



Tầng Liên kết dữ liệu

HAMMING CODE - 1

- Mỗi hamming code

- có M bit, đánh số từ 1 đến M
- Bit parity: $\log_2 M$ bits, tại các vị trí lũy thừa của 2
- Dữ liệu thật được đặt tại các vị trí không là lũy thừa của 2
- VD: $M = 7$
 - $\log_2 7 = 3$: dùng 3 bits làm bit parity (1, 2, 4)
 - Có 4 vị trí có thể đặt dữ liệu (3, 5, 6, 7)

- Đặc điểm:

- sửa lỗi 1 bit
- nhận dạng được 2 bit lỗi
- Sửa lỗi nhanh hơn Parity code 2 chiều

HAMMING CODE - 2

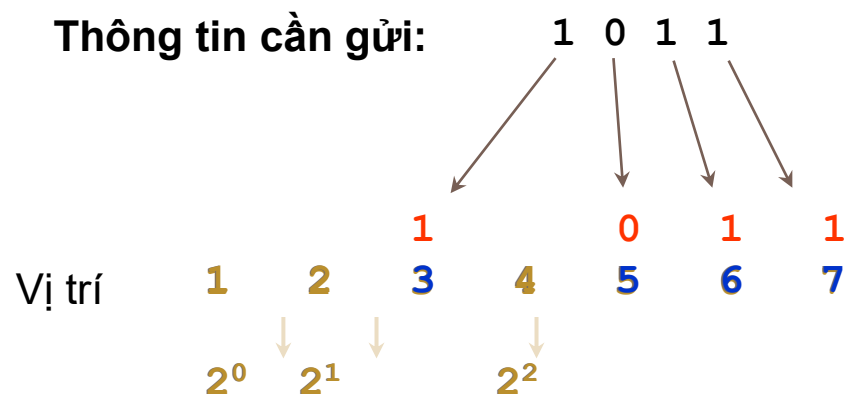
○ Bên gửi:

- Chia dữ liệu cần gửi đi thành các khối dữ liệu (với số bit là số vị trí có thể đặt vào Hamming Code)
- Với mỗi khối dữ liệu → tạo 1 Hamming Code
 - Đặt các bit dữ liệu vào các vị trí không phải là lũy thừa của 2 trong Hamming Code
 - lưu ý: vị trí được đánh số từ 1 đến M
 - Tính lấy bits
 - Tính giá trị của các bit parity

HAMMING CODE – 3

○ Ví dụ:

- $M = 7$
- Dùng parity lẻ
- Thông tin cần gửi: 1011

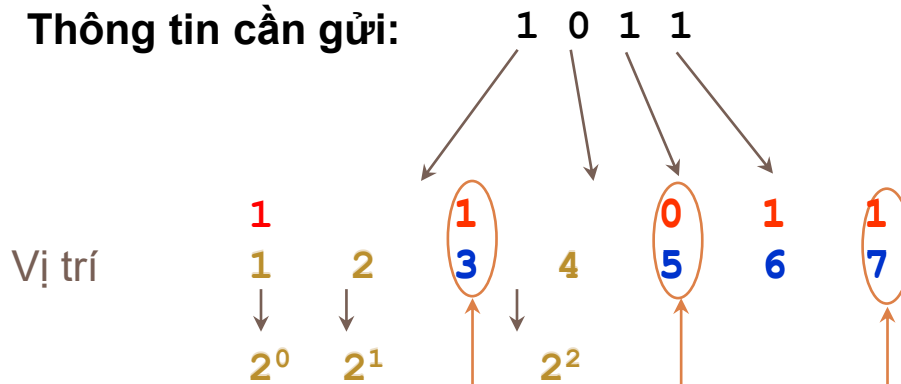


Tính lấy bits:

$$\begin{array}{rclcl} 3 & = & 2^1 + 2^0 & = & 0 \quad 1 \quad 1 \\ 5 & = & 2^2 + 2^0 & = & 1 \quad 0 \quad 1 \\ 6 & = & 2^2 + 2^1 + & = & 1 \quad 1 \quad 0 \\ 7 & = & 2^2 + 2^1 + 2^0 & = & 1 \quad 1 \quad 1 \end{array}$$

HAMMING CODE - 4

Thông tin cần gửi:



lấy bits:

$$\begin{array}{rclclcl}
 3 & = & 2^1 & + & 2^0 & = & 0 & 1 & 1 \\
 5 & = & 2^2 & + & 2^0 & = & 1 & 0 & 1 \\
 6 & = & 2^2 & + & 2^1 & + & 2^0 & = & 1 & 1 & 0 \\
 7 & = & 2^2 & + & 2^1 & + & 2^0 & = & 1 & 1 & 1
 \end{array}$$

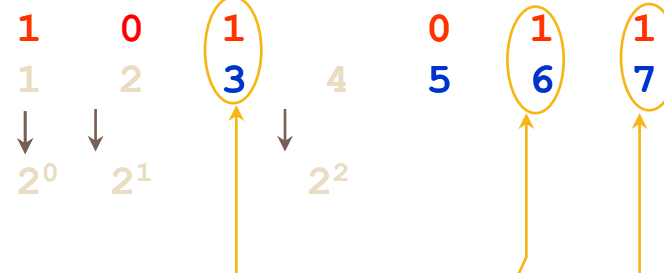
Vị trí 2^0 :

- Xét cột 2^0 trong lấy bit → các vị trí có bit 1
- Lấy các bit DL tại các vị trí có bit 1 trong lấy bit → tính bit parity cho các bit dữ liệu này

HAMMING CODE - 5

Thông tin cần gửi: 1 0 1 1

Vị trí



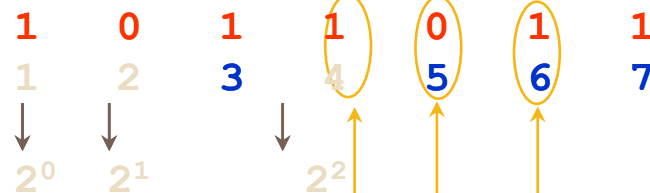
lấy bits:

$$\begin{array}{rclclcl}
 3 & = & 2^1 & + & 2^0 & = & 0 & 1 & 1 \\
 5 & = & 2^2 & + & 2^0 & = & 1 & 0 & 1 \\
 6 & = & 2^2 & + & 2^1 & + & 2^0 & = & 1 & 1 & 0 \\
 7 & = & 2^2 & + & 2^1 & + & 2^0 & = & 1 & 1 & 1
 \end{array}$$

HAMMING CODE - 6

Thông tin cần gửi: 1 0 1 1

Vị trí



lấy bits:

$$\begin{array}{rclclcl}
 3 & = & & 2^1 + 2^0 & = & 0 & 1 & 1 \\
 5 & = & 2^2 & + 2^0 & = & 1 & 0 & 1 \\
 6 & = & 2^2 + 2^1 & + & = & 1 & 1 & 0 \\
 7 & = & 2^2 + 2^1 + 2^0 & = & 1 & 1 & 1
 \end{array}$$

HAMMING CODE - 7

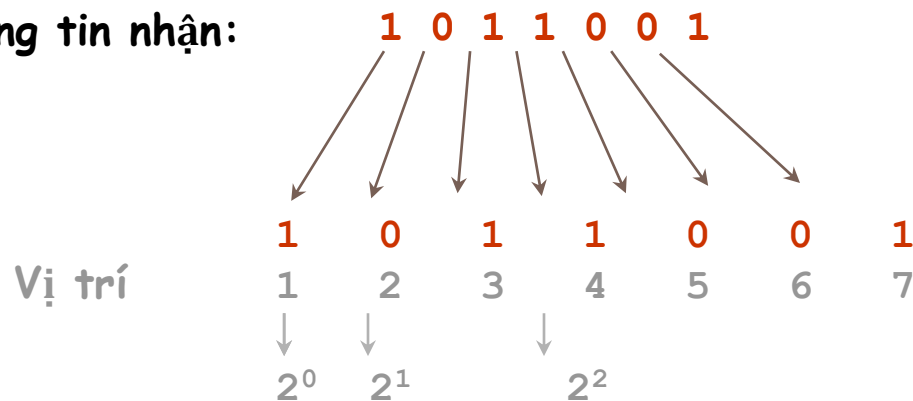
- Dữ liệu cần gửi: 1011
- Dữ liệu gửi: 1011011

HAMMING CODE - 8

- Bên nhận: với mỗi Hamming Code
 - Điền các bit Hamming Code nhận vào các vị trí từ 1 đến M
 - Tính lấy bit
 - Kiểm tra các bit parity
 - Nếu tại bit 2^i phát hiện sai → đánh dấu Error, hệ số $k_i = 1$
 - Ngược lại, đánh dấu No Error = 0, hệ số $k_i = 0$
 - Vị trí bit lỗi: $pos = \sum 2^i \cdot k_i$

HAMMING CODE – 9

Thông tin nhận:



Tính lấy bits:

$$\begin{array}{rclcl} 3 & = & 2^1 + 2^0 & = & 0 \ 1 \ 1 \\ 5 & = & 2^2 + 2^0 & = & 1 \ 0 \ 1 \\ 6 & = & 2^2 + 2^1 & = & 1 \ 1 \ 0 \\ 7 & = & 2^2 + 2^1 + 2^0 & = & 1 \ 1 \ 1 \end{array}$$

HAMMING CODE – 10

Thông tin nhận:

1 0 1 1 0 0 1

Vị trí

1 0 1 1 0 0 1
1 2 3 4 5 6 7

2^0 2^1 2^2

Tính lấy bits:

$$\begin{aligned} 3 &= 2^1 + 2^0 = 0 1 1 \\ 4 &= 2^2 + 2^0 = 1 0 1 \\ 6 &= 2^2 + 2^1 = 1 1 0 \end{aligned}$$

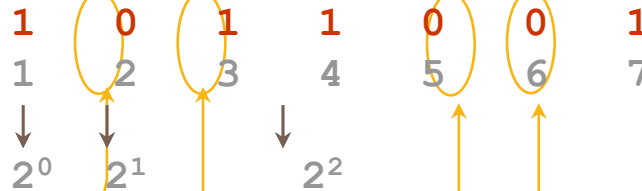
Odd parity: Không có lỗi

HAMMING CODE – 11

Thông tin nhận:

1 0 1 1 0 0 1

Vị trí



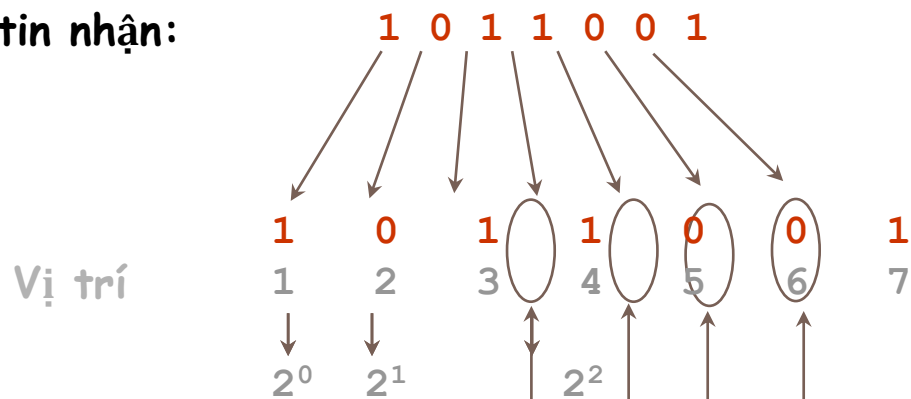
Tính lấy bits:

$$\begin{array}{rclclcl} 3 & = & 2^1 & + & 2^0 & = & 0 & 1 & 1 \\ 5 & = & 2^2 & & + & 2^0 & = & 1 & 0 & 1 \\ 6 & = & 2^2 & + & 2^1 & & = & 1 & 1 & 0 \\ 7 & = & 2^2 & + & 2^1 & + & 2^0 & = & 1 & 1 & 1 \end{array}$$

Odd parity: LỖI

HAMMING CODE – 12

Thông tin nhận:



Tính lấy bits:

| | | | | | | |
|---|---|-------------------|---|---|---|---|
| 3 | = | $2^1 + 2^0$ | = | 0 | 1 | 1 |
| 5 | = | $2^2 + 2^0$ | = | 1 | 0 | 1 |
| 6 | = | $2^2 + 2^1$ | = | 1 | 1 | 0 |
| 7 | = | $2^2 + 2^1 + 2^0$ | = | 1 | 1 | 1 |

Odd parity: LỖI

HAMMING CODE – 13

| | | | | 2^2 | 2^1 | 2^0 | |
|---|---|-------------------|-------------|-------|-------|-------|---|
| 3 | = | | $2^1 + 2^0$ | = | 0 | 1 | 1 |
| 5 | = | 2^2 | $+ 2^0$ | = | 1 | 0 | 1 |
| 6 | = | $2^2 + 2^1$ | | = | 1 | 1 | 0 |
| 7 | = | $2^2 + 2^1 + 2^0$ | | = | 1 | 1 | 1 |

E = error in column
NE = no error in column

| | | |
|---|---|----|
| E | E | NE |
| ↓ | ↓ | ↓ |
| 1 | 1 | 0 |

= 6

➔ Lỗi bit thứ 6 trong Hamming Code

Dữ liệu nhận đúng: 10**11**011

Dữ liệu thật: 1011

HAMMING CODE – CÁCH 2

- **Lưu ý:** số bit Parity ràng buộc theo công thức

$$2^p \geq d + p + 1$$

- p: số bit Parity
- d: số bit data

- **Ví dụ:**

- $d = 7 \Rightarrow p = 4 \Rightarrow M = d + p = 11$
- $d = 4 \Rightarrow p = 4 \Rightarrow M = d + p = 8$
- $M = 8, \log_2 8 = 3 \Rightarrow p = 3 \Rightarrow d = 5$: không thỏa công thức $\Rightarrow p = 4 \Rightarrow d = 4$

HAMMING CODE – CÁCH 2

Tạo từ mã Hamming Code như sau:

- Đánh dấu tất cả vị trí lũy thừa của 2 là bit parity (vị trí 1, 2, 4, 8, 16, 32, 64, ...)
- Tất cả vị trí còn lại là bit dữ liệu sẽ được mã hóa (vị trí 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, ...)
- Giá trị bit Parity được tính dựa vào tính chẵn lẻ của một số bit dữ liệu trong từ mã. Vị trí bit Parity sẽ xác định luân phiên chuỗi các bit được tính hay bỏ qua.

HAMMING CODE – CÁCH 2 (TT)

- Vị trí 1: lấy 1 bit, bỏ qua 1 bit... (1,3,5,7,9,11,13,15,...)
- Vị trí 2: lấy 2 bits, bỏ qua 2 bits... (2,3,6,7,10,11,14,15,...)
- Vị trí 4: lấy 4 bits, bỏ qua 4 bits... (4,5,6,7,12,13,14,15,20,21,22,23,...)
- Vị trí 8: lấy 8 bits, bỏ qua 8 bits... (8-15,24-31,40-47,...)
- Vị trí 16: lấy 16 bits, bỏ qua 16 bits... (16-31,48-63,80-95,...)
- Bit Parity có giá trị 1 khi tổng số bit 1 ở những vị trí nó kiểm tra là số lẻ (odd) và ngược lại.

HAMMING CODE – CÁCH 2 (TT)

○ Ví dụ dữ liệu cần gửi: 1001 1010

- $2^p \geq d + p + 1$
- $d = 8 \Rightarrow P = 4$

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| | | 1 | | 0 | 0 | 1 | | 1 | 0 | 1 | 0 |
| p1 | p2 | | p4 | | | | p8 | | | | |
| | | d3 | | d5 | d6 | d7 | | d9 | d10 | d11 | d12 |

HAMMING CODE – CÁCH 2 (TT)

○ Xác định p1: (lấy 1, bỏ 1)

- vị trí 1,3,5,7,9,11 ,
- giá trị ? 1 0 1 1 1
- 4 số 1 => chẵn => **p1 = 0**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| | | 1 | | 0 | 0 | 1 | | 1 | 0 | 1 | 0 |
| p1 | p2 | | p4 | | | | p8 | | | | |
| | | d3 | | d5 | d6 | d7 | | d9 | d10 | d11 | d12 |

HAMMING CODE – CÁCH 2 (TT)

○ Xác định p2: (lấy 2, bỏ 2)

- vị trí 2,3,6,7,10,11
- giá trị **?1 01 01**
- **3 số 1 => lẻ => p2 = 1**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| 0 | | 1 | | 0 | 0 | 1 | | 1 | 0 | 1 | 0 |
| p1 | p2 | | p4 | | | | p8 | | | | |
| | | d3 | | d5 | d6 | d7 | | d9 | d10 | d11 | d12 |

HAMMING CODE – CÁCH 2 (TT)

○ Xác định p4: (lấy 4, bỏ 4)

- vị trí 4,5,6,7,12
- giá trị **?001 0**
- **1 số 1 => lẻ => p4 = 1**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| 0 | 1 | 1 | | 0 | 0 | 1 | | 1 | 0 | 1 | 0 |
| p1 | p2 | | p4 | | | | p8 | | | | |
| | | d3 | | d5 | d6 | d7 | | d9 | d10 | d11 | d12 |

HAMMING CODE – CÁCH 2 (TT)

○ Xác định p8: (lấy 8, bỏ 8)

- vị trí 8,9,10,11,12
- giá trị **?1010**
- 2 số 1 \Rightarrow chẵn \Rightarrow **p8 = 0**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | | 1 | 0 | 1 | 0 |
| p1 | p2 | | p4 | | | | p8 | | | | |
| | | d3 | | d5 | d6 | d7 | | d9 | d10 | d11 | d12 |

HAMMING CODE – CÁCH 2 (TT)

○ Dữ liệu gửi: 0111 0010 1010

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| p1 | p2 | | p4 | | | | p8 | | | | |
| | | d3 | | d5 | d6 | d7 | | d9 | d10 | d11 | d12 |

HAMMING CODE – CÁCH 2 (TT)

○ Dữ liệu nhận được: 0111 0010 1**1**10

| V | X | | V | | | | X | | | | |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| p1 | p2 | | p4 | | | | p8 | | | | |
| | | d3 | | d5 | d6 | d7 | | d9 | d10 | d11 | d12 |

**p2, p8 sai => vị trí bit dữ liệu sai là $2+8 = 10$
=> d10 sửa 1 thành 0**

Dữ liệu gửi là: 0111 0010 1**0**10

Dữ liệu thật là: 1001 1010