MỤC LỤC

**Giới thiệu chung**

[**I.Mã hóa kênh** 2](#_Toc25920253)

[1.1 Sơ đồ hệ thống truyền tin 2](#_Toc25920254)

[1.2 Định lí Shannon 3](#_Toc25920255)

[1.3 Đánh giá chất lượng truyền dẫn 3](#_Toc25920256)

[**II. Điều khiển lỗi** 4](#_Toc25920257)

[2.1 Nguyên lý 4](#_Toc25920258)

[2.2 Các yếu tố thiết kế tuyến 4](#_Toc25920259)

[2.3 Các phương pháp điều khiển lỗi 4](#_Toc25920260)

[2.4 Phân loại các mã điều khiển lỗi 6](#_Toc25920261)

[**III. Mã khối** 6](#_Toc25920262)

[3.1 Mã khối 6](#_Toc25920263)

[3.2 Mã khối tuyến tính 6](#_Toc25920264)

[3.3 Mã vòng 7](#_Toc25920265)

[3.4 Khoảng cách hamming 7](#_Toc25920266)

[3.5 Khả năng phát hiện và sửa lỗi của mã khối 7](#_Toc25920267)

[3.6 Mã khối tuyến tính 8](#_Toc25920268)

[3.7 Mã khối tuyến tính-giải mã 10](#_Toc25920269)

[**IV. Mã chập**: 11](#_Toc25920270)

[4.1 Khái niệm: 11](#_Toc25920271)

[4.2 Ví dụ mã chập: 12](#_Toc25920272)

[4.3 Biểu diễn mã chập bằng đa thức sinh: 12](#_Toc25920273)

[4.4. Biểu mã chập bằng sơ đồ cây: 13](#_Toc25920274)

[4.5 Biểu diễn mã chập bằng sơ đồ trạng thái: 14](#_Toc25920275)

4.[6 biễu diễn mã chập bằng sơ đồ lưới 14](#_Toc25920276)

[**V. Giải mã mã chập bằng thuật toán Viterbi:** 15](#_Toc25920277)

Giới thiệu chung

Phần đầu của chương này sẽ trình bày tổng quát về điều khiển lỗi áp dụng trong hệ thống thông tin số , bao gồm giới thiệu về các phương pháp điều khiển lỗi, phân loại các mã điều khiển lỗi.

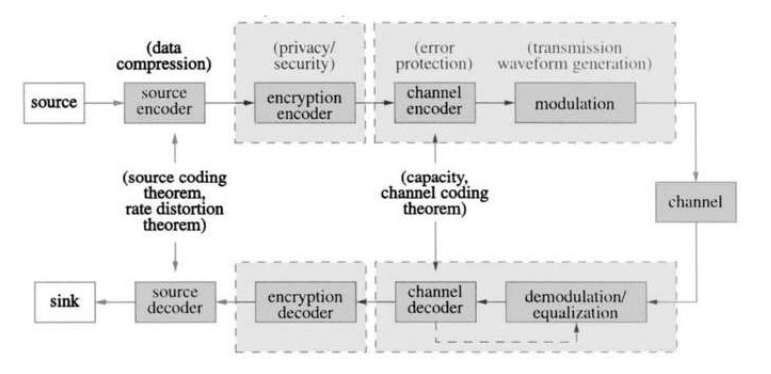
Phần sau của chương tập trung vào các loại mã điều khiển lỗi, bao gồm hai loại chính là mã khối (*Block Code*) và mã chập (*Convolutional Code*).

Phần mã khối tuyến tính (*Linear Block Code*) sẽ tập trung trình bày về mã vòng (*Cyclic Code*) và một loại mã vòng đơn giản là mã Hamming.

Phần mã chập ở cuối chương sẽ trình bày phương pháp dùng sơ đồ cây (*Tree Diagram*), sơ đồ lưới (*Trellis Diagram*) và sơ đồ trạng thái *(State Diagram*) để minh họa cho quá trình mã hóa mã chập. Phần giải mã mã chập trình bày thuật toán Viterbi dùng sơ đồ lưới.

# I.Mã hóa kênh

## 1.1 Sơ đồ hệ thống truyền tin



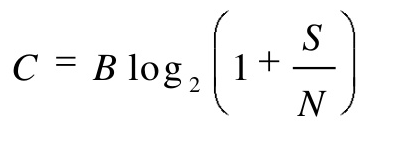
* Hệ thống thông tin gồm 3 thành phần:
  + Nguồn phát tin
  + Kênh truyền tin
  + Bên nhận tin
* Encrypter: mã hóa, ẩn giấu thông tin ban đầu từ nguồn phát tin, nhằm tránh sự xâm phạm thông tin không mong muốn.( Toàn vẹn thông tin, bảo mật).
* Channel Coder: bộ mã hóa kênh. Bổ sung thông tin dư thùa nhằm cho phép việc phát hiện/sửa lỗi thông tin ở phía bên nhận tin.
* Modulator: bộ điều chế tín hiệu. Chuyển đổi dòng tín hiệu số (*bit, digital symbol*) thành tín hiệu phug hợp với việc truyền dẫn trên kênh truyền.
* Channel: kênh thông tin, là môi trường truyền dẫn thông tin từ nguồn tin tới bên nhận tin.

Phía bên nhận tin, quá trình xử lí thu nhận và xử lí thông tin tương ứng với bên phát tin: từ bộ giải điều chế (*demodulator*), tới bộ giải mã kênh (*channel decoder*) và giải mã hóa (*dencrypter*).

## 1.2 Định lí Shannon

Theo quan điểm của ngành thông tin, tài nguyên thông tin chủ yếu là công suất, thời gian và băng thông của tín hiệu. Đối với một môi trường thông tin cho trước, ba tài nguyên này có thể mâu thuẩn lẫn nhau. Việc cân đối các mâu thuẫn này tùy vào trường hợp cụ thể. Tuy nhiên, nhìn chung thì ta có thể đạt được tốc độ truyền số liệu cao nhất trong một băng thông nhỏ nhất Trong khi giữ cho chất lượng truyền dẫn ở mức chấp nhận được. Trong thông tin số thì chất lượng truyền dẫn có liên quan mật thiết với xác suất lỗi bit Pb tại đầu thu.

* Định lý về dung lượng kênh của Shannon:



Trong đó:

S: Công suất tín hiệu được truyền đi (W)

N: Mật độ nhiễu trung bình trong toàn kênh (W)

B: Băng thông của kênh (Hz)

C: Dung lượng giới hạn của kênh truyền (bit/s)

Đã chỉ ra giới hạn lý thuyết của tốc độ truyền số liệu từ bộ phát có công suất cho trước, qua một kênh với băng thông cho trước, hoạt động trong môi trường có nhiễu đã biết. Tuy nhiên, để thức hiện được giới hạn lý thuyết này, ta phải tìm được một phương pháp mã hóa phù hợp ( theo Shannon thì phương pháp này có tồn tại ).

## 1.3 Đánh giá chất lượng truyền dẫn

Đại lượng đo lỗi thông thường là tỷ lệ lỗi bit BER (*Bit Error Rate*) hay xác suất lỗi bit (Pb). Pb đơn giản là xác suất một bit nhị phân bất kỳ truyền đi bị lỗi. BER là tỷ số trung bình, được tính là tích của Pb và Rb, ở đây Rb là tốc độ bit trong kênh.

* Pb  điển hình trong một số hệ thống:
* Hệ thốngPCM tuyến tính là 10-7
* Hệ thống PCM nén phi tuyến tính là 10-5
* Hệ thống ADPCM là 10-4

Điều khiển lỗi nhằm mục đích là giảm tỷ lệ lỗi trong một hệ thống khi tỷ lệ này lớn quá mức cho phép.

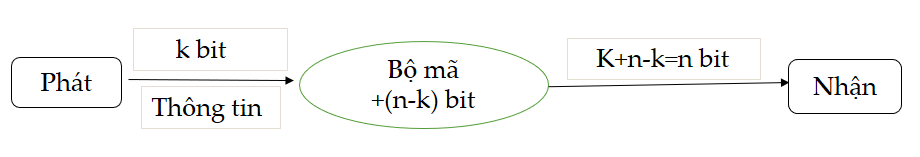
# Điều khiển lỗi

## 2.1 Nguyên lý

Nguyên lý chung là thêm vào tập mã cần truyền 1 tập bit kiểm tra nào đó để bên nhận có thể kiểm soát lỗi

Bên phát: bỏ sung thêm thông tin (thêm bit) và bit cần gửi

Bên thu: nhận thông tin bổ sung ở phía phát, kiểm tra phát hiện và sửa lỗi



n-k : bit kiểm tra

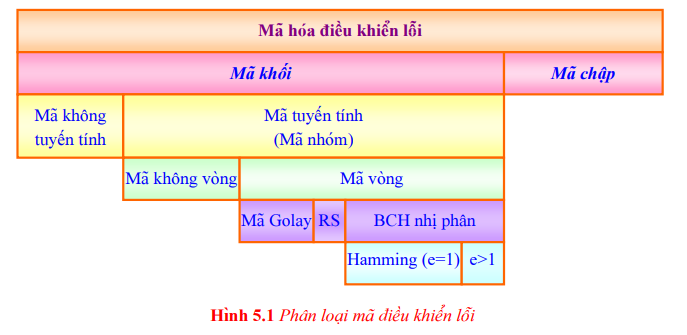
## 2.2 Các yếu tố thiết kế tuyến

* Truyền thông tin với tốc độ bit yêu cầu tùy theo dịch vụ.
* Truyền thông tin trong một băng thông hạn chế của một kênh truyền sẵn có.
* Truyền thông tin với một công suất hạn chế với một ứng dụng cụ thể
* Truyền thông tin trong một khoảng thời gian trễ hạn chế.
* Chất lượng dẫn truyền ở mức chấp nhận được

## 2.3 Các phương pháp điều khiển lỗi

* Giải pháp đầu tiên và dễ thấy nhất là tăng công suất phát, nhưng không phải lúc nào cũng có thể thực hiện được. Ví dụ như đối với một máy điện thoại bỏ túi thì không chấp nhận khối lượng pin quá lớn.
* Giải pháp thứ hai, rất hiệu quả trong việc chống lại lỗi chùm gây ra bởi Fading, là sử dụng phân tập (*Diversity*). Có ba kiểu phân tập chính là phân tập không gian, phân tập tần số và phân tập thời gian. Cả ba kiểu phân tập này đều đưa thêm độ dư vào trong dữ liệu phát bằng cach truyen gấp đôi: qua hai đường, tại hai tần số hay hai thời điểm khác nhau. Trong phân tập không gian, sử dụng antenna đặt tại những vị trí đủ xa để có một trong các antenna đó thu được tín hiệu tốt nhất, ít bị Fading nhất. Phân tập tần số sử dụng hai hay nhiều tần số khác nhau để phát cùng một tin. Phân tập tần số có thể là trong băng hay ngoài băng tùy vào khoảng cách tần số giữa các sóng mang. Trong hệ thống phân tập thời gian, phát cùng một tin nhưng vào hai hay nhiều thời điểm khác nhau.
* Giải pháp thứ ba là truyền song công, hay còn gọi là kiểm tra echo ( *Echo Checkin*). Ở đây, khi bộ phát tin đến bộ thu, tin được phát ngược về bộ phát trên một kênh hồi tiếp riêng. Nếu tin phát ngược về khác với tin đi thì biết là có lỗi. Phương pháp này có khuyết điểm là yêu cầu băng thông gấp đôi so với truyền trên một hướng nên không chấp nhận khi tận dụng phổ.
* Phương pháp thứ tư để đối phó với BER cao là yêu cầu lặp lại tự động ARQ (Automatic Repeat Request). Trong hệ thống ARQ, mã phát hiện lỗi (*Error Detecting Code*) được sử dụng để bên thu kiểm tra lỗi trong khối số liệu thu và trả lời cho bên phát trên một kênh hồi tiếp. Tín hiệu trả lời là chấp nhân ACK (*ACKnowledgment*) khi số liệu thu đúng là không chấp nhân NAK (*Non – AcKnowledgment*) khi số lệu thu sai. Nếu bên phát nhận NAK, Bên phát phải tiến hành truyền lại khối số liệu bị lỗi. Có hai kỹ thuật ARQ chính là dừng và đợi (*Stop And Wait ARQ*) và ARQ liên tục (*Continuous ARQ*). Trong hệ thống ARQ dừng và đợi, sau khi phát khối số liệu đi, bên phát dừng lại và chờ nhận trả lời từ bên thu, rồi tùy theo trả lời đó là ACK hay NAK mà bên phát phát khối sô liệu tiếp theo hay phát lại khối sô liệu vừa rồi. Nếu thời gian chờ quá thời gian quy định (gọi là *time-out*), bên phát coi như khối dữ liệu vừa phát bị lỗi và tiến hành phát lại. Hạn chế của phương pháp này là thời gian trễ truyền dẫn lớn. Trong hệ thống ARQ liên tục, các khối sô liệu đều mang số thứ tự - N – và bản tin trả lời ACK/NAK cũng mang sô thứ tự N tương ứng. Bên phát liên tục phát đi các khối số liệu mà không chờ nhận trả lời từ bên thu. Bên thu kiếm tra lỗi các khối số liệu thu và trả lời về cho bên phát bản tin ACK/NAK kèm theo số thứ tự của khối tin tương ứng. Khi nào bên phát nhận trả lời NAK từ bên thu , bên phát sẽ phát lại tất cả các khối số liệu kể từ khối số bị lỗi đối với ARQ lùi lại N (*Go-Back-N ARQ*), hoặc bên phát sẽ chỉ phát lại khối số liệu bị lỗi đối với ARQ chọn lọc (*Selective ARQ*). Mặc dù ARQ chọn lọc rất hiệu quả trong sử dụng băng thông nhung yêu cầu dung lượng bộ nhớ lớn hon ARQ lùi lại N, đặc biệt trong các kết nối tốc độ cao. ARQ phù hợp với các hệ thống thông tin máy tính, vì ở đó có sẵn kênh truyền song công để bên thu có thể phát lại cho bên phát bản tin ACK/NAK. Tuy nhiên, trong các đường truyền dài với tốc độ cao, điển hình như thông tin vệ tinh thì rất khó thực hiện ARQ.
* Phương pháp thứ năm để giảm BER là thực hiện mã hóa sửa lỗi không phản hồi FECC(*Forward Error Correction Coding*). Trong lịch sử, việc chấp nhận sử dụng rộng rãi FECC có trễ hơn so với các phương pháp khác, bởi vì độ phức tạp và giá cả của nó cao hơn. Ngày nay, độ phức tạp đã giảm xuống nhờ sự gia tăng các chip mã hóa/ giải mã VLSI.PECC lợi dụng sự khác nhau giữa tốc độ truyền dẫn và thông lượng kênh để giảm xác suất lỗi Pb. Việc giảm xác suất lỗi bị trả giá bằng việc tăng thời gian trễ truyền dẫn, do tăng độ dư cho đủ để mã có thể phát hiện và sửa được lỗi và do mất thời gian kiểm tra khối số liệu thu để sửa lỗi. Tuy nhiên, lợi ích của FECC có được thường nhiều hơn khuyết điểm về độ trễ lớn.

## 2.4 Phân loại các mã điều khiển lỗi



# Mã khối

## 3.1 Mã khối

* Mã khối đặc trưng bởi hai số nguyên n và k .
* Bộ mã hóa mã khối với k bit tin và n bit mã hóa ra. Từ mã n bit được tạo ra duy nhất từ k bit tin và (n-k) là số bit kiểm tra dư
* Tỷ lệ mã R=k/n là tiêu chuẩn để đánh giá độ dư của mã. Tỷ lệ mã thường nằm trong khoảng ½ đến 1.

Bộ mã hóa khối

n bit tin n bit mã hóa

k bit (n-k) bit

Phần tin phần dư

Từ mã n bit

Hình 2.1 Mã khối hệ thống (n, k)

## 3.2 Mã khối tuyến tính

* Có chứa các từ mã gồm toàn số 0
* Có tính chất đóng. Ta xét với mã nhị phân. Tức tổng hai từ mã bất kì trong bộ mã bằng 1 từ mã nằm trong bộ mã đó.

Với hai từ mã ci, cj bất kỳ ci+cj =ck ,ck cũng là một từ mã

|  |  |
| --- | --- |
| Giá trị | Từ mã |
| a | 000000 |
| b | 110011 |
| c | 001100 |
| d | 111111 |

c , b

ta có :1 , 0

## 3.3 Mã vòng

* Nếu loại bỏ từ mã chỉ toàn bit 0 thì có mã khối con mà mã vòng.
* Là lớp con của mã khối tuyến tính.
* Không gồm từ mã toàn bit 0.
* Có tính chất dịch vòng . Dịch vòng một từ mã cũng được một từ mã thuộc cùng bộ mã.

Nếu W=a0a1 …an-2 an-1 là một từ mã thì V= an-1 a0a1 …an-2  cũng là một từ mã

* Dịch vòng được thực hiện từ trái sang phải

Ví dụ:

0110001, 1100010, 1000101, 0001011, 0010110, 0101100, 1011000.

## 3.4 Mã hammng và khảng cách hamming

* Mã hamming có khả năng sửa sai 1 lỗi.
* Mã khối có khả năng sửa lỗi phụ thuộc vào khoảng cách hamming.
* Khoảng cách hamming giữa hai từ mã có chiều dài bằng nhau được định nghĩa là số vị trí có giá trị khác nhau giữa hai từ mã đó.Kí kiệu là d
* Khoảng cách hamming của một bộ mã được xác định là khoảng cách hamming nhỏ nhất giữ các từ mã.

Ví dụ:

c1 =01010 ,c2  =10010, c3=11110 là bộ mã gồm 3 từ mã đều thuộc bộ mã c

Chiều dài của từ mã là 5 (kí hiệu L(c)=5)

d(c1 ,c2 )=2

d(c1 ,c3 )=2 =>d(c) =2

d(c3 ,c2 )=2

* Trong lượng hamming chỉ số các ký tự có giá trị là 1. Kí hiêu W.

## 3.5 Khả năng phát hiện và sửa lỗi của mã khối

* Mỗi quan hệ giữa khoảng cách hamming và khả năng phát hiện và sửa lỗi của mã khối
* Khoảng cách hamming giữa các từ mã trong cùng một bộ mã có liên quan đến khả năng phát hiện sai và sửa sai của bộ mã đó
* Cụ thể: d (1)

Trong đó: d là khoảng cách hamming

r số lỗi phát hiện được

s số lỗi sửa được.

s.

**Minh họa:**

* Giả sử cho bộ mã C có 8 từ mã như sau

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ký tự | A | B | C | D | E | F | G | H |
| Từ mã | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

* Từ bộ mã trên tạo 1 bộ mã C­1 có khoảng cách hamming đều bằng 2. Chọn từ mã B(001) là từ mã xuất phát.

Ta có bộ mã C1 như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ký tự | B | C | E | H |
| Từ mã | 001 | 010 | 100 | 111 |

* Từ bộ mã trên tạo 1 bộ mã C­3 có khoảng cách hamming đều bằng 3. Chọn từ mã B(001) là từ mã xuất phát.

Ta có bộ mã C3 như sau:

|  |  |  |
| --- | --- | --- |
| Ký tự | B | G |
| Từ mã | 001 | 110 |

* bộ mã C1(d=1) không có khả năng phát hiện lỗi
* bộ mã C2(d=2) phát hiện được 1 lỗi ,không sửa được lỗi
* bộ mã C3(d=3) phát hiện và sửa được một lỗi
* Ta kết luận được rằng: nếu khoảng cách hamming là 2 thì có khả năng phát hiện được 1 lỗi, nếu khoảng cách hamming là 3 thì có khả năng sửa được 1 lỗi và phát hiện được 2 lỗi.

## 3.6 Mã khối tuyến tính

a. ví dụ về mã khối tuyến tính

cho một mạch tạo mã khối tuyến tính (4,7) gồm 4 bit tin (I1 đến I4) và 3 bit kiểm tra ( P1 đến P3)

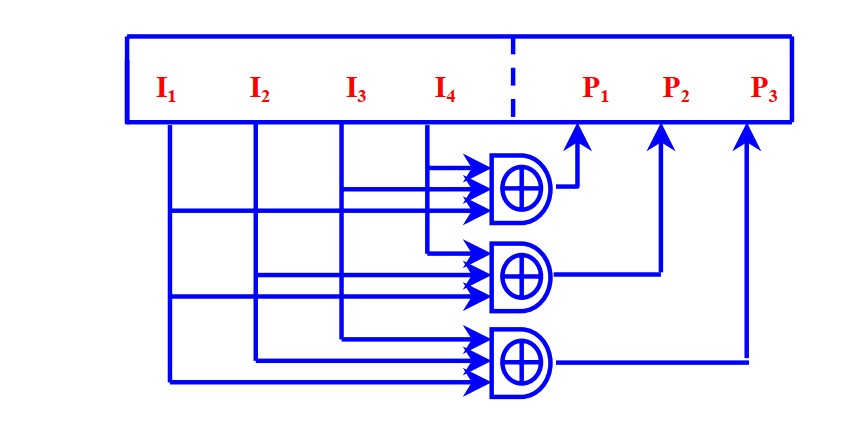
gia sử mỗi qua hệ giữa bit tin và bit kiểm tra là:

P1 =

P2 =

P3 =

Nếu các bit tin I1 =1, I2=0, I3=1, I4=1, các bit P1=1, P2 =0, P3=0.



Hình a.1 mạch tạo mã khối (4x7)

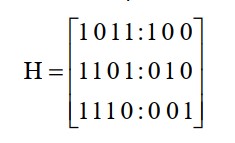
Viết lại mối quan hệ giữa bit tin và bit kiểm tra như sau

P1 =

P2 =

P3 =

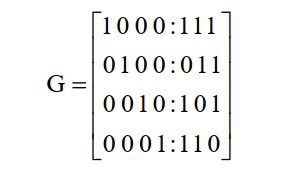
Ta được ma trận kiểm tra H như sau:



Bên phải của đường chấm chấm là các hệ số của các bit tin I1 đến I4 , bên phải đường chấm chấm là ma trận đơn vị.

b. Ma trận sinh

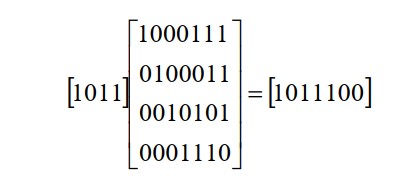
* Ma trận sinh ký hiệu là G, là một ma trận 4x7, tạo ra bằng cách kết hợp một ma trận đơn vị 4x4 với ma trận hoán vị của ma trận bên trái đường chấm chấm của H. ma trận sinh trong ví dụ là:



Mã khối tuyến tính bằng vector tin nhân với ma trận G. kí hiệu là

C=m x G

Ta được mã khối tuyến tính (4 x 7) như sau:



Ta xác định được bit tin là 1011

Bit dư là 100

Từ mã khối phát đi là 1011100

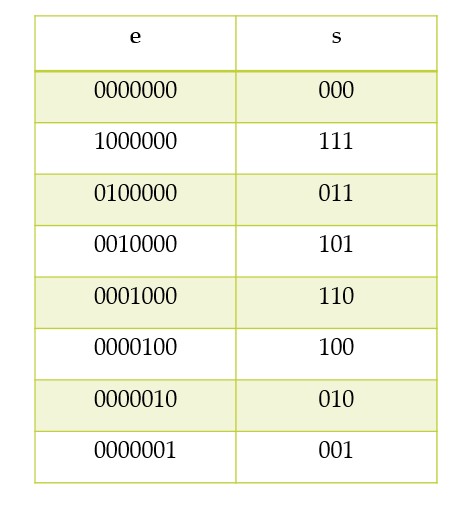
## 3.7 Mã khối tuyến tính-giải mã

1. Syndrome –vector sửa sai

* Syndrome là từ mã độc lập với từ mã đã phát, phụ thuộc vào từ mã thu bị lỗi.
* Gọi từ mã phát đi là t= (t0 t1 …tn-1 )
* Gọi từ mã thu được là r = (r0 r1 …rn-1 )
* Gọi vector sau là e= (e0 e1 …en-1 )

Trong đó ei =1 nếu ti ri, ei =0 nếu ti ri .

b. Bảng syndrome:

Tập hợp tất cả các syndrome có thể có ( trong trường hợp sai một lỗi ). 

* s=0 nếu và chỉ nếu r là từ mã phát đi ( r t ) hoặc vector sai không phát hiện được.
* s 0 thì r không phải là từ mã phát đi (r t ) và do đó có sai e 0.
* Các syndrome không giống nhau nên có thể căn cứ vào syndrome để biệt được vị trí lỗi, từ đó ta có thể sửa lỗi

c. Giải mã

* vector syndrome là s
* c là vector biểu diễn cho từ mã khối (n,k)

c=m x G

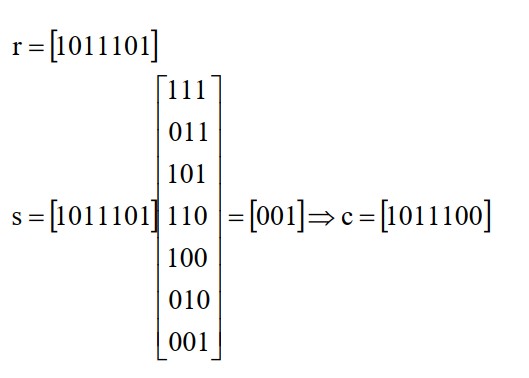
* e là vector lỗi, r là từ mã thu

r= c

* syndrome được tính như sau:

s= r x = (c ) x =c x x =0 x= e x

xét trên ví dụ trên, giả sử thu được từ mã 1011101, từ mã được sửa như sau

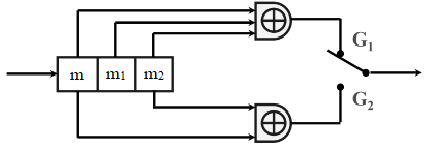


# IV. Mã chập:

## 4.1. khái niệm:

Mã chập là một loại mã sửa sai được sử dụng trong mã hóa kênh với từ mã được tạo ra nhờ việc cộng các bit thông tin và các bit có trong các thanh ghi dịch. Để tạo ra mã chập, ta dùng bộ lập mã chập có nguyên tắc hoạt động dựa vào việc “chập” dòng bit thông tin đầu vào với đáp ứng xung của bộ tạo mã.

Ta có mô hình của một hệ thống tạo mã chập:



Hình 1: sơ đồ khối của bộ tạo mã chập có k = 1, n = 2, K = 3;

Theo sơ đồ trên :

G1 = m + m­1 + m2

G2 = m + m2

Trong đó :

* k: số bít vào bộ mã hóa tại một thời điểm.
* n: số bít ra tại một thời điểm.
* K: độ dài ràng buộc.

Từ mã phụ thuộc vào :

* Bit thứ K vào hiện tại.
* (K-1) bit vào trước đó.

## 4.2. Ví dụ mã chập:

Dựa theo sơ đồ khối ở hình 1:

Tham số mã: k = 1, n = 2, K = 3;

Chuỗi vào : 1010

Giai đoạn mã hóa thực hiện như sau:

* Thời điểm ban đầu: thanh ghi là 000
* Bit 1 vào: thanh ghi là 100

G1 = 1 + 0 + 0 = 1

G2 = 1 + 0 = 1

Từ mã 11

* Bit 0 vào: thanh ghi là 010

G1 = 0+ 1 + 0 = 1

G2 = 0 + 0 = 0

Từ mã 10

* Bit 1 vào: thanh ghi là 101

G1 = 1 + 0 + 1 = 0

G2 = 1 + 1 = 0

Từ mã 00

* Bit 0 vào: thanh ghi là 010

G1 = 0 + 1 + 0 = 1

G2 = 1 + 0 = 0

Từ mã 01

Chuỗi mã hóa: 11100010

Đây là cách bộ bã hóa hoạt động.

## 4.3. Biểu diễn mã chập bằng đa thức sinh:

Có thể biểu diễn bộ mã hóa mã chập bằng các đa thức sinh. Mỗi đa thức sinh biểu diễn cho một bộ cộng modulo 2. Đa thức sinh có bậc <= K-1 miêu tả sự kết nối giữa đầu ra của một vị trí trong thanh ghi dịch với một bộ modulo 2. Theo Hinh 1 thì hai đa thức sinh là G1 = 1 + 1.x + 1.x2  và G2 = 1 + 1.x2

* Chuỗi vào: m(x)
* Chuỗi mã hóa: U(x) = [m(x).G1(x),m(x).G2(x)]

Ví dụ:

Chuỗi vào: 1010 => m(x) = 1 +0.x + 1.x2  + 0.x3

m(x).G1 = (1 +0.x + 1.x2  + 0.x3)( 1 + 1.x + 1.x2) = 1 +1.x + 0.x2  + 1.x3+ 0.x4  + 0.x5

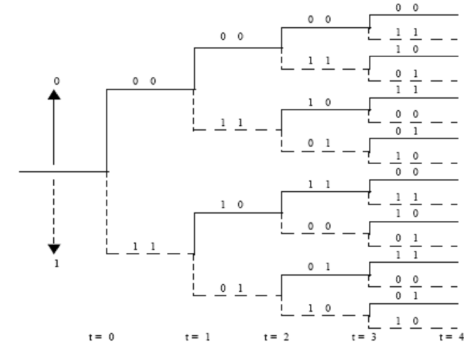
m(x).G2 = (1 + 1.x2)(1 + 1.x + 1.x2) = 1 +0.x + 0.x2  + 0.x3 + 1.x4  + 0.x5

U(x) = [(1,1) +(1,0)x + (0,0)x2  + (1,0)x3 + (0,1)x4  + (0,0)x5]

chuỗi mã hóa: 111000100100

## 4.4. Biểu mã chập bằng sơ đồ cây:

Hình 2 là sơ đồ cây biểu diễn mã chập cho ví dụ trên. Ban đầu thanh ghi có trạng thái là 0, đọc sơ đồ cây theo phương ngang từ trái qua phải, mỗi nhánh cây biểu diễn một từ mã hai bit ra tương ứng với một bit vào. Nếu bit vào là 0 thì sẽ di chuyển sang nhánh tiếp theo ở phía trên, nếu bit vào là 1 thì sẽ di chuyển đến bit tiếp theo ở dưới

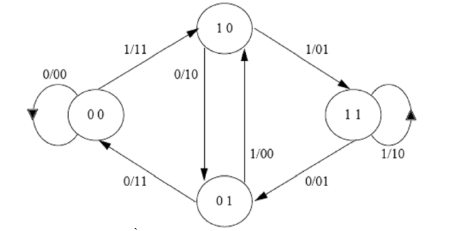


Hình 2. sơ đồ cây mã chập

Giả sử chuỗi vào là 1010 thì ta có chuỗi ra là 11100010. Nếu số bit vào là k thì số nhanh của cây sẽ là 2k , cây rất cồng kềnh.

## **4.5. Biểu diễn mã chập bằng sơ đồ trạng thái**:

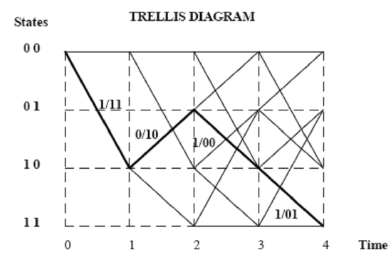
Vì biểu diễn theo sơ đồ cây rất phức tạp nếu số lượng bit vào lớn nên chúng ta có thể biểu diễn bằng sơ đồ trạng thái. Trong sơ đồ trạng thái ta có thể thấy được trạng thái của thanh ghi, bit vào, và bit ra.



Hình 3 sơ đồ trạng thái mã chập

Hình 3 là sơ đồ trạng thái của bộ mã chập trong ví dụ. Phía trong vòng tròn là ký trạng thái của thanh ghi khi có bit vào. Mũi tên là chiều biến đổi trạng thái của các thanh ghi, phía trên là các số có dạng 1/11 chỉ bit vào là 1 và bit ra là 11.

Theo sơ đồ trạng thái nếu ta có một chuỗi vào là 1101 thì bit ra sẽ là 11010110. Sơ đồ trạng thái giúp mã hóa một cách nhanh chóng nhưng lại không dùng để giải mã được.

4.6 Biễu diễn mã chập bằng sơ đồ lưới 

Hinh 4 sơ đồ hinh lưới

Theo sơ đồ trạng thái thì thanh ghi có 4 trạng thái là 00, 01, 10, 11 được sắp xếp từ trên xuống. Theo sơ đồ lưới thì từ nút tại trạng thái 00 và trờ về sau đều đươc chia ra hai nhánh, một là khi bit 0 vào còn lại là khi bit 1 vào. Các chỉ số trên các nhánh có dạng 1/11 có nghĩa là bit vào là 1 và bit ra là 11.

Ví dụ khi có chuỗi vào là 1011 thì chuỗi ra là 11100001. Quan sát sơ đồ lưới ta có thể thấy được trình tự các bit vào trong bộ mã hóa, diều này giúp t có thể giải mã chuỗi nhận được khi có sơ đồ lưới.

# V. Giải mã mã chập bằng thuật toán Viterbi:

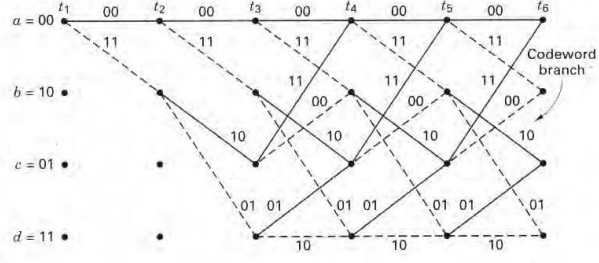
Khi nhận được được một đoạn mã được mã hóa bằng mã chập thì chúng ta có thể áp dụng thuật toán viterbi để giải mã đoạn mã đó.

Những khái niệm cần biết trong thuật toán viterbi:

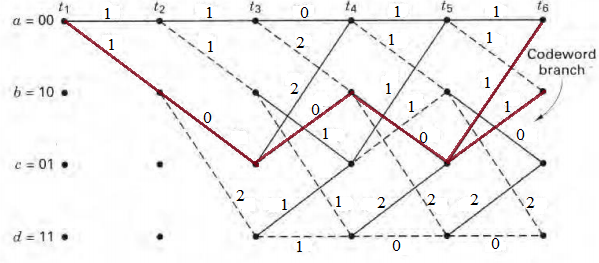
* Metric: khoảng cách Hamming từ từ mã nhận được đền từ mã có được trên lưới.
* Khoảng cách Hamming của đường: tổng metric các nhánh tạo thành đường đó

Trong quá trình giải mã thì chúng ta cần tìm ra đường đi có khoảng cách Hamming là ngắn nhất để có được chuỗi tin đúng.nếu có hai đường đi có độ dài bằng nhau thì ta lựa chọn ngẫu nhiên.

Để giải mã thì chúng ta cần phải có một lưới mã chập tổng quát của bộ mã hóa:



Giả sử chuỗi mã nhận được là 1010001010 , xét lưới mã chập để thấy khoảng cách Hamming của từng nhánh.



Có hai đường đi có độ dài bằng nhau nên chúng ta sẽ chọn ngẫu nhiên một trong hai kết quả là 10100 hoặc 10101.