TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TPHCM

Khoa: Công nghệ Điện tử

Môn: Mạng Máy Tính

GV. Ths. Nguyễn Thanh Đăng

A. DATA LINK

- Điều khiển luồng (dòng)
- Phát hiện lỗi
- Xử lý lỗi

Điều khiển luồng

- Là kỹ thuật nhằm đảm bảo rằng bên phát không làm tràn dữ liệu bên nhận
- Hai phương pháp được sử dụng:
  - Phương pháp dừng và chờ (Stop and Wait)
    - Đơn giản nhất,
    - Kém hiệu quả, chỉ có một khung tin được truyền tại một thời điểm
  - Phương pháp cửa sổ trượt –(Sliding Window Flow Control)
    - Hiệu quả
    - Cho phép truyền nhiều khung tin cùng một lúc trên kênh truyền

Phương pháp dừng và chờ

- Truyền một gói tin và chờ báo nhận
  - Bên phát truyền một khung tin
  - Sau khi nhận được khung tin, bên nhận gửi lại xác nhận
  - Bên phát phải đợi đến khi nhận được xác nhận thì mới truyền khung tin tiếp theo
- Không hiệu quả
  - Bên nhận có thể dừng quá trình truyền bằng cách không gửi khung tin xác nhận
  - Tại một thời điểm chỉ có một khung tin trên đường truyền → chậm
  - Trường hợp độ rộng của kênh truyền lớn hơn độ rộng của khung tin thì nó tỏ ra cực kỳ kém hiệu quả.

Phương pháp cửa sổ trượt

- Cho phép nhiều khung tin được truyền tại một thời điểm ->Truyền thông hiệu quả hơn.
- A và B được kết nối trực tiếp song công (full-duplex).
- B có bộ đệm cho n khung tin -> B có thể chấp nhận n khung tin, A có thể truyền n khung tin mà không cần đợi xác nhận từ bên B
- Mỗi khung tin được gán nhãn bởi một số thứ tự.
- B xác nhận khung tin đã được nhận bằng cách gửi xác nhận cùng với số thứ tự của khung tin tiếp theo mà nó mong muốn nhận

Phương pháp cửa sổ trượt

- A duy trì danh sách các số thứ tự được phép gửi
- B duy trì danh sách số thứ tự chuẩn bị nhận

  - Gọi là cửa sổ của các khung tin

  - Điều khiển dòng cửa sổ trượt

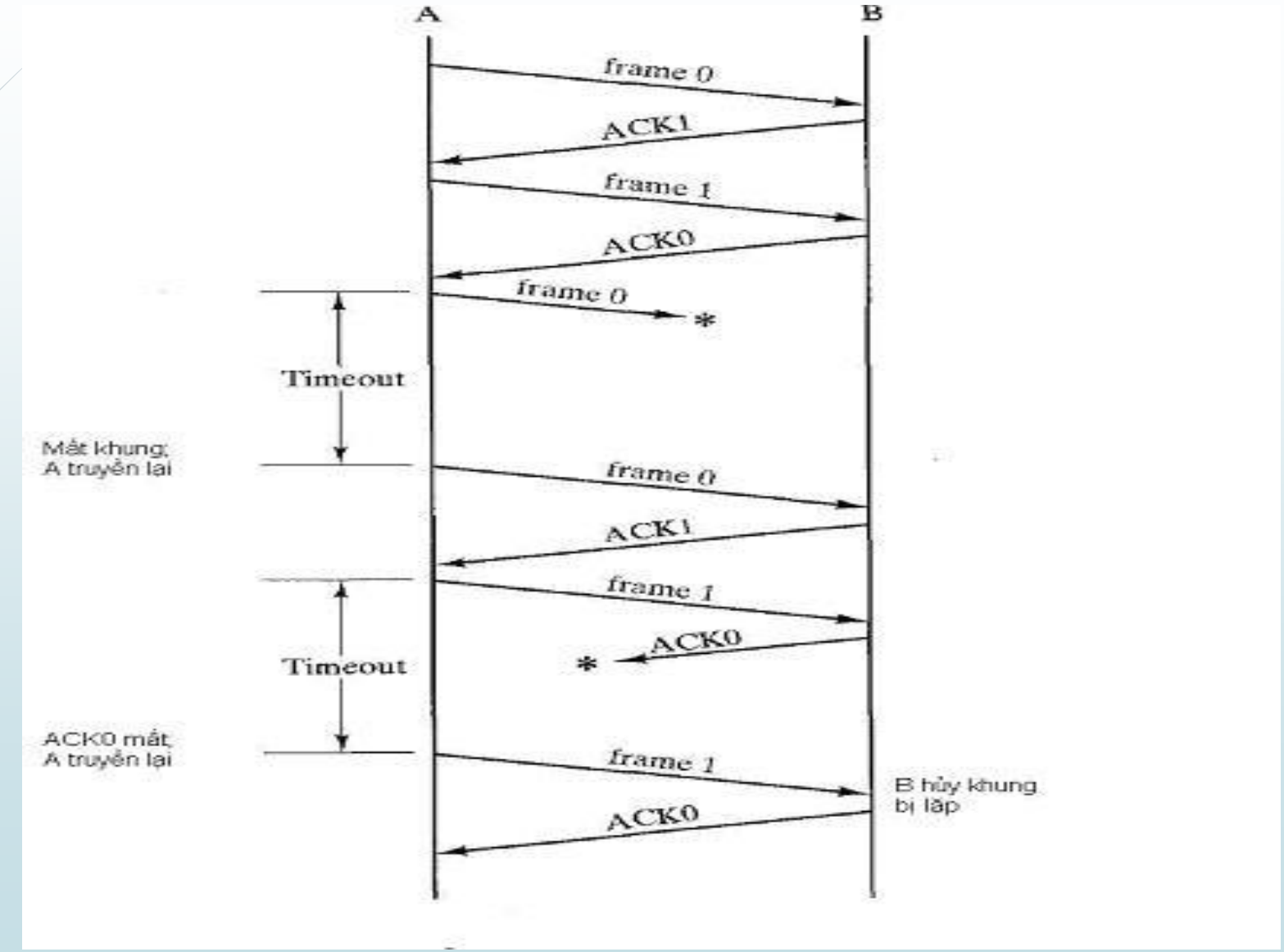
Phương pháp cửa sổ trượt

- Đối với đường truyền 2 chiều thì mỗi bên phải sử dụng hai cửa sổ:
  - Một cho phát và một cho nhận
  - Mỗi bên đều phải gửi dữ liệu và gửi xác nhận tới bên kia
- Số thứ tự được lưu trữ trong khung tin
  - Bị giới hạn, trường k bit thì số thứ tự được đánh số theo Module của 2^k
  - Kích thước của cửa sổ không nhất thiết phải lấy là maximum (ví dụ trường 3 bit, có thể lấy độ dài cửa sổ là 4)

Xử lý lỗi: ARQ dừng và chờ

- Trên cơ sở kĩ thuật điều khiển luồng dừng-và-chờ
- Kiểm soát lỗi:
  - Khung tin tới bên nhận bị hỏng: Truyền lại, sử dụng đồng hồ đếm giờ time-out
  - Báo nhận bị hỏng: Time-out, bên phát gửi lại, sử dụng label 0/1 và ACK0/ACK1 phát hiện lỗi

Xử lý lỗi: ARQ dừng và chờ



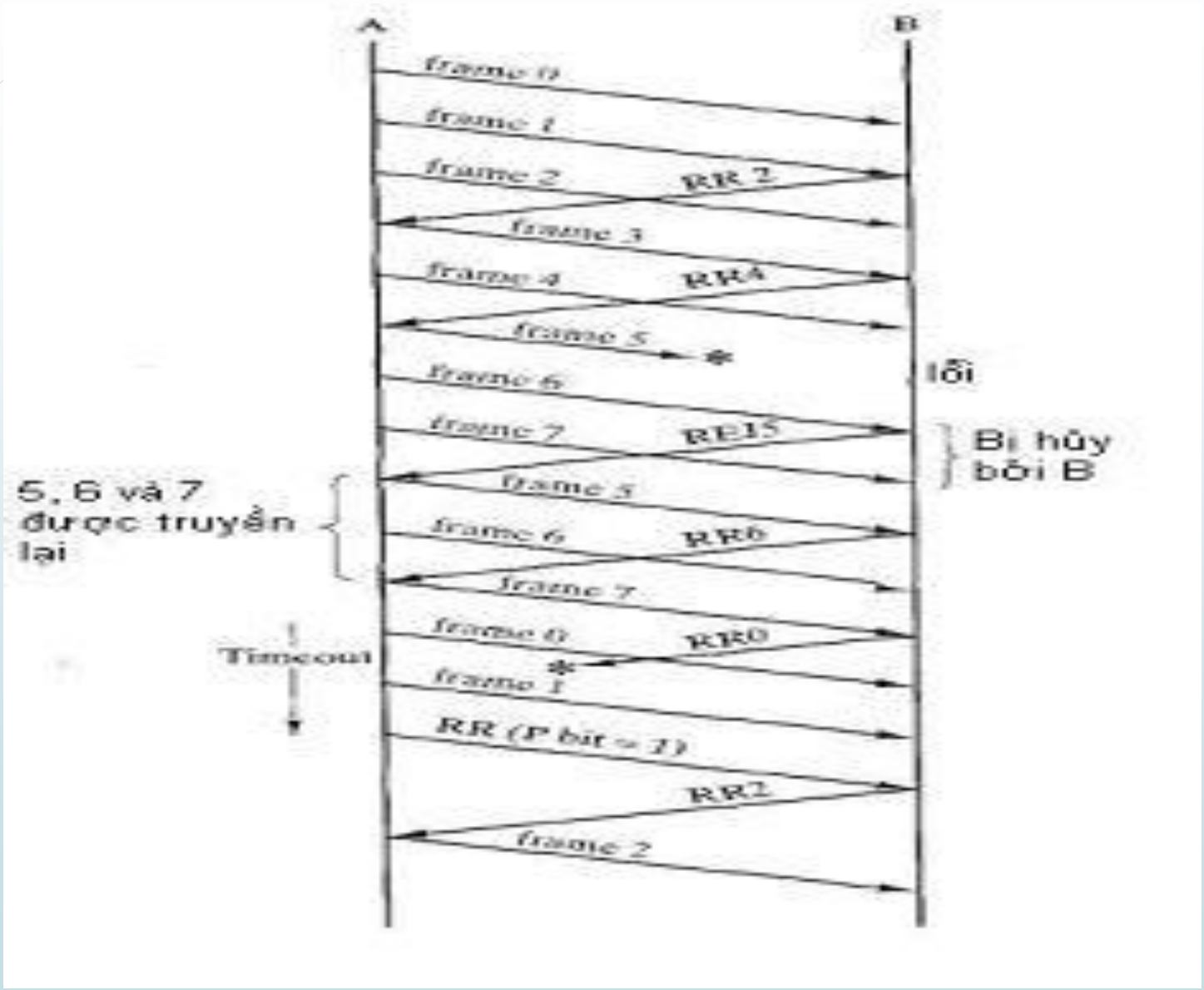
Xử lý lỗi: ARQ Quay-lùi-N

- Trên cơ sở kĩ thuật điều khiển luồng bằng Cửa sổ trượt
- Kiểm soát lỗi:
  - Khung hỏng:
    - Khung i-1 thành công, i lỗi, bên nhận gửi SREJ i, bên phát gửi lại
    - Khung i mất, i+1 được nhận không đúng trình tự, REJ i, bên gửi phát lại i và các khung sau đó
    - Chỉ khung i được truyền và bị mất, bên nhận không biết i đã được truyền đi, bên phát gửi time-out và gửi RR với P=1, khi bên phát nhận được RR từ bên nhận nó sẽ phát lại i

Xử lý lỗi: ARQ Quay-lui-N

- RR hỏng:
  - B nhận khung i và gửi RR(i+1), RR(i+1) mất, A có thể nhận RR(>i+1) trước khi RR(i+1) time-out, và có nghĩa là khung i đã thành công.
  - RR(i+1) time-out, A cố gắng gửi RR với P-bit cho đến khi nhận được RR từ B một số lần nhất định, nếu vẫn không nhận được thì Khởi động lại giao thức
- Reject hỏng:
  - A time-out, A gửi RR với P=1 cho đến khi nhận được RRi từ B thì A sẽ gửi lại khung i

Xử lí lỗi: ARQ Quay-lui-N



Xử lý lỗi: ARQ Chọn-Hủy (Selective-Reject)

- Chỉ truyền lại những khung có báo nhận là lỗi (SREJ)
- Phải duy trì đủ bộ đệm đủ lớn
- Đảm bảo tính logic phức tạp để gửi và nhận các khung theo đúng trình tự.
- ARQ Chọn-Hủy phải giải quyết được sự chồng chéo giữa cửa sổ gửi và nhận.

Xử lý lỗi: ARQ Chọn-Hủy (Selective-Reject)

- Trạm A gửi các khung từ 0 đến 6 tới trạm B.
- Trạm B nhận tất cả 7 khung và báo nhận tích lũy với RR 7
- Vì lí do nào đó ví dụ như nhiễu làm RR 7 bị mất trên đường truyền.
- Đồng hồ ở A hết hạn và A truyền lại khung 0.
- B đã điều chỉnh trước cửa sổ nhận để có thể nhận các khung 7, 0, 1, 2, 3, 4 và 5. Do đó mà khung 7 được coi là bị mất và khung nhận được này là khung số 0 mới, và được chấp nhận bởi B.

Giới thiệu Phương Pháp CRC

- CRC là một phương pháp để phát hiện lỗi bằng cách gắn thêm một khối bit phía sau khối dữ liệu .
- Thuật toán để tạo ra khối bit CRC là dựa trên phép cộng cơ số 2 .
- CRC là phần dư của phép chia nhị phân không nhớ .

Phát hiện lỗi

- Lý do một hay nhiều bit thay đổi trong khung tin được truyền:
  - Tín hiệu trên đường truyền bị suy yếu
  - Tốc độ truyền
  - Mất đồng bộ
- Việc phát hiện ra lỗi để khắc phục, yêu cầu phát lại là cần thiết và vô cùng quan trọng trong truyền dữ liệu.

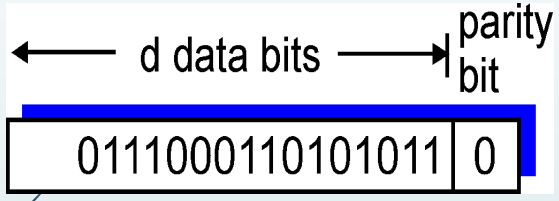
Phát hiện lỗi: Parity Check

- Là kỹ thuật đơn giản nhất.
- Đưa một bit kiểm tra tính chẵn lẻ vào sau khối tin.
- Giá trị của bit này được xác định dựa trên số các số 1 là chẵn (even parity), hoặc số các số 1 là lẻ (odd parity).
- Lỗi sẽ không bị phát hiện nếu trong khung tin có 2 hoặc một số chẵn các bit bị đảo.
- Không hiệu quả khi xung nhiễu đủ mạnh.

Kiểm tra Parity

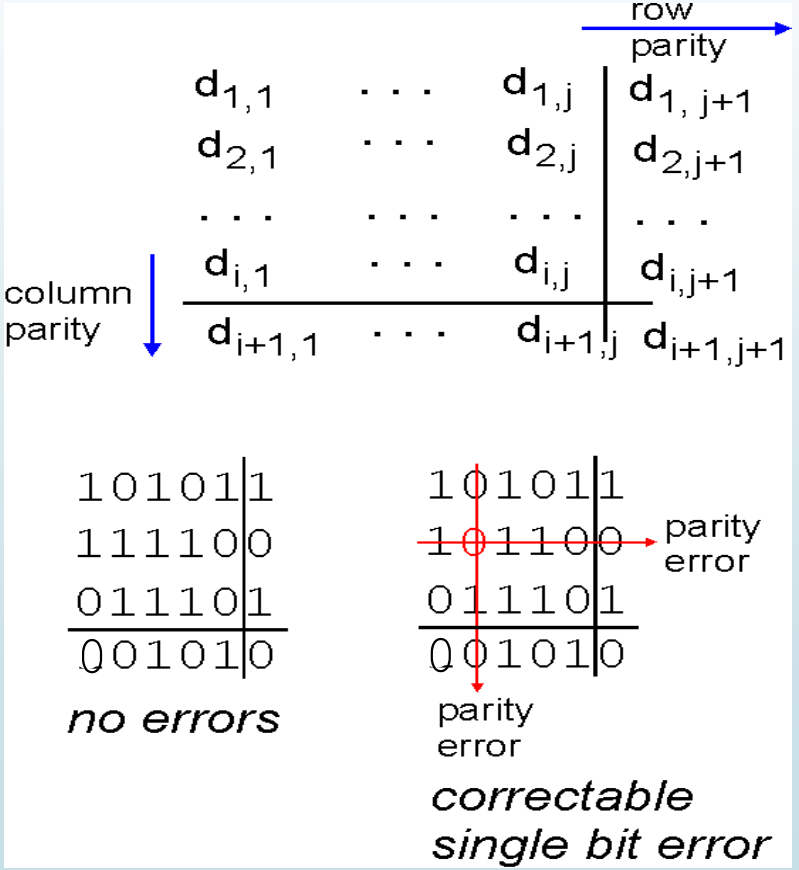
Bit Parity đơn:

phát hiện các lỗi bit



Bit Parity 2 chiều:

phát hiện & sửa các lỗi bit



Phát hiện lỗi: Cyclic redundancy Check (CRC)

Giá trị chuỗi bit kiểm tra hay chuỗi CRC là số dư của phép chia của chuỗi bit dữ liệu cho một chuỗi bit đa thức sinh (Generator Polynomial).

Đa thức sinh là số chia sẽ khác nhau tùy vào mỗi giao thức quy định.

Phép chia trong tính toán CRC sử dụng cách tính modulo-2. (Modulo-2 thực chất là XOR hai số hạng).

Giả sử đa thức chuỗi dữ liệu cần truyền là M(x):

$$M(x) = a_m.x^m + a_{m-1}.x^{m-1} + a_{m-2}.x^{m-2} + \cdots + a_1.x^1 + a_0$$

Đa thức sinh là G(x):

$$G(x) = a_n.x^n + a_{n-1}.x^{n-1} + a_{n-2}.x^{n-2} + \cdots + a_1.x^1 + a_0$$

Trong đó:

- a_m và a_n bằng 1 hoặc 0
- Độ dài chuỗi CRC bằng độ dài đa thức sinh trừ 1 và bằng số mũ lớn nhất của đa thức sinh và bằng n.

- Để tạo CRC, chuỗi dữ liệu cần truyền sẽ được mở rộng thêm n bit về phía bên phải:

$$T(x) = a_n . x^n . M(x)$$

Điều này, tương ứng với việc dịch trái n bit chuỗi dữ liệu M(x)

Cuối cùng, chia T(x) cho G(x) và lấy số dư. Số dư chính là chuỗi CRC n bit.

$$\boxed{R(x) = T(x) \,\%\, G(x)}$$

•Kiểm tra CRC được thực hiện bằng 1 trong 2 cách sau:
Lấy chuỗi dữ liệu có cả các bit kiểm tra CRC chia cho đa thức sinh. Nếu số dư khác "0" thì dữ liệu nhận bị lỗi.

•Tách chuỗi dữ liệu và chuỗi CRC riêng. Chỉ lấy chuỗi dữ liệu chia cho đa thức sinh rồi lấy số dư phép chia so sánh với chuỗi CRC. Nếu hai chuỗi khác nhau thì dữ liệu nhận bị lỗi.

Chuyển đổi giữa chuỗi số nhị phân và đa thức

Với số bit : m

Ta được chuỗi bit : b

Đa thức nhị phân biểu diễn chuỗi bit trên là :

$$b_{(m-1)}.x^{(m-1)} + b_{(m-2)}.x^{(m-2)} + \ldots + b_2.x^2 + b_1.x^1 + b_0.x^0$$

Ví dụ : Với chuỗi bit 101 có thể biểu diễn dưới dạng đa thức nhị phân sau:

$$1.x^2 + 0.x^1 + 1.x^0 = x^2 + 1$$

Một số đa thức sinh

CRC-n	G(x)	USE
CRC-1	$x+1$	Hardware (parity bit)
CRC-4	$x^4 + x + 1$	PCM-30
CRC-5 - CCITT	$x^5 + x^3 + x + 1$	ITU-G.704
CRC-5 – USB	$x^5 + x^2 + 1$	USB token packets
CRC-7	$x^7 + x^3 + 1$	Telecom systems, MMC
CRC-8	$x^8 + x^7 + x^6 + x^4 + x^2 + 1$	
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$	use: telecom systems
CRC-32 - MPEG2	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	
…		

Cách xác định CRC-r

_Chuyển dữ liệu chuyển đi thành số nhị phân (D) .

_Chọn mẫu r + 1 bit (G) (G còn được gọi là đa thức sinh bậc r) .

Chọn r bit CRC , R , như thế :

+ Lấy D * 2 rồi XOR cho G .

+ Ta được phần dư R (CRC)

+ Bên nhận tiếp tục lấy D + R rồi chia lại cho G nếu phần dư khác 0 -> phát hiện lỗi .

+ Có thể kiểm tra tất cả các lỗi nhỏ hơn r + 1 bits

_Sử dụng phổ biến trong thực tế :

+ ATM : CRC-8-ATM $x^8 + x^2 + x + 1$ (ATM HEC)

+ HDLC : CRC-64-ISO $x^{64} + x^4 + x^3 + x + 1$ (HDLC – ISO 3309)

Phát hiện lỗi: Các bước tạo và kiểm tra CRC

- Ví dụ về tính toán CRC-4, tương ứng với số bit kiểm tra là 4 bit, với đa thức sinh như sau:

  x^4 + x + 1 (b10011)

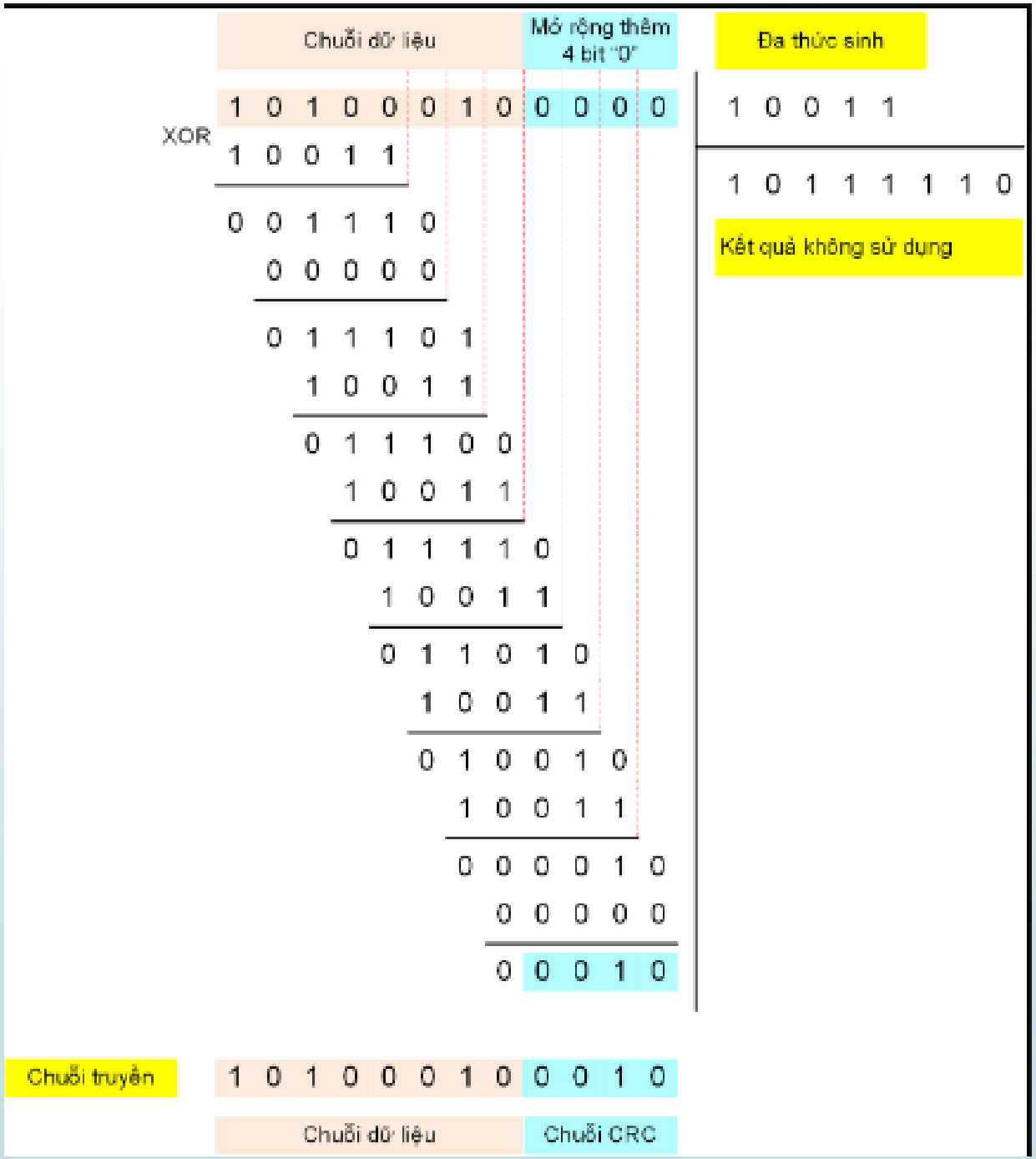
  Chuỗi dữ liệu cần truyền có 8 bit như sau:

  x^7 + x^5 + x (b1010_0010)

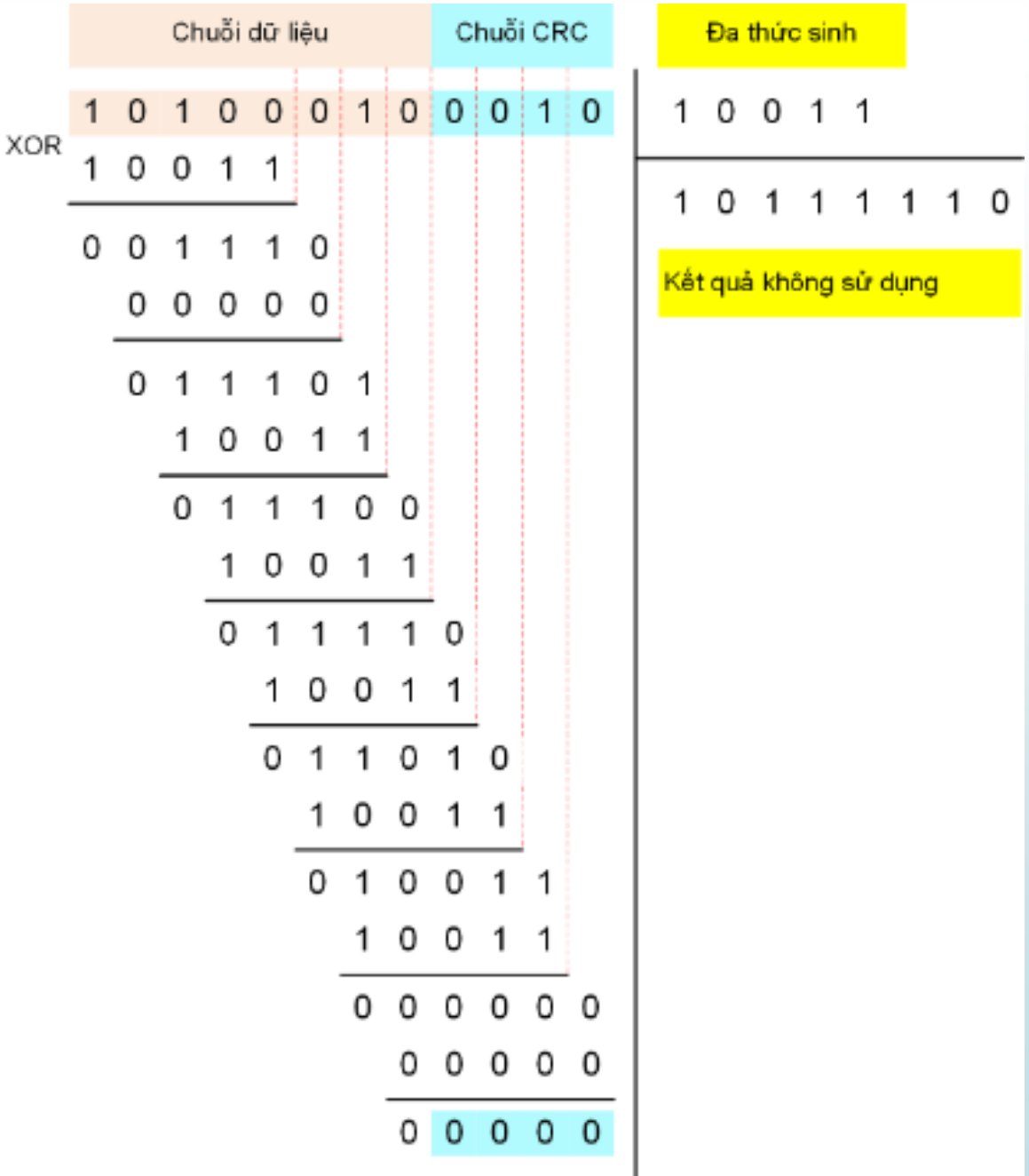
- Chuỗi dữ liệu trước khi chia sẽ được mở rộng thêm 4 bit "0": (do CRC_4)

  x^11 + x^9 + x^5 (b1010_0010_0000)

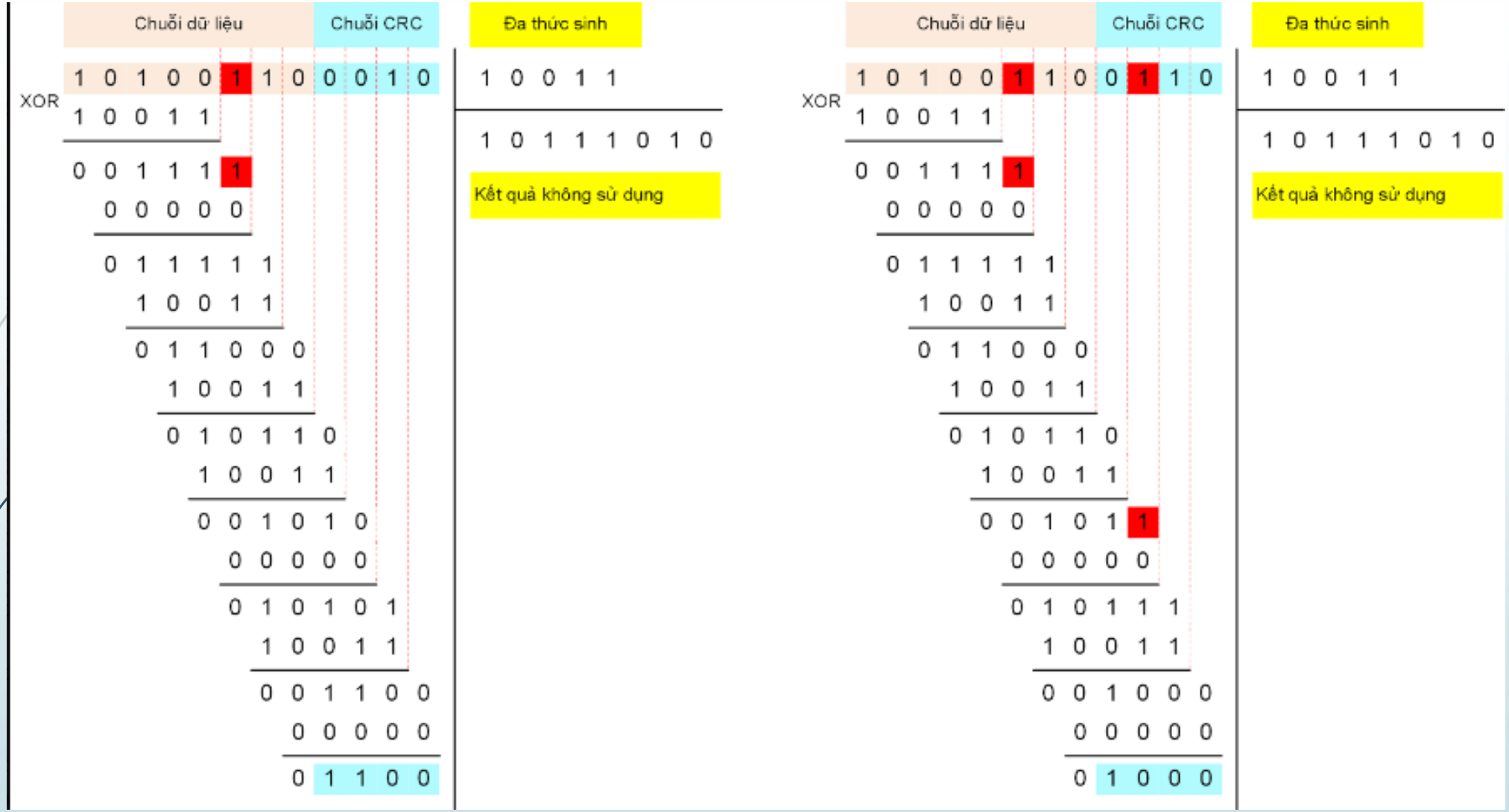
Tính chuỗi CRC



Kiểm tra CRC được thực hiện trên chuỗi dữ liệu có đính kèm CRC như sau:



Kiểm tra CRC
- Trường hợp nhận đúng, số dư bằng 0

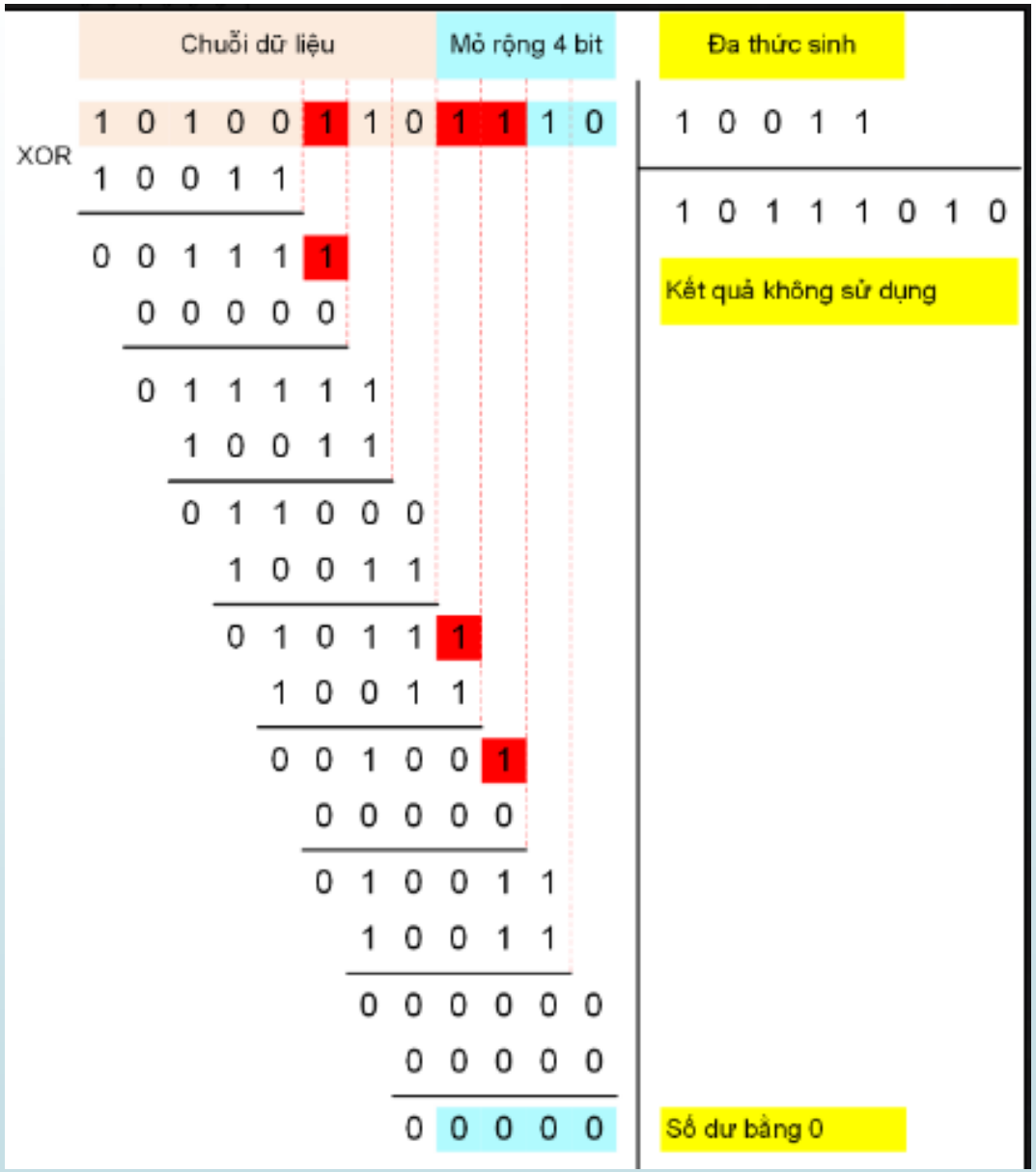


Kiểm tra CRC bằng cách chia chuỗi dữ liệu có CRC với đa thức sinh

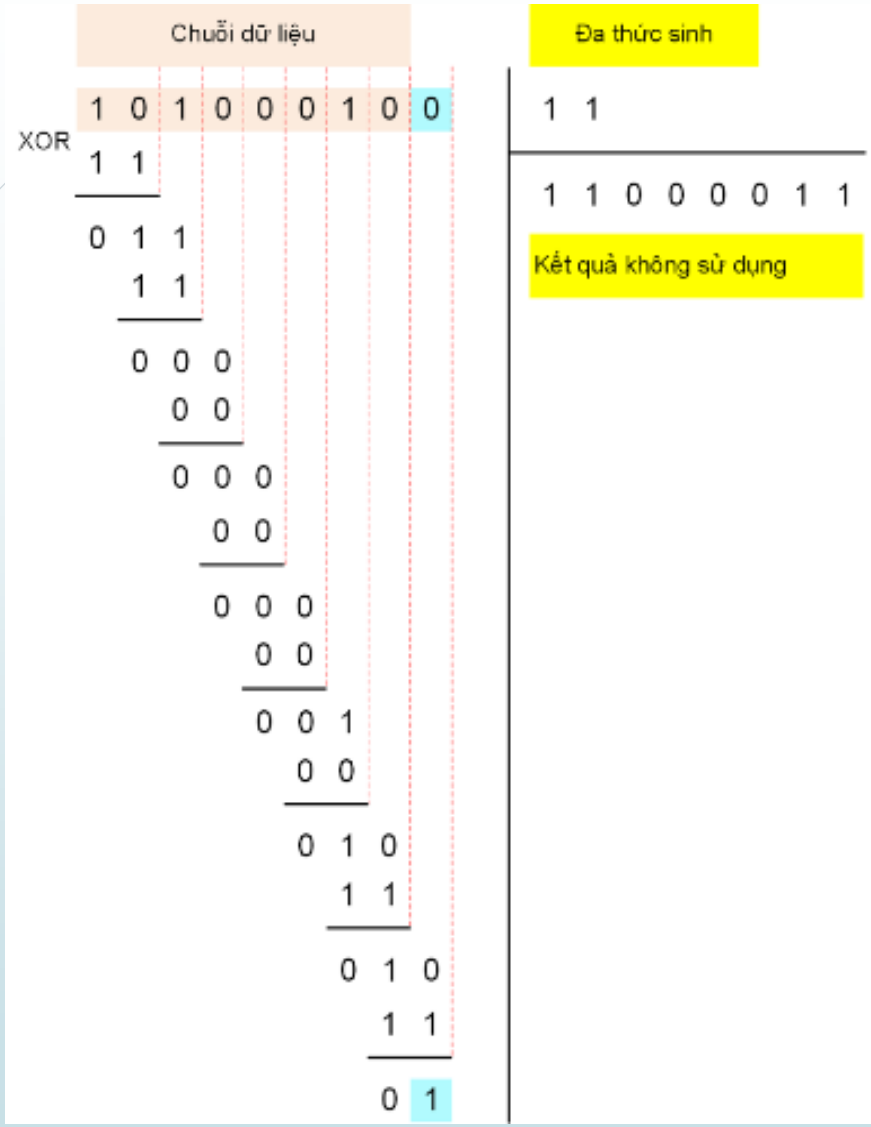
- Trường hợp sai 1 bit và trường hợp sai 2 bit, số dư khác 0

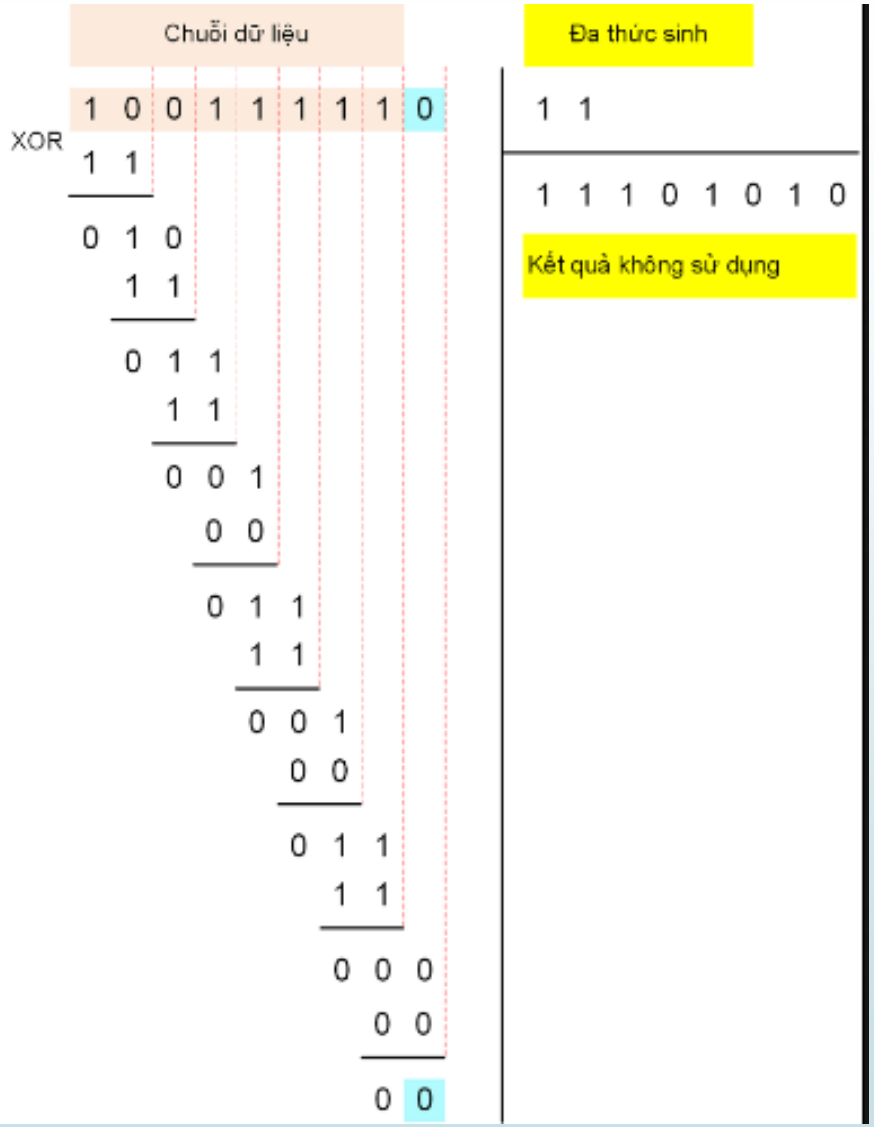
Bộ nhận sẽ không phát hiện được lỗi dữ liệu khi chuỗi dữ liệu bị sai và chuỗi CRC cũng sai trùng với giá trị CRC của chuỗi dữ liệu bị sai. Tuy nhiên, xác suất để xảy ra đúng trường hợp này là thấp. Xác suất này càng thấp khi chuỗi CRC càng dài.

Kiểm tra CRC không phát hiện được lỗi khi cả dữ liệu và CRC cùng sai đồng thời



Ví dụ: Xét đa thức sinh g(x) = x + 1(11_2), đây là đa thức CRC-1, tính CRC cho chuỗi 8 bit b10100010 và chuỗi b10011111.





So sánh kết quả với phương pháp tính parity chẵn đã trình bày phía trên chúng ta có thể nhận thấy sự tương đồng. CRC-1 chính là phương pháp kiểm tra parity.

CRC ví dụ

1.Xác định mã CRC dùng thuật toán Mod 2

_r là bậc cao nhất của G (ở đây r = 3)

$_2^r = 2^3 = 8 = 1000_{(2)}$

$_D*2^r$: 1 0 1 1 1 0

1 0 0 0

1 0 1 1 1 0 0 0 0

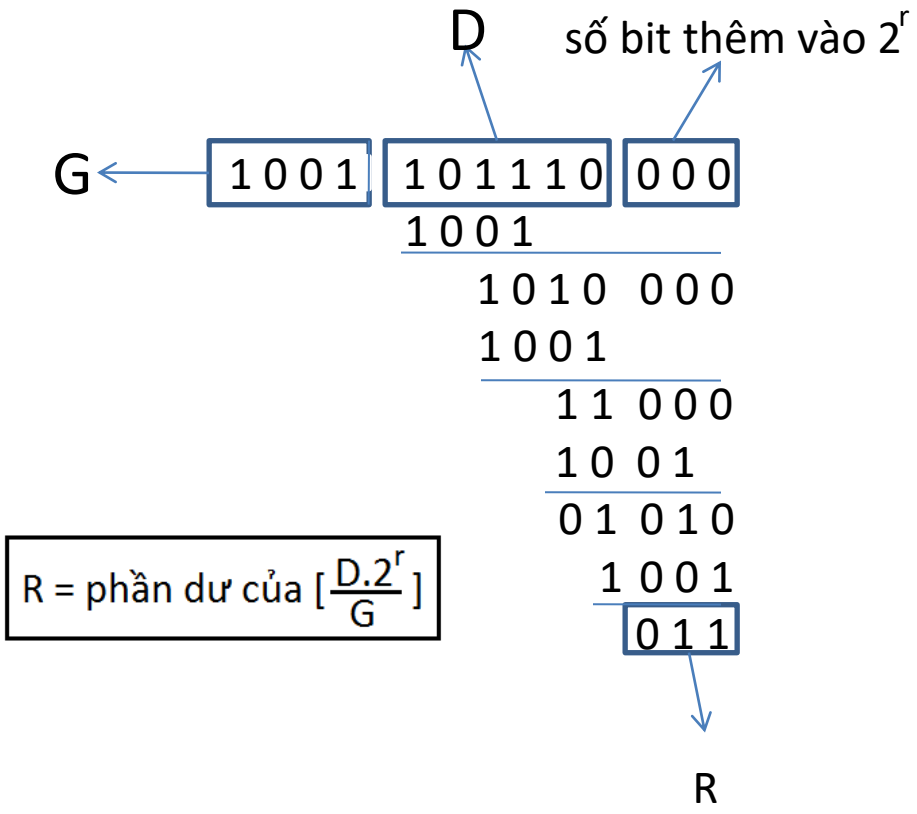
_Lấy ($D*x^r$)/ G số dư cuối cùng là R

Nếu dãy nhị phân đầu vào bên trên có bit đầu tiên bên trái là 0, không làm gì hết và dịch số chia sang phải một bit.
Nếu dãy nhị phân đầu vào bên trên có bít đầu tiên bên trái là 1, lấy dãy số đầu vào trừ đi số chia.

Phép trừ Mod 2 là phép trừ nhị phân không nhớ nên ta sẽ có :

1 - 1 = 0
0 - 0 = 0
1 - 0 = 1
0 - 1 = 1

Tìm CRC với mẫu G = 1001 của chuỗi nhị phân D= 101110:



CRC ví dụ

1.Dùng phép biểu diễn đa thức

_Lấy lại ví dụ trên : Tìm CRC của chuỗi nhị phân D= 101110 với mẫu G = 1001

_Chuyển G và D sang đa thức ta sẽ có

$G(x) = x^3 + 1$

$D(x) = x^5 + x^3 + x^2 + x$

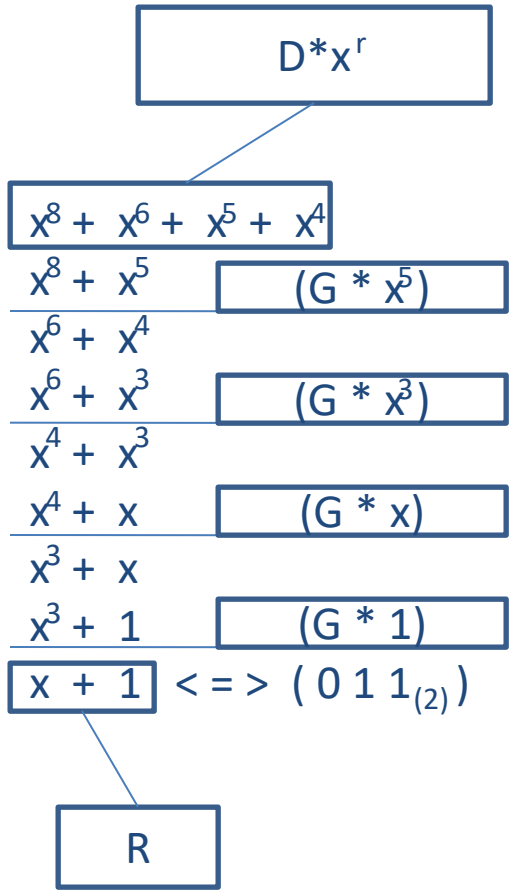
_Nhân D cho x^r

$x^5 + x^3 + x^2 + x$

x^3

$x^8 + x^6 + x^5 + x^4$

_Lấy ($D*x^r$)/ G số dư cuối cùng là R



Bài Tập : Tìm CRC-4 của chuỗi số nhị phân sau : D=1101001010101010, G= x^3+x+1

THANKS FOR LISTENING!

Bài Tập : Tìm CRC-4 của chuỗi số nhị phân sau :

D=11010010101010, G= x^3+x+1