits used in the calculation of the state of the polynomial decoding register after eight shifts. CCITT V.41 code.

| | Shift register | | | | | | | | | | | | | | | |
|---|----------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | 1 | | | | z^5 | | | | | | | | z^{12} | | | |
| | D_0 | D_1 | D_2 | D_3 | D_4 | D_5 | D_6 | D_7 | D_8 | D_9 | D_{10} | D_{11} | D_{12} | D_{13} | D_{14} | D_{15} |
| | a_8 | a_9 | a_{10} | a_{11} | a_{12} | a_{13} | a_{14} | a_{15} | a_0 | a_1 | a_2 | a_3 | a_4 | a_5 | a_6 | a_7 |
| Bits used in the calculation of the remainder at each stage | a_{12} | a_{13} | a_{14} | a_{15} | x_{12} | x_{13} | x_{14} | x_{15} | a_{11} | a_{12} | a_{13} | a_{14} | a_{15} | a_9 | a_{10} | a_{11} |
| | x_{12} | x_{13} | x_{14} | x_{15} | | a_8 | a_9 | a_{10} | a_{15} | x_{12} | x_{13} | x_{14} | x_{15} | a_{13} | a_{14} | a_{15} |
| | x_8 | x_9 | x_{10} | x_{11} | | a_{12} | a_{13} | a_{14} | x_{15} | | | | | a_8 | x_{13} | x_{14} |
| | | | | | | x_{12} | x_{13} | x_{14} | x_{11} | | | | | a_{12} | x_9 | x_{10} |
| | | | | | | x_8 | x_9 | x_{10} | | | | | | x_{12} | | |
| | | | | | | | | | | | | | | x_8 | | |

Table 3.10 Bits used after simplification during the calculation of the state of the polynomial decoding register after eight shifts, CCITT V.41 code.

| | Shift register | | | | | | | | | | | | | | | |
|--|-------------------|-------------------|----------------------|----------------------|----------|----------------------|----------------------|----------------------|-------------------|-------------------|-------------------|-------------------|-------------------|----------------|----------------|-------------------------------|
| | 1 | | | | z^5 | | | | z^{12} | | | | | | | |
| | D_0 | D_1 | D_2 | D_3 | D_4 | D_5 | D_6 | D_7 | D_8 | D_9 | D_{10} | D_{11} | D_{12} | D_{13} | D_{14} | D_{15} |
| Bits used in the calculation of the remainder at each stage | u_8 u_{12} | u_9 u_{13} | u_{10} u_{14} | u_{11} u_{15} | u_{12} | u_{13} u_{12} | u_{14} u_{13} | u_{15} u_{14} | a_0 u_{11} | a_1 u_{12} | a_2 u_{13} | a_3 u_{14} | a_4 u_{15} | a_5 u_8 | a_6 u_9 | a_7 u_{10} u_{11} |

Table 3.11 Example of eight successive shifts with the CCITT V.41 polynomial coder.

| Shift register | | | | | | | | | | | | | | | | | | |
|----------------|---------|---------|---------|---------|---------|---------|---------|---------|----------|---------|------------|------------|------------|------------|------------|------------|-------|----------|
| | 1 | | | z^5 | | | | | z^{12} | | | | | | | | | |
| | a_0 | a_1 | a_2 | a_3 | a_4 | a_5 | a_6 | a_7 | a_8 | a_9 | a_{10} | a_{11} | a_{12} | a_{13} | a_{14} | a_{15} | Input | |
| Initial state | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | x_{15} |
| Shift 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | x_{14} |
| Shift 2 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | x_{13} |
| Shift 3 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | x_{12} |
| Shift 4 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | x_{11} |
| Shift 5 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | x_{10} |
| Shift 6 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | x_9 |
| Shift 7 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | x_8 |
| Shift 8 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | | |
| Final state | a_0^1 | a_1^1 | a_2^1 | a_3^1 | a_4^1 | a_5^1 | a_6^1 | a_7^1 | a_8^1 | a_9^1 | a_{10}^1 | a_{11}^1 | a_{12}^1 | a_{13}^1 | a_{14}^1 | a_{15}^1 | | |

| | | | | | | | | | |
|-------|---|---|----|----|----|----|----|----|----------|
| i | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | |
| a_i | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | |
| x_i | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | \oplus |
| <hr/> | | | | | | | | | |
| u_i | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | |

Figure 3.32 Computation of the u_i

| | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|---|----------|
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| 89_H | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | |
| a_i | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | \oplus |
| <hr/> | | | | | | | | | |
| a_{i+8}^2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | |

Figure 3.33 Computation of the high byte of the result after 8 shifts

3.3.4. Các tính chất chủ yếu của mã đa thức

cuu duong than cong . com

cuu duong than cong . com

3.3.5. Các mã đa thức chuẩn²

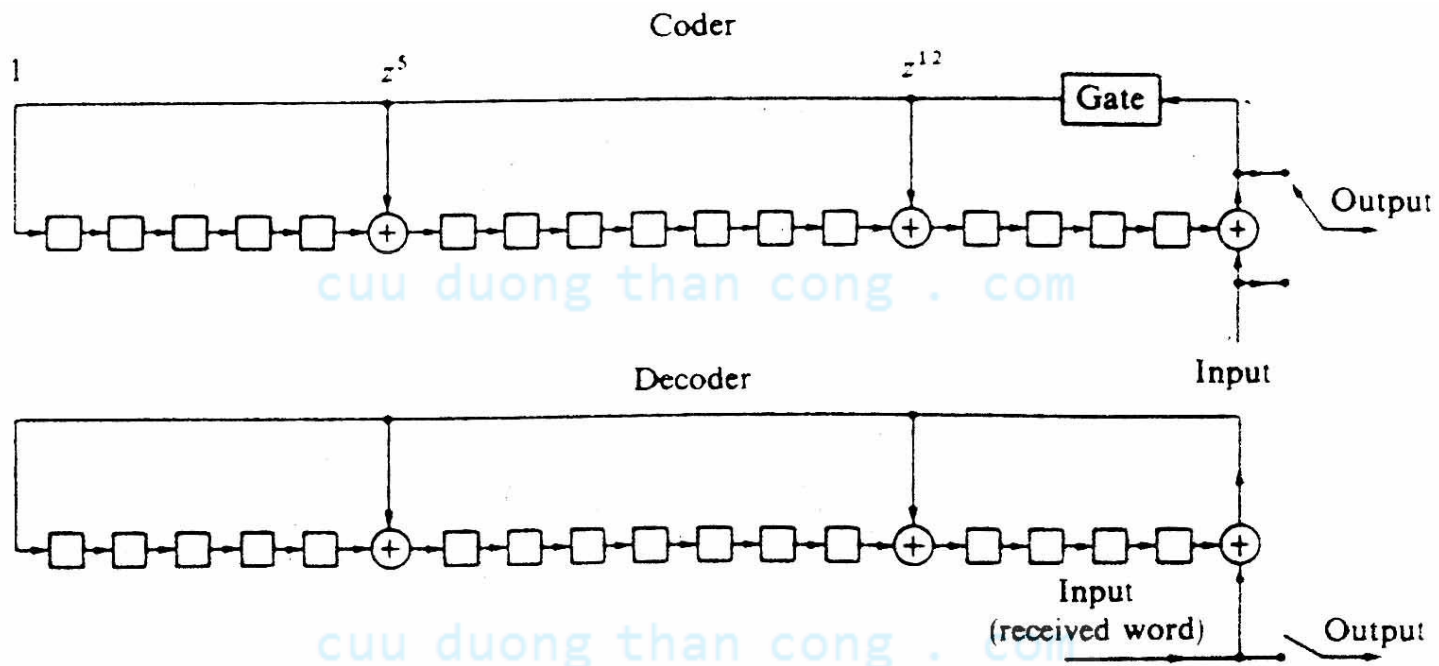


Figure 3.34 Coder and decoder for the CCITT V.41 code

cuu duong than cong . com

cuu duong than cong . com

3.4. MÃ VÒNG

- Các tính chất chính của mã vòng
- Giải mã sửa lỗi Meggitt
- Mã BCH
- Mã Golay
- Các mã lửa
- Mã Abramson
- Mã Reed-Solomon

3.4.1. Các tính chất chính của mã vòng

Table 3.12 List of primitive polynomials.

| Degree of polynomial r | Primitive polynomials |
|-----------------------------|-----------------------------|
| 3 | $1 + z + z^3$ |
| 4 | $1 + z + z^4$ |
| 5 | $1 + z^2 + z^5$ |
| 6 | $1 + z + z^6$ |
| 7 | $1 + z^3 + z^7$ |
| 8 | $1 + z^2 + z^3 + z^4 + z^8$ |
| 9 | $1 + z^4 + z^9$ |
| 10 | $1 + z^3 + z^{10}$ |

3.4.2. Giải mã sửa lỗi Meggitt

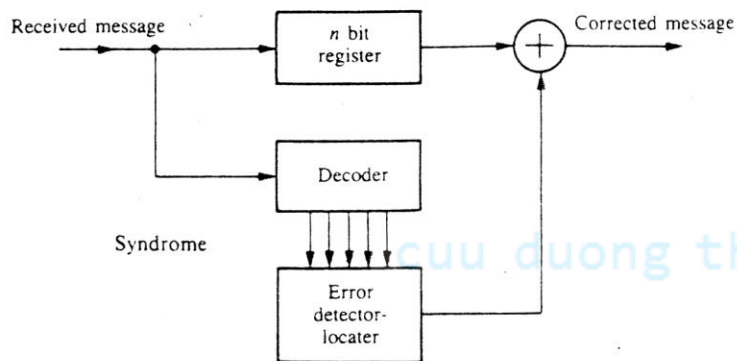


Figure 3.35 General error correction system for cyclic codes

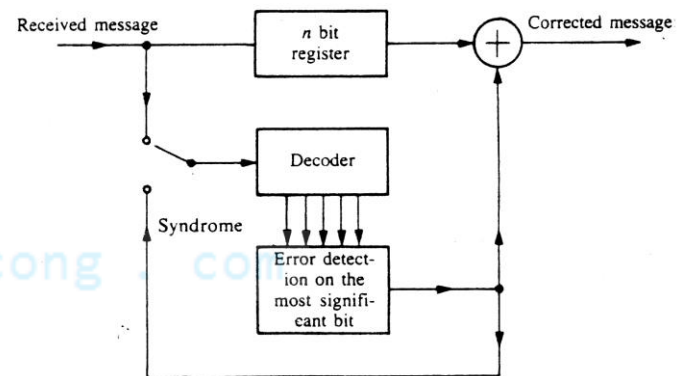


Figure 3.36 Meggitt decoder-corrector

3.4.3. Mã BCH

cuu duong than cong . com

cuu duong than cong . com

3.4.4. Mã Golay

cuu duong than cong . com

cuu duong than cong . com

3.4.5. Các mã lửa

cuu duong than cong . com

cuu duong than cong . com

3.4.6. Mã Abramson

cuu duong than cong . com

cuu duong than cong . com

3.4.7. Mã Reed-Solomon

cuu duong than cong . com

cuu duong than cong . com

cuu duong than cong . com

cuu duong than cong . com

3.5. MÃ CHẬP

- Nguyên lý mã hóa chập
- Biểu diễn ma trận của mã chập
- Nguyên lý phát hiện và sửa lỗi bằng mã chập
- Thuật toán Viterbi
- Giải mã tuần tự

3.5.1. Nguyên lý mã hóa chập

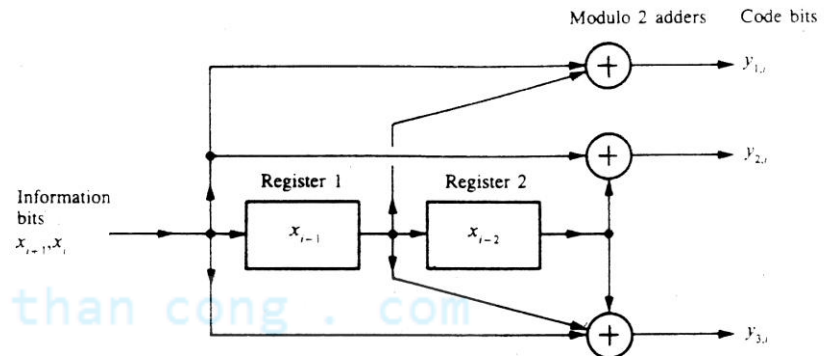


Figure 3.37 (3,1) Convolutional coder

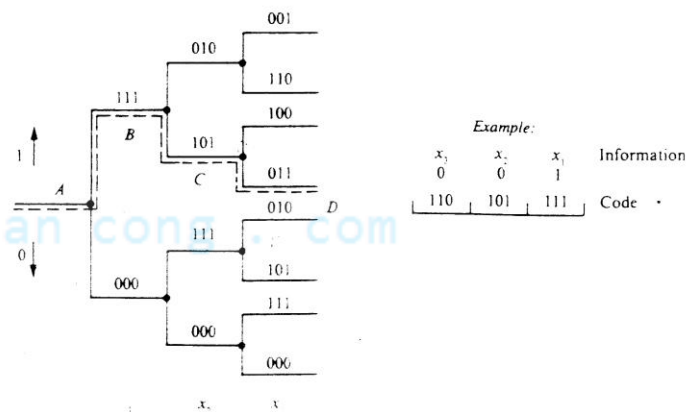


Figure 3.38 Tree diagram for the convolutional coder of Figure 3.37

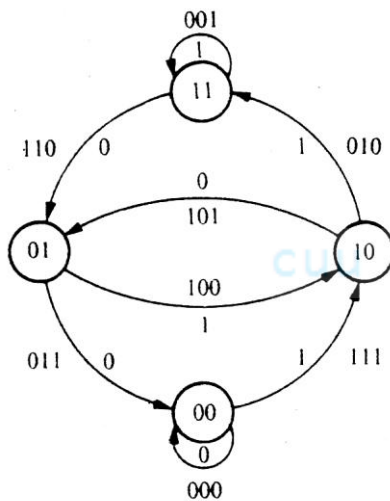


Figure 3.39 State diagram of the coder of Figure 3.37

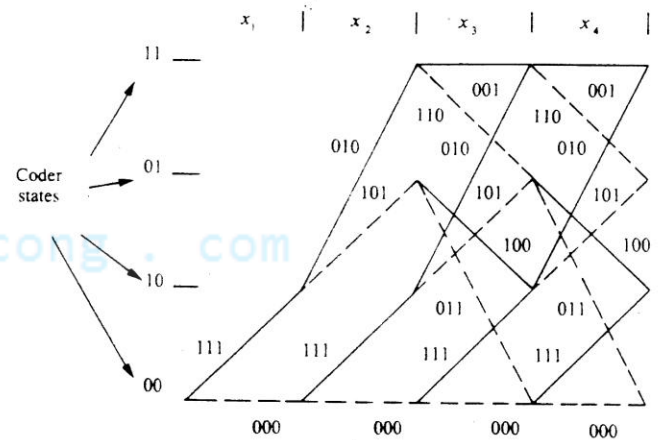


Figure 3.40 Lattice diagram of the coder of Figure 3.37

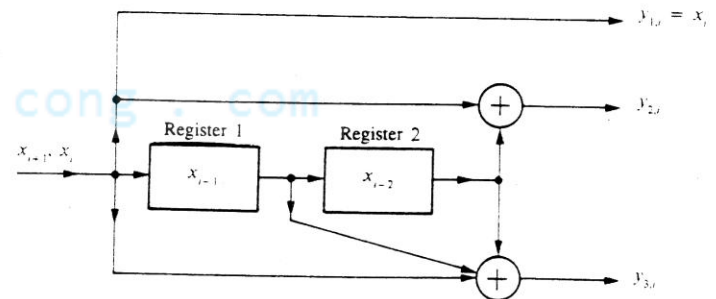


Figure 3.41 Convolutional coder for a (3,1) separable code

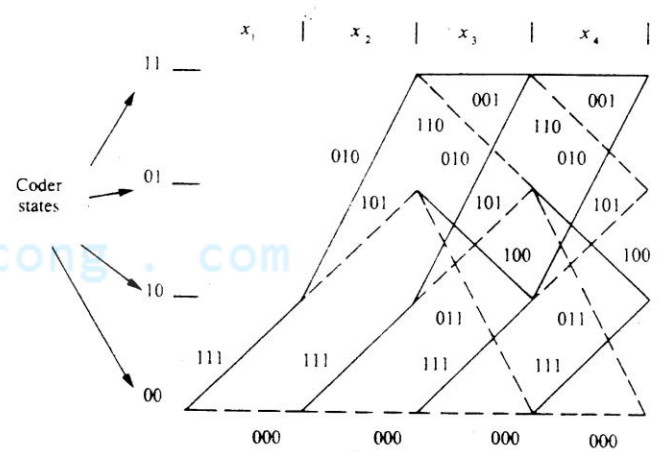


Figure 3.40 Lattice diagram of the coder of Figure 3.37

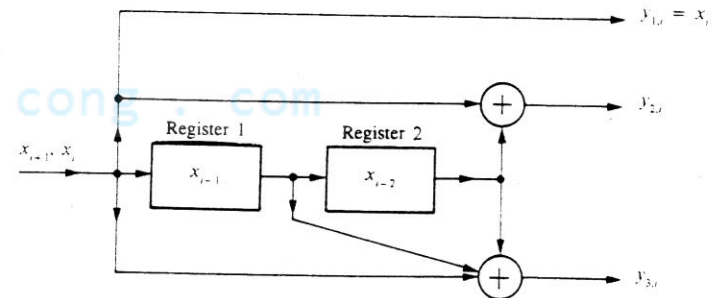


Figure 3.41 Convolutional coder for a (3, 1) separable code

3.5.2. Biểu diễn ma trận của mã chập

cuu duong than cong . com

cuu duong than cong . com

3.5.3. Nguyên lý phát hiện và sửa lỗi bằng mã chập

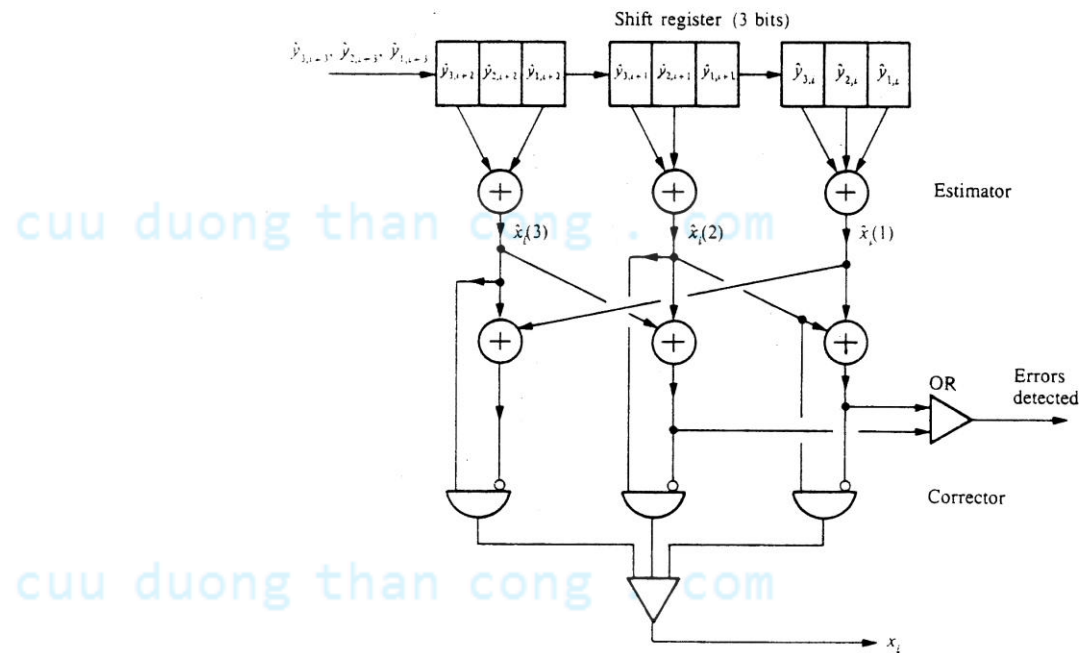


Figure 3.42 Detection and correction circuit for the code corresponding to the coder of Figure 3.37

3.5.4. Thuật toán Viterbi

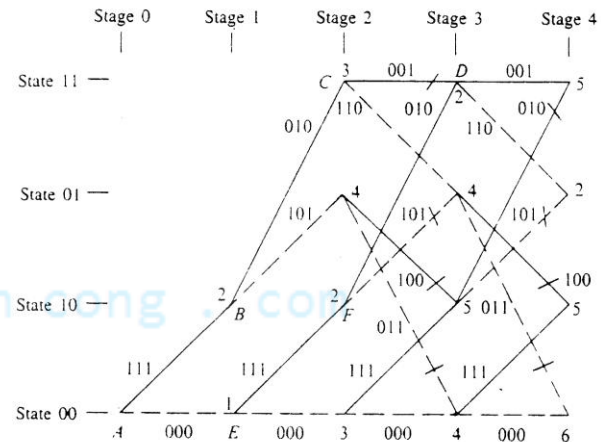


Figure 3.43 Decoding using Viterbi's algorithm. Eliminated paths are crossed through

3.5.5. Giải mã tuần tự

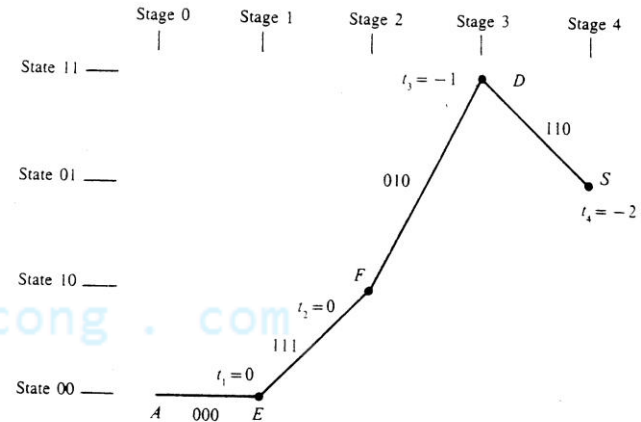


Figure 3.44 Sequential decoding

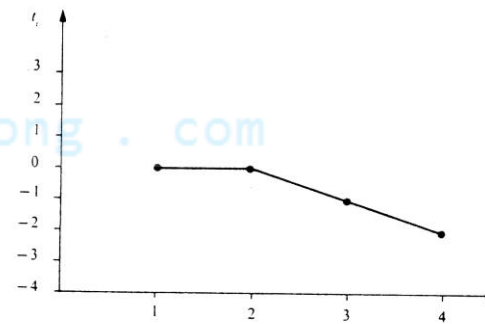


Figure 3.45 Weighted decoding distance

Mã khối tuyến tính

- Việc mã hoá cho một tổ hợp k -bit (vec tơ mang tin) là độc lập với các tổ hợp trước và sau nó. [cuu duong than cong . com](http://cuuduongthancong.com)
- Sử dụng cho mô hình kênh không nhớ (memoryless channel)

[cuu duong than cong . com](http://cuuduongthancong.com)

Không gian vector trên trường số nhị phân

- Xét các bộ n-bit (n-tuple)
 - Có 2^n bộ khác nhau
- Các toán tử Cộng và Nhân modulo 2 được định nghĩa theo bảng như sau:

| A | B | $A \odot B$ | $A \oplus B$ | $A \odot B \oplus B$ | $A \odot B \oplus A$ |
|---|---|-------------|--------------|----------------------|----------------------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

Trên trường số nhị phân không có toán tử trừ, hay nói cách khác: toán tử cộng và trừ là một .

Định nghĩa : Không gian vector V_n bao gồm 2^n vector có kích thước n-bit và hai toán tử **Cộng** và **Nhân** modulo 2

Không gian con S của V_n

- Không gian con S của V_n :

- Chứa vector không $v_0 \in S$

- Khép kín $v_i, v_j \in S \Rightarrow v_i \oplus v_j \in S$

Thí dụ V_4 :

| | | | |
|------|------|------|------|
| 0000 | 0001 | 0010 | 0011 |
| 0100 | 0101 | 0110 | 0111 |
| 1000 | 1001 | 1010 | 1011 |
| 1100 | 1101 | 1110 | 1111 |

| v_i | S1 | Hà | Tình | Đông |
|-------------------|------|------|------|------|
| v_0 | 0000 | 0000 | 0000 | 0000 |
| v_1 | 0100 | 0001 | 0010 | 0101 |
| v_2 | 1000 | 0101 | 0100 | 1010 |
| $v_3 = v_1 + v_2$ | 1100 | 0100 | 0110 | 1111 |

cuu duong than cong . com

cuu duong than cong . com

Mã khối tuyến tính

- Định nghĩa : mã khối tuyến tính là không gian con của không gian V_n . Được đặc trưng bởi (n,k) với n là kích thước vector từ mã và k là kích thước vector mang tin, tỷ số mã $R = k/n$.

cuu duong than cong . com

Phát hiện và sửa lỗi

- Ma trận kiểm tra
- Bảng triệu chứng (syndrome)

[cuu duong than cong . com](http://cuuduongthancong.com)

[cuu duong than cong . com](http://cuuduongthancong.com)

Khả năng phát hiện và sửa lỗi của mã khối tuyến tính

- Hàm trọng Hamming
- Khoảng cách Hamming
- Khả năng phát hiện lỗi tối đa
- Khả năng sửa lỗi
- Khả năng phát hiện và sửa lỗi đồng thời
- Phân bố Hamming
- Xác suất lỗi không phát hiện được

Hàm trọng Hamming

- Hàm trọng Hamming $W(V_i)$ của vector từ mã V_i là số digit khác không của V_i .
- Thí dụ:

| | m_i | V_i | $W(V_i)$ |
|-------|-------|---------|----------|
| V_0 | 000 | 000 000 | |
| V_1 | 100 | 110 100 | 3 |
| V_2 | 010 | 011 010 | 3 |
| V_3 | 110 | 101 110 | 4 |
| V_4 | 001 | 101 001 | 3 |
| V_5 | 101 | 011 101 | 4 |
| V_6 | 011 | 110 011 | 4 |
| V_7 | 111 | 000 111 | 3 |

Tính khoảng cách hamming của 2 vector

- $d(v1, v2) = d(110100, 011010) = 4 = W(v1+v2) = W(v3)$
- $d(v4, v6) = d(101001, 110011) = 3 = W(v4+v6) = W(v2)$

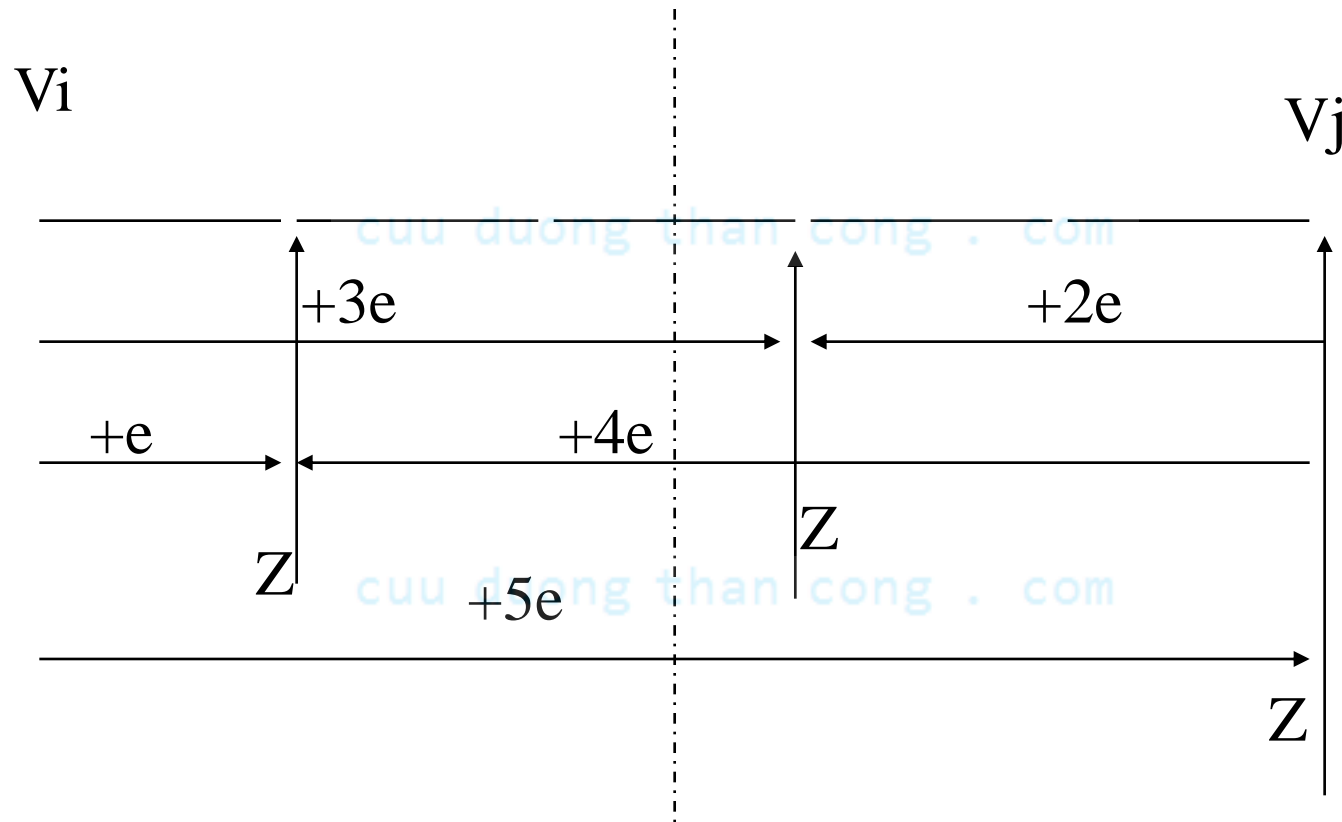
cuu duong than cong . com

Khoảng cách Hamming

- Khoảng cách Hamming giữa 2 vector từ mã V_i và V_j , ký hiệu $d(V_i, V_j)$ bằng số digit khác nhau giữa V_i và V_j .
- Nhận xét:
 - $d(V_i, V_j) = W(V_i + V_j)$.
 - $d_{\min} = W_{\min}$.

$$\begin{aligned}
 d_{\min} &= \min_{d(V_i, V_j); V_i, V_j \in (n, k)} \\
 &= \min_{W(V_i); V_i \in (n, k)} W_{\min}
 \end{aligned}$$

Nguyên lý Maximum Likelihood



3.2.3. Khả năng phát hiện và sửa lỗi của mã khối. Khoảng cách Hamming

- Khả năng phát hiện lỗi tối đa: $e = d_{\min} - 1$
- Khả năng sửa lỗi tối đa: $t = \text{int}[(d_{\min} - 1)/2]$
- Khả năng phát hiện α và sửa lỗi β đồng thời phải thỏa mã điều kiện: $\alpha + \beta + 1 \leq d_{\min}$.

Với $d_{\min} = 7$

| α | β |
|----------|---------|
| 6 | 0 |
| 5 | 1 |
| 4 | 2 |
| 3 | 3 |

Phân bố Hamming

- A_j là số vector từ mã có $W(v_i) = j$
- $\{A_j\}$ với $j = 1, 2, \dots, n$ là phân bố Hamming của mã khối (n, k)
- Ví dụ: mã $(6, 3)$ có phân bố Hamming như sau: $\{0, 0, 4, 3, 0, 0\}$

Các vec tơ lỗi không phát hiện được
Cho xác suất lỗi bit của kênh là P_B .

| e | $P(e)$ | | |
|--------|---|---------------------------|--|
| 110100 | $P_B \cdot P_B \cdot (1 - P_B) \cdot P_B \cdot (1 - P_B) \cdot (1 - P_B)$ | $P_B^3 \cdot (1 - P_B)^3$ | |
| 011010 | $(1 - P_B) \cdot P_B \cdot P_B \cdot (1 - P_B) \cdot P_B \cdot (1 - P_B)$ | $P_B^3 \cdot (1 - P_B)^3$ | |
| 101110 | $P_B \cdot (1 - P_B) \cdot P_B \cdot P_B \cdot P_B \cdot (1 - P_B)$ | $P_B^4 \cdot (1 - P_B)^2$ | |
| 101001 | | $P_B^3 \cdot (1 - P_B)^3$ | |
| 011101 | | $P_B^4 \cdot (1 - P_B)^2$ | |
| 110011 | | $P_B^4 \cdot (1 - P_B)^2$ | |
| 000111 | | $P_B^3 \cdot (1 - P_B)^3$ | |
| | $P_{nd} = 4 \cdot P_B^3 (1 - P_B)^3 + 3 \cdot P_B^4 (1 - P_B)^2$ | | |

Công thức tính xác suất lỗi không phát hiện được

$$P_{nd} = \sum_{j=1}^n A_j P_B^j (1 - P_B)^{n-j}$$

$$P_B \leq 10^{-3}$$

ma (6,3)

$$P_{nd} = \sum_{j=1}^6 A_j P_B^j (1 - P_B)^{n-j} = 4 P_B^3 (1 - P_B)^3 + 3 P_B^4 (1 - P_B)^2$$

$$\approx 4 \cdot 10^{-9} + 3 \cdot 10^{-12} = 4,003 \cdot 10^{-9} \approx 4 \cdot 10^{-9}$$

3.2.4. Mã khối tuyến tính hệ thống

- Mã (n, k) được đặc trưng bằng:
 - + Ma trận sinh G
 - + Ma trận kiểm tra H
- Các thông số n, k, R, P_{nd} .

3.2.5. Nguyên lý mã hóa với mã tuyến tính hệ thống

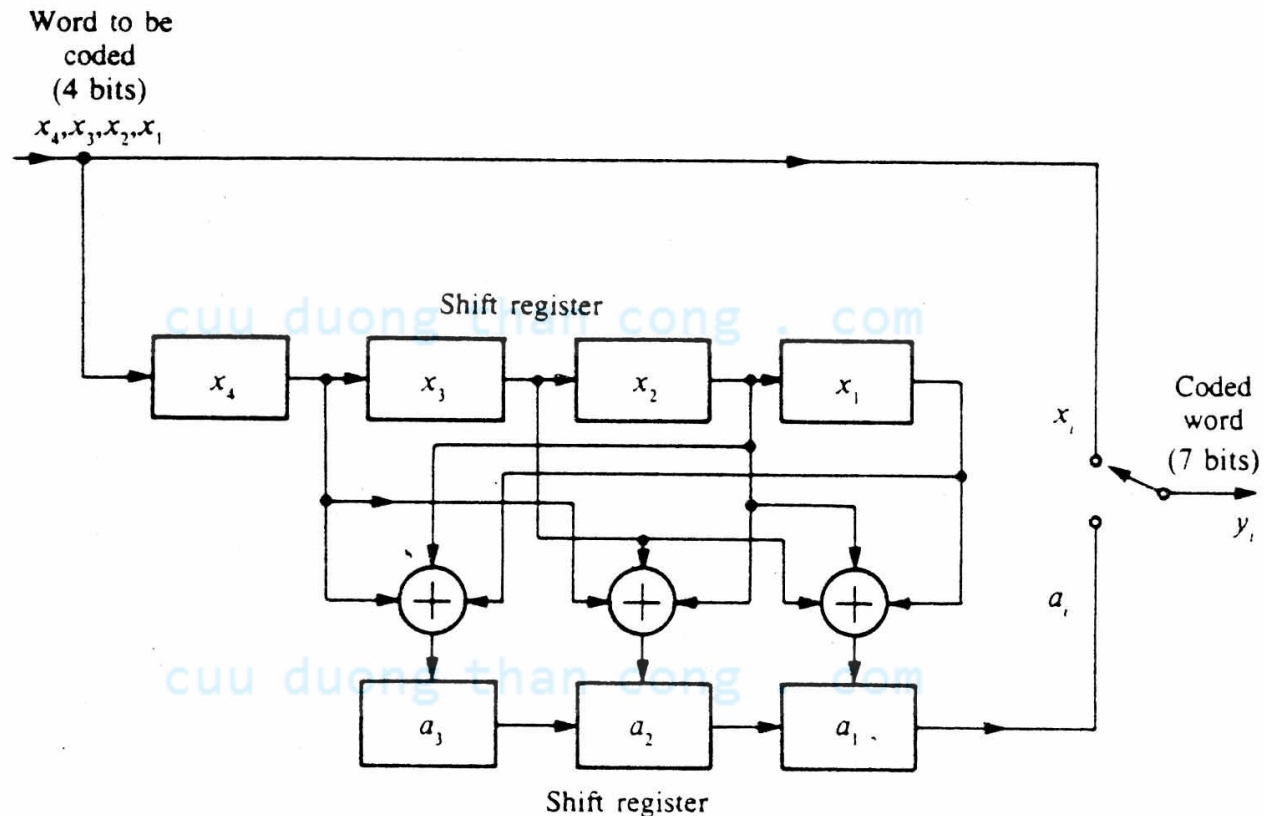


Figure 3.24 Implementation of a coder for a systematic linear code

3.2.6. Nguyên lý phát hiện lỗi với mã khối tuyến tính hệ thống

cuu duong than cong . com

cuu duong than cong . com

3.2.7. Nguyên lý sửa lỗi với mã khối tuyến tính hệ thống

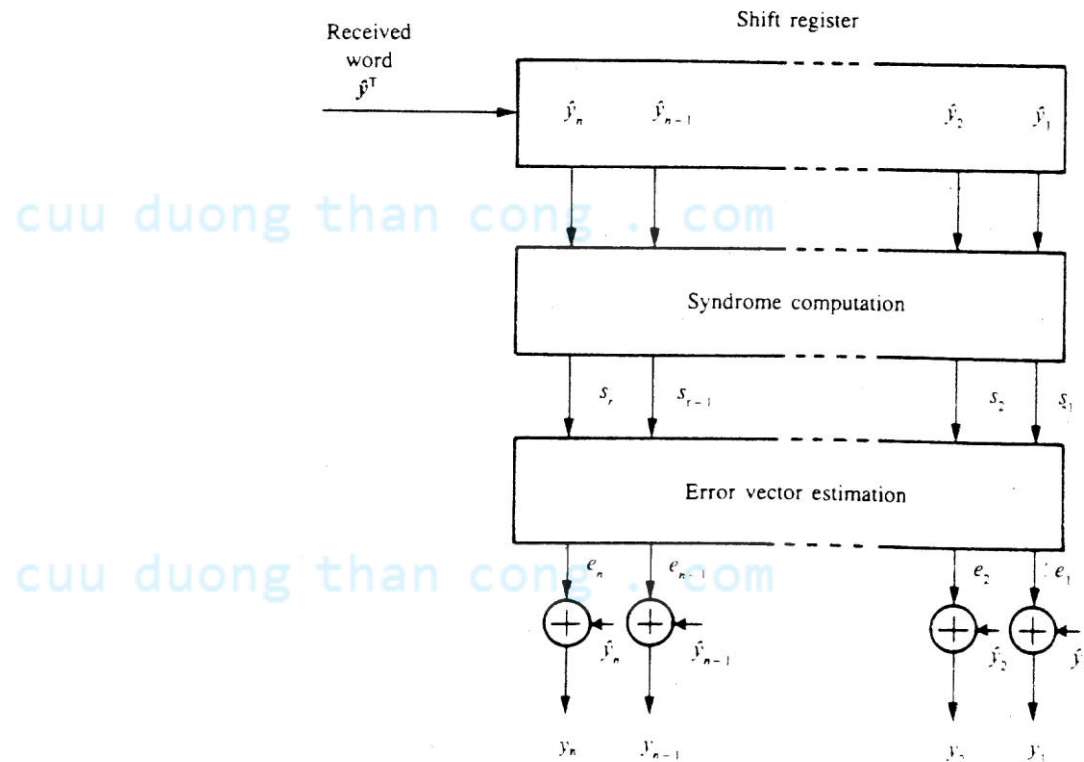


Figure 3.25 Error correction system

- Bảng mã (7,4)

Table 3.4 The words of the (7, 4) code defined by (3.58) and their weights.

| Code words | | | | | | | Weights of code words |
|------------|-------|-------|-------|-------|-------|-------|--------------------------|
| x_1 | x_2 | x_3 | x_4 | a_1 | a_2 | a_3 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 3 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 4 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 3 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 3 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 4 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 3 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 4 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 4 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 3 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 4 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 4 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 3 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 4 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 |

Triệu chứng lỗi

Table 3.5 Syndromes of the error configurations corresponding to a single transmission error for the (7,4) code defined by (3.58).

| Error vectors | | | | | | | Syndromes | | |
|---------------|-------|-------|-------|-------|-------|-------|-----------|-------|-------|
| e_1 | e_2 | e_3 | e_4 | e_5 | e_6 | e_7 | s_1 | s_2 | s_3 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

Sơ đồ sửa lỗi

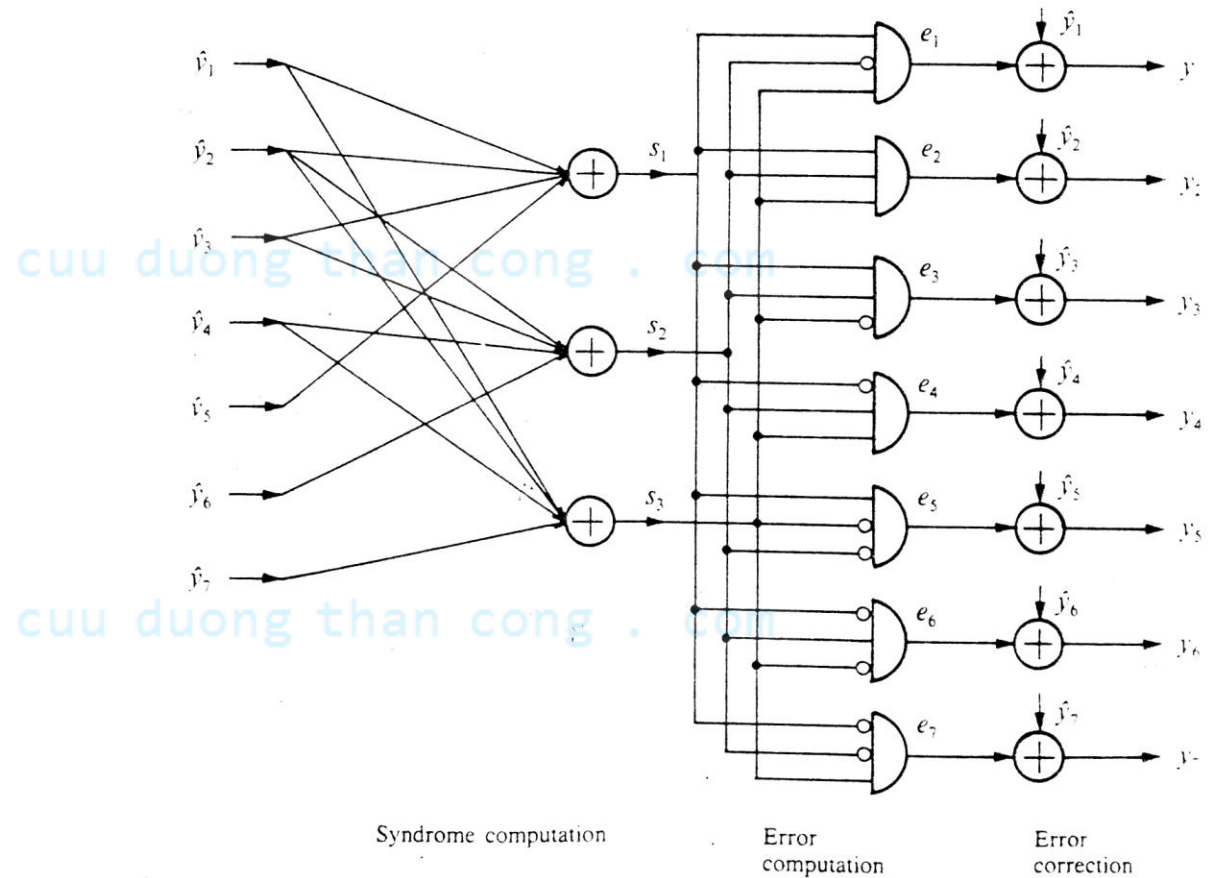


Figure 3.26 Implementation of the single error corrector for the (7,4) code defined by (3.58)

3.2.8. Mã Hamming

- $n = 2^r - 1$
- $m = n - r$
- $d_{\min} = 3$
- $t = 1$

Table 3.6 Number of information and redundancy bits for standard Hamming codes.

| Length of code word (number of bits) (n) | 3 | 7 | 15 | 31 | 63 | 127 |
|--|---|---|----|----|----|-----|
| Number of information bits (m) | 1 | 4 | 11 | 26 | 57 | 120 |
| Number of redundancy bits (r) | 2 | 3 | 4 | 5 | 6 | 7 |

Table 3.7 Computation of parity bits with a Hamming code.

| Code word | y_1 a_0 | y_2 a_1 | y_3 x_1 | y_4 a_2 | y_5 x_2 | y_6 x_3 | y_7 x_4 | y_8 a_3 | y_9 x_5 | y_{10} x_6 | y_{11} x_7 | y_{12} x_8 | y_{13} x_9 | y_{14} x_{10} | y_{15} x_{11} | |
|--------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-------------------|-------------------|-------------------|-------------------|----------------------|----------------------|-------|
| Parity a_0 | x | | x | | x | | x | | x | | x | | x | | x | 2^0 |
| Parity a_1 | | x | x | | | x | x | | | x | x | | | x | x | 2^1 |
| Parity a_2 | | | | x | x | x | x | | | | | x | x | x | x | 2^2 |
| Parity a_3 | | | | | | | | x | x | x | x | x | x | x | x | 2^3 |

Mã (15,11)

- Qui luật tính các bit parity

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 3.27 Generating matrix of the (15, 11) Hamming code

Table 3.8 Locating an error with a Hamming code.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Information word x^T | | | | | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| Transmitted code word y^T | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Error e^T | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| Received word y^T | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Parities calculated for the received word | 1 | 0 | | 1 | | | | 0 | | | | | | | |

5 → The bit in error is in position 5

Nguyên lý sửa lỗi

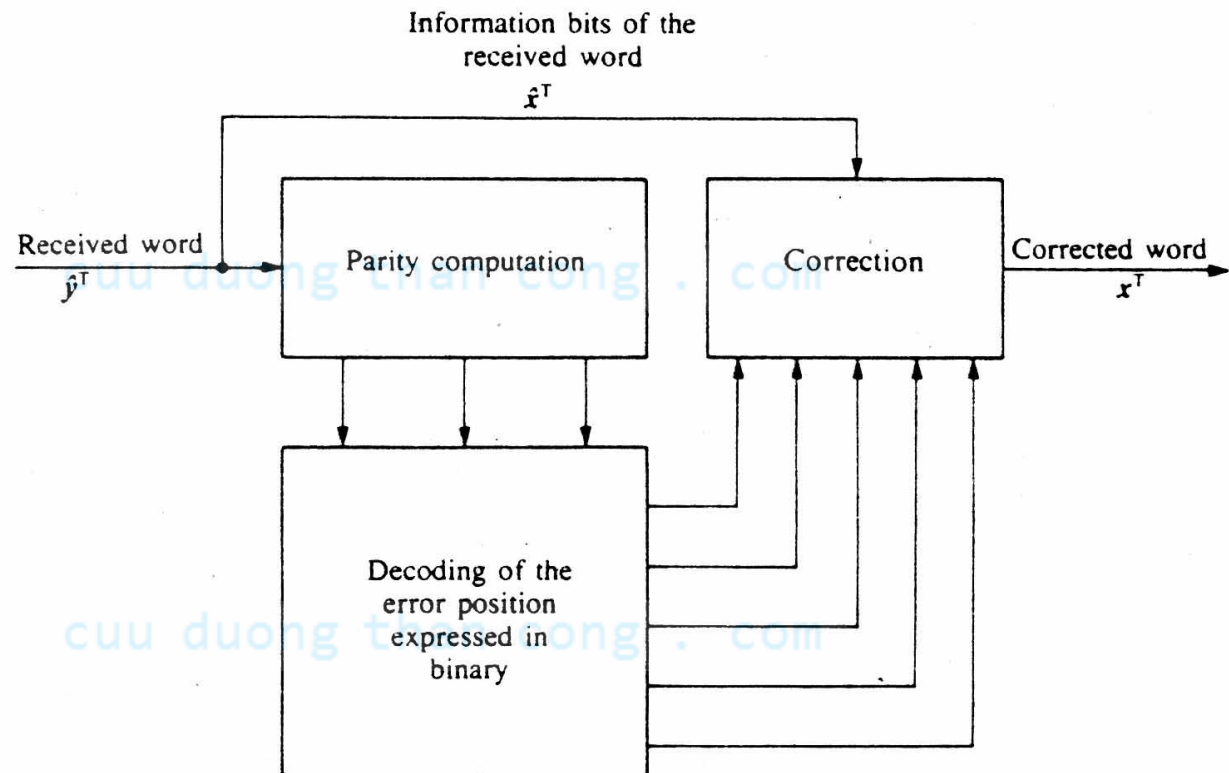


Figure 3.28 The principle of a Hamming code error corrector

Ma trận sinh dạng chuẩn²

$$H = \left[\begin{array}{cccccccccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array} \right]$$

Matrix I_{11} Matrix P

Figure 3.29 Standard form of the generating matrix of the (15, 11) Hamming code

Không gian vector con trên trường số nhị phân

cuu duong than cong . com

cuu duong than cong . com

Thí dụ mã (4,2)

| m_i | V_i |
|-------|-------|
| 00 | 0000 |
| 10 | 1000 |
| 01 | 0101 |
| 11 | 1101 |

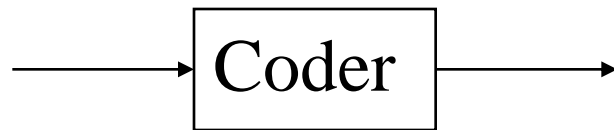
$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

| m_i | V_i |
|-------|-------|
| 00 | 0000 |
| 10 | 1010 |
| 01 | 0101 |
| 11 | 1111 |

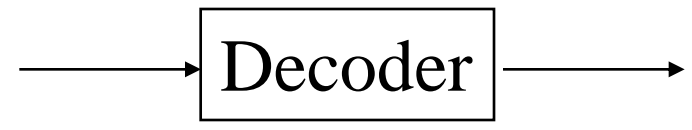
Thí dụ mã hóa bằng mã(4,2)

10 11 01 00



| m_i | V_i |
|-------|-------|
| 00 | 0000 |
| 10 | 1000 |
| 01 | 0101 |
| 11 | 1101 |

1000 1101 0101 0000



| m_i | V_i |
|-------|-------|
| 00 | 0000 |
| 10 | 1000 |
| 01 | 0101 |
| 11 | 1101 |

10110100

Thí dụ mã (6,3)

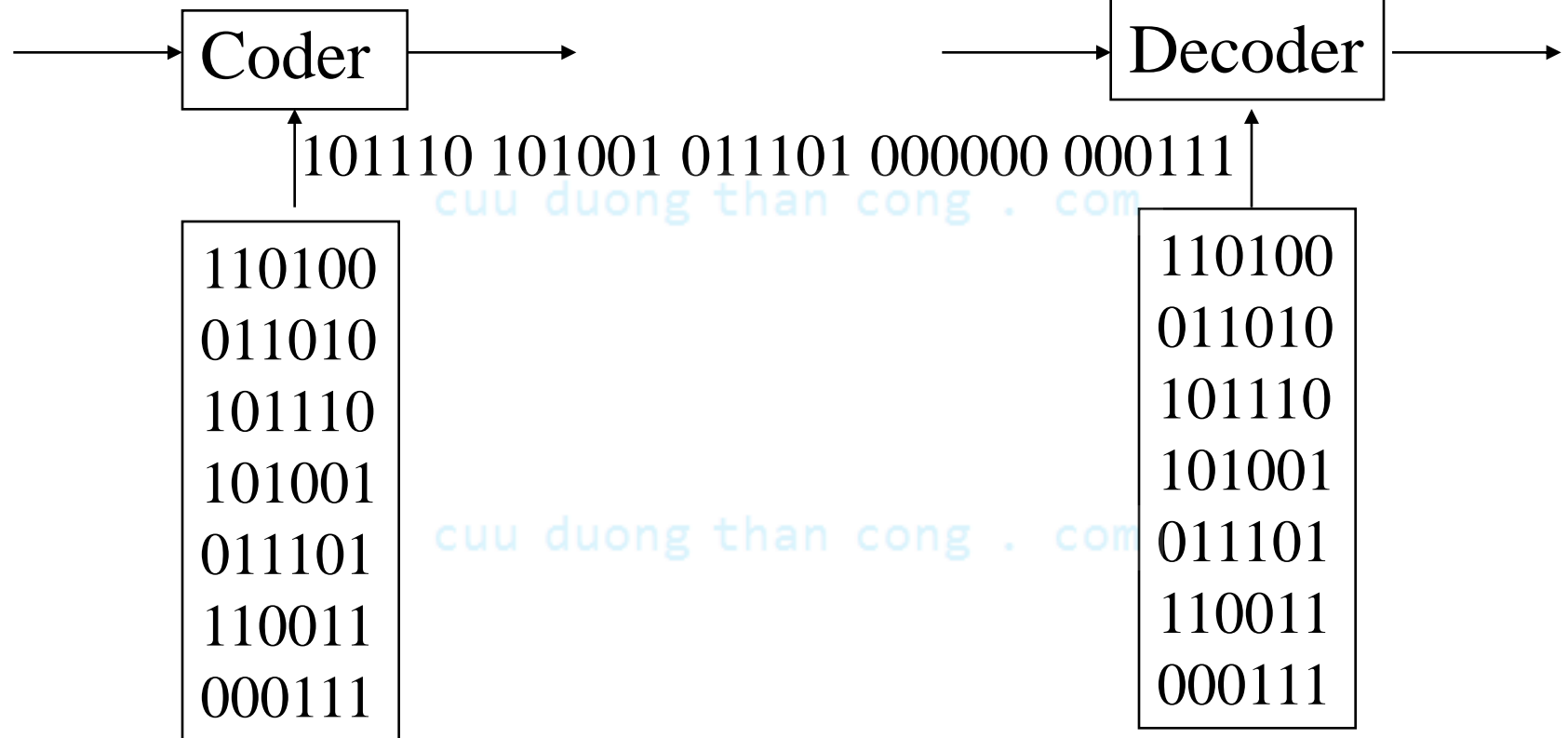
- Bảng mã

| | m_i | V_i | |
|----|-------|---------|------------|
| V0 | 000 | 000 000 | |
| V1 | 100 | 110 100 | |
| V2 | 010 | 011 010 | |
| V3 | 110 | 101 110 | $V1+V2$ |
| V4 | 001 | 101 001 | |
| V5 | 101 | 011 101 | $V1+V4$ |
| V6 | 011 | 110 011 | $V2+V4$ |
| V7 | 111 | 000 111 | $V1+V2+V4$ |

Mã hoá bằng bảng mã

110 001 101 000 111

110 001 101 000 111



Không gian V6

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 000000 | 100000 | 010000 | 110000 | 001000 | 101000 | 011000 | 111000 |
| 000100 | 100100 | 010100 | 110100 | 001100 | 101100 | 011100 | 111100 |
| 000010 | 100010 | 010010 | 110010 | 001010 | 101010 | 011010 | 111010 |
| 000110 | 100110 | 010110 | 110110 | 001110 | 101110 | 011110 | 111110 |
| 000001 | 100001 | 010001 | 110001 | 001001 | 101001 | 011001 | 111001 |
| 000101 | 100101 | 010101 | 110101 | 001101 | 101101 | 011101 | 111101 |
| 000011 | 100011 | 010011 | 110011 | 001011 | 101011 | 011011 | 111011 |
| 000111 | 100111 | 010111 | 110111 | 001111 | 101111 | 011111 | 111111 |

Thí dụ mã (127,63)

- Bảng mã có kích thước: $(2^K - 1)n$ bit
- Bảng mã (127,63) cần dung lượng RAM:
 $(2^{63} - 1)127 \text{ bit} \sim 127 \cdot 2^{30} \cdot 2^{30} \cdot (2^3) \sim 127 \text{ Tỷ GByte}$

cuu duong than cong . com

Ma trận sinh

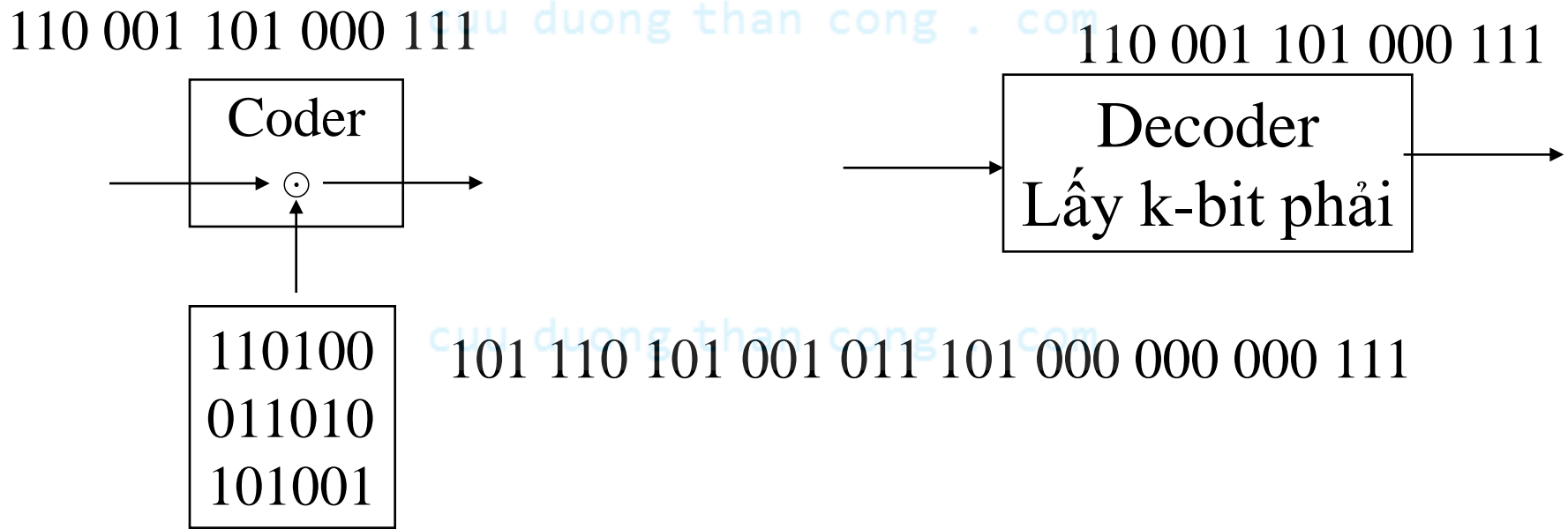
- Định nghĩa: Mã khối tuyến tính (n,k) được đặc trưng bằng ma trận sinh \mathbf{G} có kích thước $(\mathbf{k} \times \mathbf{n})$, trong đó mỗi hàng là một véc tơ cơ sở

cuu duong than cong . com

Mã hoá bằng ma trận sinh

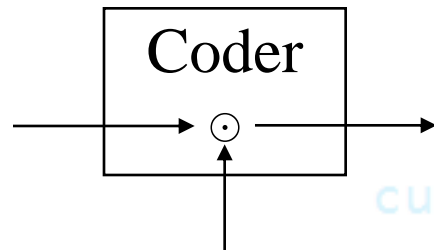
$$v_i = m_i \odot G$$

và giải mã nhờ tính chất hệ thống



Mã (9,4)

1011 0011 0010 1000 1010 1001



| |
|-----------|
| 001111000 |
| 010110100 |
| 111000010 |
| 011100001 |

10101 1011 10010 0011 11100 0010

1011 0011 0010



Tính chất hệ thống (systematic)

- Một mã khối (n,k) được gọi là có tính chất hệ thống nếu k -bit bên phải của vec tơ từ mã chính là vec tơ mang tin.

cuu duong than cong . com

Phát hiện lỗi nhờ ma trận kiểm tra H

- Định nghĩa: Một mã khối (n,k) được đặc trưng bằng ma trận sinh G thì cũng tồn tại một ma trận kiểm tra H sao cho $G.H^T=0$.
- Nếu mã có tính chất hệ thống thì ma trận sinh và ma trận kiểm tra chuyển vị có dạng sau:

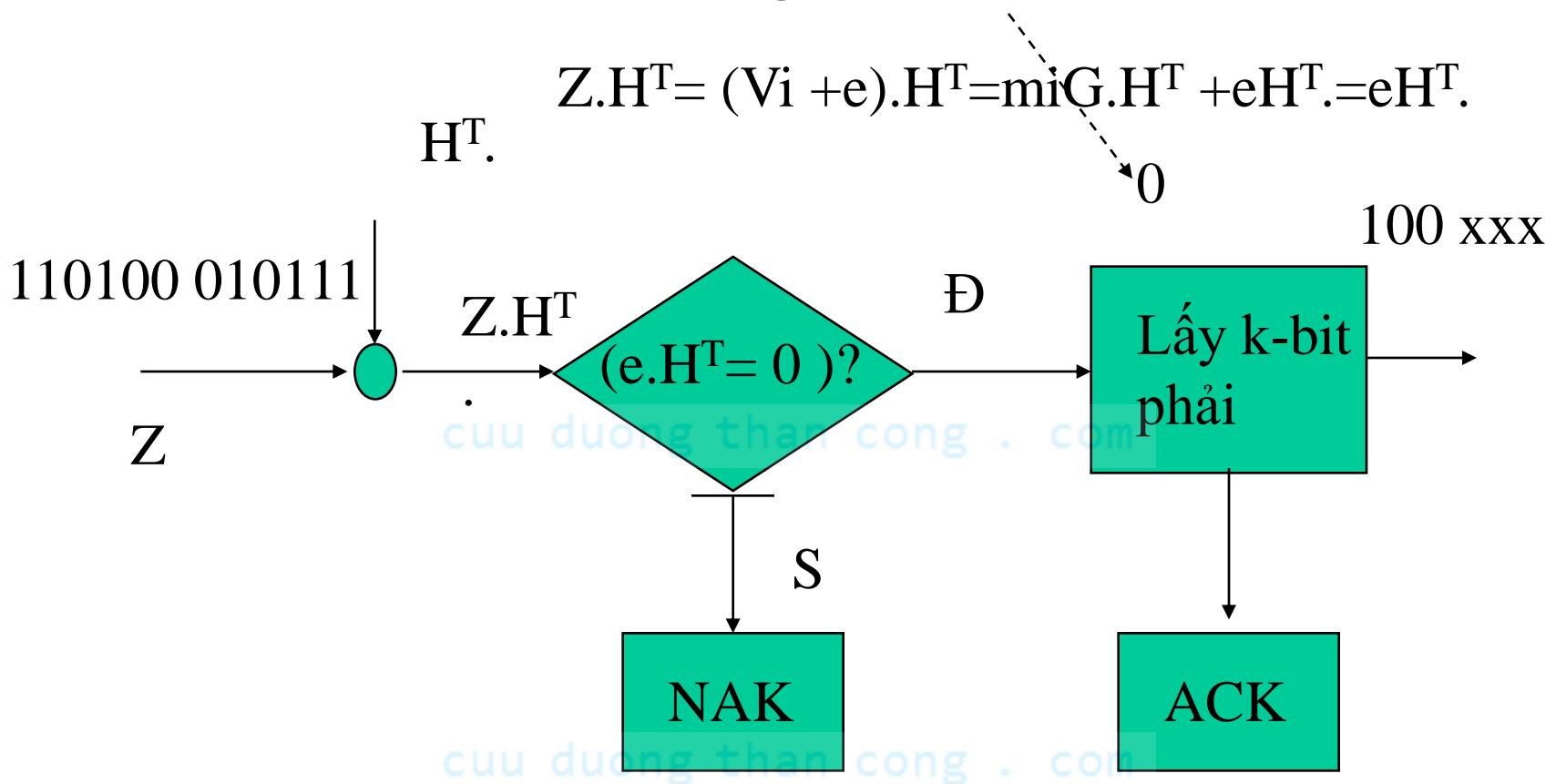
$$G = \begin{bmatrix} I_k & P_{k, n-k} \end{bmatrix}$$

$$H^T = \begin{bmatrix} P_{k, n-k}^T \\ I_{n-k} \end{bmatrix}$$

Thí dụ mã (6,3)

$$G.H^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Phát hiện lỗi bằng ma trận kiểm tra



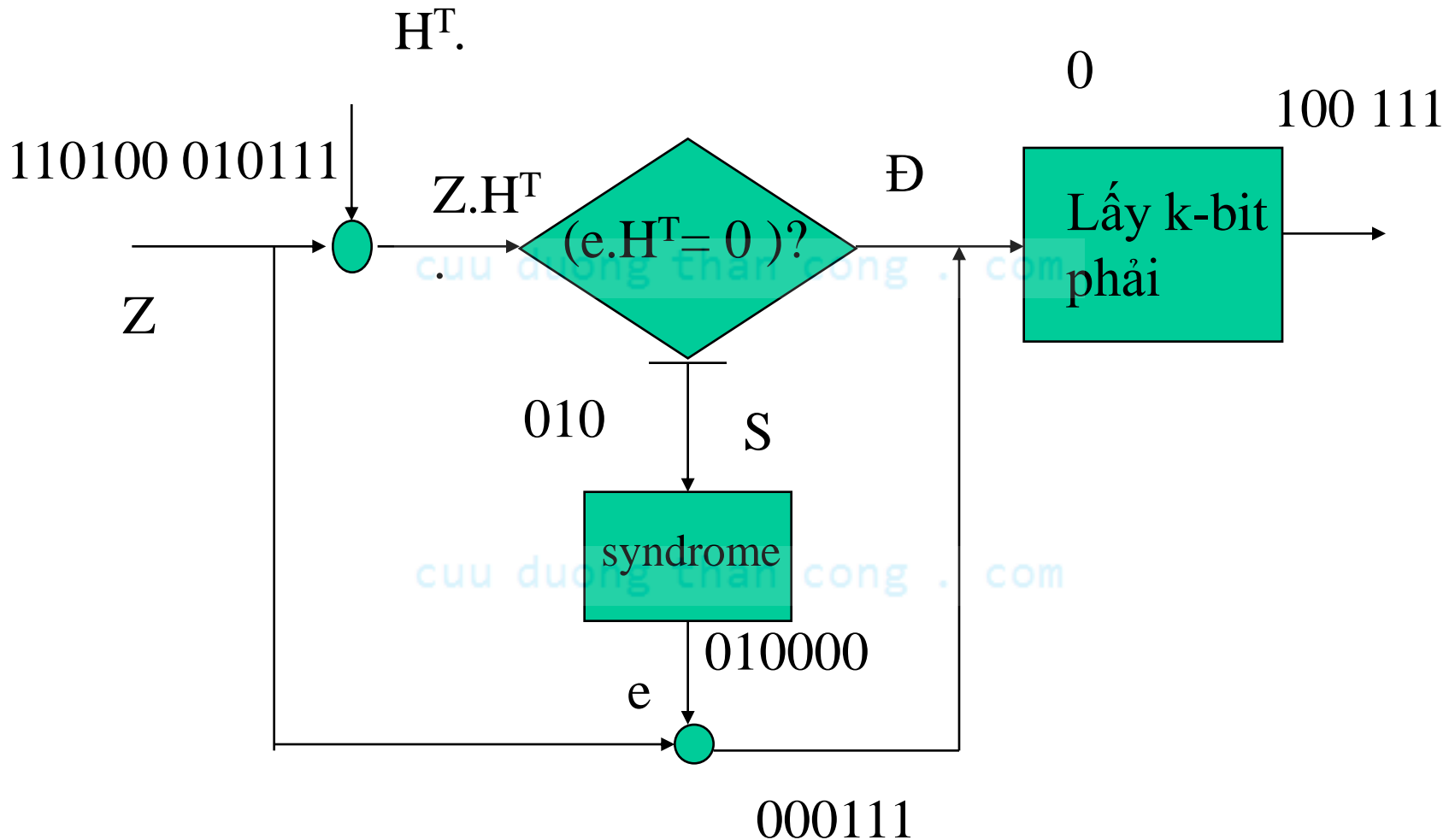
e là vector lỗi, có kích thước n -bit và vị trí lỗi ứng với digit 1

Lưu ý

- Acknowledgment (ACK): phúc đáp dương tính
- Negative Acknowledgment (NAK) : phúc đáp âm tính

cuu duong than cong . com

Sửa lỗi bằng bảng triệu chứng (syndrome)



Bảng triệu chứng của mã (6,3)

| E | V0 | V1 | V2 | ... | V7 | syndrome |
|---------------|---------------|---------------|---------------|-----|---------------|------------|
| 000000 | 000000 | 110100 | 011010 | ... | 000111 | 000 |
| 100000 | 100000 | 010100 | 111010 | ... | 100111 | 100 |
| <i>010000</i> | <i>010000</i> | <i>100100</i> | <i>001010</i> | | <i>010111</i> | <i>010</i> |
| 001000 | 001000 | 111100 | 010010 | | 001111 | 001 |
| 000100 | 000100 | 110000 | 011110 | | 000011 | 110 |
| 000010 | 000010 | 110110 | 011000 | | 000101 | 011 |
| 000001 | 000001 | 110101 | 011011 | | 000110 | 101 |
| 010100 | 010100 | 100000 | 001110 | ... | 010011 | 100 |
| 011010 | 011010 | 101110 | 000000 | ... | 011101 | 000 |

Các vectơ lỗi có triệu chứng bằng không

| E | V0 | V1 | V2 | ... | V7 | syndrome |
|--------|--------|--------|--------|-----|--------|----------|
| 000000 | 000000 | 110100 | 011010 | ... | 000111 | 000 |
| 110100 | 110100 | 000000 | 101110 | | 110011 | 000 |
| 011010 | 011010 | 101110 | 000000 | ... | 011101 | 000 |
| 101110 | 101110 | | | | 101001 | 000 |
| 101001 | 101001 | | 110011 | | 101110 | 000 |
| 011101 | 011101 | | | | | 000 |
| 110011 | 110011 | | 101001 | | | 000 |
| 000111 | 000111 | | | ... | 000000 | 000 |
| | | | | | | |

Bài tập

- Mã (8,3)
- Ma trận sinh
- Ma trận kiểm tra chuyển vị

$$G = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$H^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Bảng mã (8,3)

| mi | vi | W(vi) | |
|-----|----------|-------|----------|
| 000 | 00000000 | | |
| 100 | 00111100 | 4 | V1 |
| 010 | 10101010 | 4 | V2 |
| 110 | 10010110 | 4 | V1+V2 |
| 001 | 11100001 | 4 | V4 |
| 101 | 11011101 | 6 | V1+V4 |
| 011 | 01001011 | 4 | V2+V4 |
| 111 | 01110111 | 6 | V1+V2+V4 |

cuu duong than cong . com

cuu duong than cong . com

Khả năng phát hiện và sửa lỗi

| α | β |
|----------|---------|
| 3 | 0 |
| 2 | 1 |
| | |

- Khoảng cách Hamming tối thiểu $d_{\min} = 4$.
- Khả năng phát hiện lỗi tối đa $e = 3$.
- Khả năng sửa lỗi tối đa $t = 1$.
- Khả năng phát hiện và sửa lỗi đồng thời:
- Xác suất lỗi không phát hiện được

$$P_{nd} = \sum_{j=1}^n A_j P_B^j (1 - P_B)^{n-j}$$

$$A_j = \{0, 0, 0, 5, 0, 2, 0, 0\}$$

$$P_{nd} = 5 P_B^4 (1 - P_B)^4 + 2 P_B^6 (1 - P_B)^2$$

$$P_B \leq 10^{-3}$$

$$P_{nd} \approx 5 \cdot 10^{-12} + 2 \cdot 10^{-18}$$

Mã (9,4)

- Ma trận sinh

$$G = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Ma trận kiểm tra chuyển vị

$$H^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Bảng mã

| mi | vi | W(vi) | |
|------|------------|-------|-----------------|
| 0000 | 00000 0000 | | V0 |
| 1000 | 00111 1000 | 4 | V1 |
| 0100 | 01011 0100 | 4 | V2 |
| 1100 | 01100 1100 | 4 | V3=V1+V2 |
| 0010 | 11100 0010 | 4 | V4 |
| 1010 | 11011 1010 | 6 | V5=V1+V4 |
| 0110 | 10111 0110 | 6 | V6=V2+V4 |
| 1110 | 10000 1110 | 4 | V7=V1+V2+V4 |
| 0001 | 01110 0001 | 4 | V8 |
| 1001 | 01001 1001 | 4 | V9=V1+V8 |
| 0101 | 00101 0101 | 4 | V10=V2+V8 |
| 1101 | 00010 1101 | 4 | V11=V1+V2+V8 |
| 0011 | 00101 0011 | 4 | V12=V4+V8 |
| 1011 | 10101 1011 | 4 | V13=V1+V4+V8 |
| 0111 | 01110 0111 | 6 | V14=V2+V4+V8 |
| 1111 | 01001 1111 | 6 | V15=V1+V2+V4+V8 |

Khả năng phát hiện và sửa lỗi

- Khoảng cách Hamming tối thiểu $d_{\min} = 4$.
- Khả năng phát hiện lỗi tối đa $e = 3$.
- Khả năng sửa lỗi tối đa $t = 1$.
- Khả năng phát hiện và sửa lỗi đồng thời:
- Xác suất lỗi không phát hiện được

| α | β |
|----------|---------|
| 3 | 0 |
| 2 | 1 |

$$P_{nd} = \sum_{j=1}^n A_j P_B^j (1 - P_B)^{n-j}$$

$$A_j \ni 0, 0, 0, 10, 0, 5, 0, 0, 0$$

$$P_{nd} = 10 P_B^4 (1 - P_B)^5 + 5 P_B^6 (1 - P_B)^3$$

$$P_B \leq 10^{-3}$$

$$P_{nd} \approx 10 \cdot 10^{-12} + 5 \cdot 10^{-18}$$

Thí dụ mã (7,3)

$$G = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} & 1 & 0 & 0 \\ P_{21} & P_{22} & P_{23} & P_{24} & 0 & 1 & 0 \\ P_{31} & P_{32} & P_{33} & P_{34} & 0 & 0 & 1 \end{bmatrix}$$

$$H^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}$$

Thí dụ mã (9,4)

$$G = [P_{k,n-k} \mid I_k]$$

- Ma trận sinh có dạng
- $V_i = m_i.G$

$$G = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & 1 & 0 & 0 & 0 \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & 0 & 1 & 0 & 0 \\ P_{31} & P_{32} & P_{33} & P_{34} & P_{35} & 0 & 0 & 1 & 0 \\ P_{41} & P_{42} & P_{43} & P_{44} & P_{45} & 0 & 0 & 0 & 1 \end{bmatrix}$$

– M1 : 1000

– M2 : 0100

– M4 : 0010

– M8 : 0001

– (M16 : 00001 khi k=5)

$$H^T = \begin{bmatrix} I_{n-k} \\ P_{k,n-k} \end{bmatrix}$$

$$H^T = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \end{bmatrix}$$

cuu duong than cong . com

cuu duong than cong . com