TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THÔNG TIN TP. HCM
CS LK TRƯỜNG TC CÔNG NGHỆ TIN HỌC – VIỄN THÔNG ĐỒNG NAI
KHOA CN TIN HỌC – VIỄN THÔNG

---

# BÀI TIỂU LUẬN

*Ngành: Điện Tử Viễn Thông. Hệ: Cao Đẳng Liên Thông*
*Lớp: C05KTLT. Niên Khóa: 2010 – 2012.*

## *Đề tài:* TÌM HIỂU CRC

*GIÁO VIÊN LỮ XUÂN TRANG*
*NHÓM SVTH: 1- PHẠM PHÚ ĐỨC*
*2- VŨ CÔNG HOAN*
*3- NGUYỄN THỊ HƯƠNG*
*4- TRẦN THỊ HƯƠNG*
*5- VÕ THỊ GIAO LINH*
*6- LƯU TUYẾT NHUNG*

*Tháng 01 năm 2012*

# NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Biên Hòa, ngày ..... tháng ..... năm 2012*

**Giáo Viên**

**Lữ Xuân Trang**

# MỤC LỤC

LỜI NÓI ĐẦU ..... 1
I. TỔNG QUAN VỀ CRC ..... 2
1. Khái niệm ..... 2
2. Ứng dụng ..... 2
3. Cách tính toán các bit CRC ..... 2
4. Những hàm CRC thường dùng và được tiêu chuẩn hóa ..... 3
II. TÌM HIỂU CRC - 4 ..... 7
1. CRC - 4 trong PCM30 ..... 7
2. Tính toán CRC-4 ..... 8
III. TÍNH TOÁN CRC - 8 ..... 10
TÀI LIỆU THAM KHẢO ..... 12

# LỜI NÓI ĐẦU

CRC (*cyclic redundancy check*) là một loại hàm băm, được dùng để sinh ra giá trị kiểm thử, của một chuỗi bit có chiều dài ngắn và cố định, của các gói tin vận chuyển qua mạng hay một khối nhỏ của tệp dữ liệu. Giá trị kiểm thử được dùng để dò lỗi khi dữ liệu được truyền hay lưu vào thiết bị lưu trữ. Giá trị của CRC sẽ được tính toán và đính kèm vào dữ liệu trước khi dữ liệu được truyền đi hay lưu trữ. Khi dữ liệu được sử dụng, nó sẽ được kiểm thử bằng cách sinh ra mã CRC và so khớp với mã CRC trong dữ liệu.

CRC rất phổ biến, vì nó rất đơn giản để lắp đặt trong các máy tính sử dụng hệ cơ số nhị phân, dễ dàng phân tích tính đúng, và rất phù hợp để dò các lỗi gây ra bởi nhiễu trong khi truyền dữ liệu.

Với thời gian có hạn và thiết xót về kiến thức của nhóm.Nếu có gì thiếu xót xin cô và các bạn góp ý. Xin chân thành cảm ơn!

**Nhóm SVTH**

**Nhóm 1 – Lớp C05KTLT**

# I. TỔNG QUAN VỀ CRC

## 1. Khái niệm

CRC là một loại mã phát hiện lỗi. Cách tính toán của nó giống như phép toán chia số dài trong đó thương số được loại bỏ và số dư là kết quả, điểm khác biệt ở đây là sử dụng cách tính không nhớ.

Độ dài của số dư luôn nhỏ hơn hoặc bằng độ dài của số chia, do đó số chia sẽ quyết định độ dài có thể của kết quả trả về.

Định nghĩa đối với từng loại CRC đặc thù quyết định số chia nào được sử dụng, cũng như nhiều ràng buộc khác

## 2. Ứng dụng

Một thiết bị CRC cho phép tính toán một chuỗi nhị phân ngắn có độ dài cố định, được gọi là giá trị kiểm tra hoặc không đúng Công ước, đối với từng khối dữ liệu được gửi hoặc được lưu trữ và gắn thêm vào các dữ liệu, tạo thành một từ mã.

Khi một từ mã nhận được hoặc đọc, thiết bị, hoặc so sánh giá trị kiểm tra của nó. mới tính từ các khối dữ liệu, hoặc tương đương, thực hiện một ước Quốc tế Quyền Trẻ em trên toàn từ mã và so sánh giá trị kiểm tra kết quả với một hằng số dư dự kiến.

Nếu các giá trị kiểm tra không phù hợp, sau đó khối chứa một lỗi dữ liệu và các thiết bị có thể có hành động khắc phục như đọc lại hoặc yêu cầu ngăn chặn được gửi một lần nữa, nếu không dữ liệu được giả định là lỗi (tuy nhiên, với một số xác suất nhỏ , nó có thể chứa các lỗi bị phát hiện, đây là bản chất cơ bản của kiểm tra lỗi).

## 3. Cách tính toán các bit CRC

CRC là số dư của phép chia đặc biệt và được diễn giải như sau:

+ Muốn tính CRC trong đa khung con n phải căn cứ vào đa khung con n-1.

+ Tổng số các bít thông tin trong đa khung con (n-1) (không tính các bit CRC trong đa khung con này) được xem như một số nhị phân lớn.

+ Số nhị phân này được triển khai thành một đa thức, gọi là đa thức đặc trưng.

+ Bậc của đa thực đặc trưng ít hơn số lượng bit trong số nhị phân một đơn vị, số lượng số hạng của đa thức băng số lượng các bit 1 trong số nhị phân.

+ Trước hết nhân đa thức đặc trưng với $x^m$ , m = số bit trong từ mã CRC.

+ Sau đó đêm tích số này chia (modulo 2) cho đa thức sinh và số dư của phép chia chính là giá trị các bit trong từ mã CRC của đa khung con n.

+ Đa thức sinh là một số nhị phân bé và được chọn một cách hợp lý.

Phía thu tiến hành giải mã CRC như sau:

Tiếp nhận đa khung con n-1, thay các bit CRC trong khung con n-1= các bit 0, triển khai nội dung của đa khung con này thành đa thức đặc trưng. Nhân đa thức đặc trưng với $x^m$ , chia tích số này cho đa thức sinh.

***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 | $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ | |
| ... | | |

Số dư của phép chia được ghi lại và so sách với bit tương ứng của CRC trong đa khung con n.

Nếu các bit của phân dư trong đa khung con n-1 trùng hợp với các bit trong CRC thuộc đa khung con n thì đa khung con n-1 không lỗi.

**4. Những hàm CRC thường dùng và được tiêu chuẩn hóa**

Các dạng mã kiểm soát lỗi CRC (*cyclic redundancy check*) được chia thành nhiều tiêu chuẩn, chúng không được tiêu chuẩn hóa thống nhất cho 1 thuật toán nào ở mỗi mức độ trên toàn cầu, các đa thức thường được xem như không phải là tối ưu nhất có thể.

Bảng dưới đây chỉ liệt kê những đa thức của những thuật toán đa dạng đang được sử dụng. Bất kỳ giao thức cá biệt.

| Tên | Đa thức | Các biểu diễn: thông thường hoặc nghịch đảo (đảo của đảo) |
|---|---|---|
| CRC-1 | $x + 1$ (hầu hết phần cứng; còn biết với tên *parity bit*) | 0x1 or 0x1 (0x1) |
| CRC-4-ITU | $x^4 + x + 1$ (ITU G.704, p. 12) | 0x3 or 0xC (0x9) |
| CRC-5-ITU | $x^5 + x^4 + x^2 + 1$ (ITU G.704, p. 9) | 0x15 or 0x15 (0x1A) |
| CRC-5-USB | $x^5 + x^2 + 1$ (USB token packets) | 0x05 or 0x14 (0x12) |
| CRC-6-ITU | $x^6 + x + 1$ (ITU G.704, p. 3) | 0x03 or 0x30 (0x21) |
| CRC-7 | $x^7 + x^3 + 1$ (Các hệ thống viễn thông, MMC,SD) | 0x09 or 0x48 (0x44) |
| CRC-8-ATM | $x^8 + x^2 + x + 1$ (ATM HEC) | 0x07 or 0xE0 (0x83) |
| CRC-8-CCITT | $x^8 + x^7 + x^3 + x^2 + 1$ (1-Wire bus) | 0x8D or 0xB1 (0xC6) |
| CRC-8-Dallas/Maxim | $x^8 + x^5 + x^4 + 1$ (1-Wire bus) | 0x31 or 0x8C (0x98) |
| CRC-8 | $x^8 + x^7 + x^6 + x^4 + x^2 + 1$ | 0xD5 or 0xAB (0xEA) |
| CRC-8-SAE | $x^8 + x^4 + x^3 + x^2 + 1$ | 0x1D or 0xB8 |

| | | |
|---|---|---|
| J1850 | | (0x8E) |
| CRC-10 | $x^{10} + x^9 + x^5 + x^4 + x + 1$ | 0x233 or 0x331 (0x319) |
| CRC-11 | $x^{11} + x^9 + x^8 + x^7 + x^2 + 1$ (FlexRay) | 0x385 or 0x50E (0x5C2) |
| CRC-12 | $x^{12} + x^{11} + x^3 + x^2 + x + 1$ (Các hệ thống viễn thông ) | 0x80F or 0xF01 (0xC07) |
| CRC-15-CAN | $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$ | 0x4599 or 0x4CD1 (0x62CC) |
| CRC-16-Fletcher | Không phải một CRC; xem Fletcher's checksum | Sử dụng trong Adler-32 A & B CRC |
| CRC-16-CCITT | $x^{16} + x^{12} + x^5 + 1$ (X.25, V.41, CDMA, Bluetooth, XMODEM, HDLC,PPP, IrDA, BACnet; known as *CRC-CCITT*, MMC,SD) | 0x1021 or 0x8408 (0x8810) |
| CRC-16-DNP | $x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$ (DNP, IEC 870, M-Bus) | 0x3D65 or 0xA6BC (0x9EB2) |
| CRC-16-IBM | $x^{16} + x^{15} + x^2 + 1$ (SDLC, USB, khác; còn được biết là *CRC-16*) | 0x8005 or 0xA001 (0xC002) |
| CRC-24-Radix-64 | $x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$ (FlexRay) | 0x864CFB or 0xDF3261 (0xC3267D) |
| CRC-30 | $x^{30} + x^{29} + x^{21} + x^{20} + x^{15} + x^{13} + x^{12} + x^{11} + x^8 + x^7 + x^6 + x^2 + x + 1$ (CDMA) | 0x2030B9C7 or 0x38E74301 |

| | | |
|---|---|---|
| | | (0x30185CE3) |
| CRC-32-Adler | Không phải một CRC; xem Adler-32 | xem Adler-32 |
| CRC-32-IEEE 802.3 | $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (V.42, MPEG-2,PNG ) | 0x04C11DB7 or 0xEDB88320 (0x82608EDB) |
| CRC-32C (Castagnoli) | $x^{32} + x^{28} + x^{27} + x^{26} + x^{25} + x^{23} + x^{22} + x^{20} + x^{19} + x^{18} + x^{14} + x^{13} + x^{11} + x^{10} + x^9 + x^8 + x^6 + 1$ | 0x1EDC6F41 or 0x82F63B78 (0x8F6E37A0) |
| CRC-32K (Koopman) | $x^{32} + x^{30} + x^{29} + x^{28} + x^{26} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{11} + x^{10} + x^7 + x^6 + x^4 + x^2 + x + 1$ | 0x741B8CD7 or 0xEB31D82E (0xBA0DC66B) |
| CRC-64-ISO | $x^{64} + x^4 + x^3 + x + 1$ (HDLC — ISO 3309) | 0x000000000000001B or 0xD800000000000000 (0x8000000000000000D) |
| CRC-64-ECMA-182 | $x^{64} + x^{62} + x^{57} + x^{55} + x^{54} + x^{53} + x^{52} + x^{47} + x^{46} + x^{45} + x^{40} + x^{39} + x^{38} + x^{37} + x^{35} + x^{33} + x^{32} + x^{31} + x^{29} + x^{27} + x^{24} + x^{23} + x^{22} + x^{21} + x^{19} + x^{17} + x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^4 + x + 1$(as described in ECMA-182 p.63) | 0x42F0E1EBA9EA3693 or 0xC96C5795D7870F42 (0xA17870F5D4F51B49) |

# II. TÌM HIỂU CRC - 4

## 1. CRC - 4 trong PCM30

Trong trường hợp PCM 30 được sử dụng để truyền số liệu thi biot SI trong khe thơi gian TS0 là bit kiểm tra dư chu trinh CRC.

| T.T khung | Bit 1 đến bit 8 của TS0 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | SI | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | C1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | A | S | S | S | S | S |
| 2 | C2 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 3 | 0 | 1 | A | S | S | S | S | S |
| 4 | C3 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 5 | 1 | 1 | A | S | S | S | S | S |
| 6 | C4 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 7 | 0 | 1 | A | S | S | S | S | S |
| 8 | C1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 9 | 1 | 1 | A | S | S | S | S | S |
| 10 | C2 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 11 | 1 | 1 | A | S | S | S | S | S |
| 12 | C3 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 13 | E | 1 | A | S | S | S | S | S |
| 14 | C4 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 15 | E | 1 | A | S | S | S | S | S |

***Bảng: Tóm tắc chức năng các bit của khe thơi gian TS0 trong mỗi đa khung 16 khung.***

Cũng có thể xem đa khung gồm đa khung con:

+ Đa khung con thứ nhất gồm khung 0 đến khung 7

+ Đa khung con thứ 2 gồm khung 8 đến khung 15.

+| Bit SI trong các khung chẵn của mỗi khung con là các bit kiểm tra dư chu trình C1, C2, C3, C4 (CRC-4).

+ Bit SI trong các khung lẻ của đa khung tạo thành từ mã đồng bộ đa khung CRC-4, bit E trong khung 13 chỉ thị lỗi bit của CRC-4 của đa khung con thứ nhất và bit E trong khung 15 chi thị lỗi bit của CRC-4 của đa khung con thứ 2.

## 2. Tính toán CRC-4

+ Phía phát thiết bị PCM 30 hình thành các đa khung con gồm 8 khung, mỗi khung có 256 bit, vì vậy đa khung con chứa 2048 bit.

+ Giả thiết trong số 2048 bit này của đa khung con n-1 có 1201 bit 1, tức là tổng số các bit 1 bằng 1201.

+ Chuyển 1201 thành số nhị phân có chiều dài là: **10010110001**

+ Triển khai số nhị phân này thành đa thức đặc trưng sau đây:

$$x^{10} + x^7 + x^5 + x^4 + x^0.$$

+ Nhân đa thức đặc trưng với $x^4$ được:

$x^{14} + x^{11} + x^9 + x^8 + x^4$, chia cho đa thức sinh $x^4 + x + x^0$.

$$\begin{array}{l|l}
x^{14} + x^{11} + \quad + x^9 + x^8 + \qquad\qquad + x^4 & x^4+x+x^0 \\
\underline{x^{14} + x^{11} + x^{10}} & \\
x^{10} + x^9 + x^8 + \qquad\qquad + x^4 & x^{10}+x^6+x^5+x^4+x^3+x^0 \\
\underline{x^{10} + \qquad + x^7 + x^6} & \\
x^9 + x^8 + x^7 + x^6 + \qquad + x^4 & \\
\underline{x^9 + \qquad + x^6 + x^5} & \\
x^8 + x^7 + \quad + x^5 + x^4 & \\
\underline{x^8 + \qquad + x^5 + x^4} & \\
x^7 + & \\
\underline{x^7 + x^4 + x^3} & \\
x^4 + x^3 & \\
\underline{x^4 + \ + x + x^0} & \\
x^3 + x + x^0 \quad = 1011 \text{ (số dư)} &
\end{array}$$

+ Các bit dư 1011 được đặt vào vị trí các bit C1, C2, C3, C4 của đa khung con n.

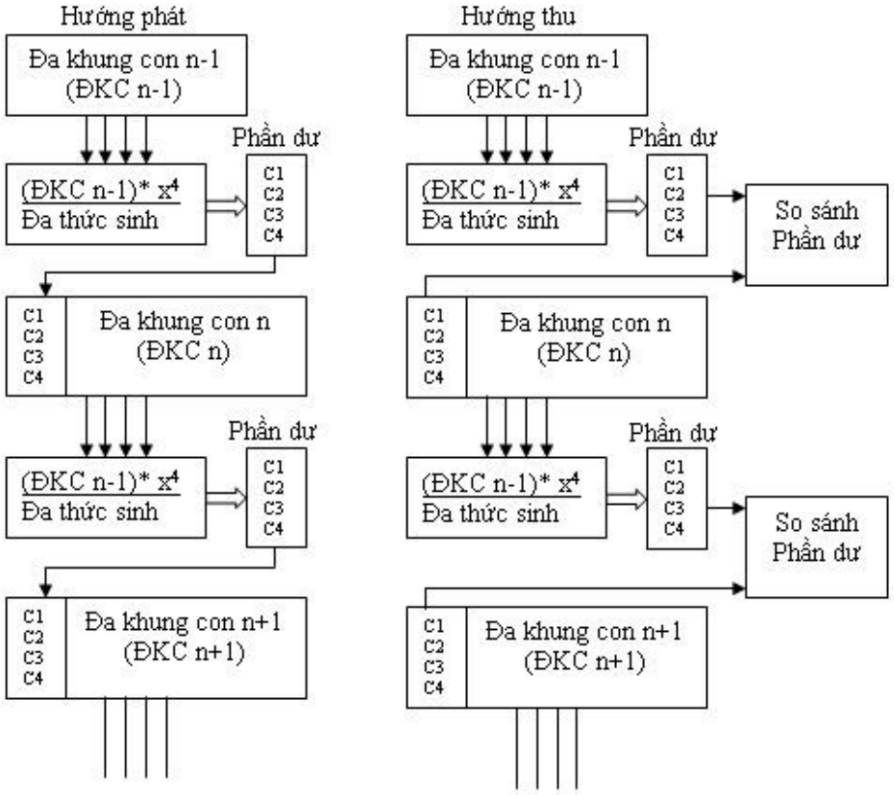
+ Đa khung con n-1 truyền đến phía thu và đếm được tổng số bit 1 bằng 1201, chuyển đổi thành số nhị phân 1001010001.

+ Triển khai số nhị phân thành đa thức đặc trưng: $x^{10} + x^7 + x^6 + x^5 + x^4 + x^0$.

+ Nhân đa thức đặc trưng với $x^4$ và chia tích số này cho đa thức sinh $x^4 + x + x^0$ được số dư là 1011.

+ Đem số dư 1011 so sách với số dư đã đặt trong đa khung con n mà phía phát chuyển tới (1011), như vậy không có lỗi trong đa khung con n-1.

+ Sơ đồ cài đặt và xử lý CRC-4 như hình sau:




Sơ đồ hình thành và xử lý CRC-4

# III. TÍNH TOÁN CRC – 8

+ Là một ví dụ về thực hiện phân chia đa thức trong phần cứng, giả sử rằng chúng tôi đang cố gắng để tính toán một CRC 8-bit của một tin nhắn 8- bit ký tự ASCII "W", đó là nhị phân $_{01010111}$ 2, số thập phân$_{87}$ 10, hoặc hệ thập lục phân 57 $_{16.}$

+ Để minh hoạ, chúng tôi sẽ sử dụng CRC-8-ATM ( HEC ) đa thức $x^8 + x^2 + x +$ 1.Viết bit đầu tiên truyền (hệ số sức mạch cao nhất của x) bên trái, điều này tương ứng với chuỗi 9- bit “100000111”.

+ Các giá trị byte 57 $_{16}$ có thể được truyền đi trong hai đơn đặt hàng khác nhau, tùy thuộc vào quy ước bit đặt hàng sử dụng . Mỗi người tạo ra một tin nhắn khác nhau đa *thức* M *(x)* .

+ Msbit đầu tiên, điều này *là* $x^6 + x^4 + x^2 + x + 1 = 01010111$, trong khi lsbit đầu tiên, nó *là* $x^7 + x^6 + x^5 + x^3 + x = 11101010$ . Những sau đó có thể được nhân để sản xuất hai đa thức tin nhắn *16-bit* $x^8 M(x) x^8$ .

+ Tính toán phần còn lại sau đó bao gồm trừ đi bội số của máy phát điện đa *thức* G *(x)* . Đây chỉ là như phân chia số thập phân dài, nhưng thậm chí còn đơn giản bởi vì bội chỉ có thể ở mỗi bước là 0 và 1, và trừ mượn "từ vô cực" thay vì giảm các chữ số trên. Bởi vì chúng ta không quan tâm về thương, không có cần phải ghi.

| Bit đầu tiên có ý nghĩa nhất | Ít nhất-đáng bit đầu tiên |
|---|---|
| 01010111 00000000<br>- 00000000 0<br>= 01010111 00000000<br>- 100000111<br>= 00010110 11000000<br>- 00000000 0<br>= 00010110 11000000<br>- 100000111<br>= 00001101 01100000<br>- 00000000 0<br>= 00001101 01100000<br>- 100000111<br>= 00000110 10110000<br>- 00000000 0<br>= 00000110 10110000<br>- 100000111<br>= 00000001 01010110<br>- 00000000 0<br>= 00000001 01010110 | 11101010 00000000<br>- 100000111<br>= 01101001 10000000<br>- 100000111<br>= 00101000 01000000<br>- 100000111<br>= 00010000 10100000<br>- 00000000 0<br>= 00010000 10100000<br>- 100000111<br>= 00000100 01100000<br>- 00000000 0<br>= 00000100 01100000<br>- 100000111<br>= 00000000 11000000<br>- 00000000 0<br>= 00000000 11000000 |

+ Quan sát rằng sau mỗi phép trừ, các bit được chia thành ba nhóm: lúc đầu, một nhóm mà là tất cả không lúc kết thúc, một nhóm là không thay đổi từ bản gốc; và một nhóm bóng mờ ở giữa đó là "thú vị" . Nhóm "thú vị" là 8 bit dài, phù hợp với mức độ đa thức. Mỗi bước, nhiều thích hợp của đa thức là trừ nhóm không trở nên lâu hơn một chút, và nhóm không thay đổi trở nên ngắn hơn một chút, cho đến khi chỉ còn lại cuối cùng là bên trái.

+ Trong ví dụ msbit đầu tiên, còn lại đa thức $x^7 + x^5 + x$. Chuyển đổi một số thập lục phân bằng cách sử dụng quy ước rằng quyền lực cao nhất *của* x là msbit, điều này là $_{A2}$ 16 . Trong lsbit đầu tiên, còn lại là $x^7 + x^4 + x^3$. Chuyển đổi sang hệ thập lục phân bằng cách sử dụng quy ước rằng quyền lực cao nhất *của* x là lsbit, đây là $_{19}$ 16 .

**** Ví dụ:***

- Giả thiết trong số 2048 bit này của đa khung con n-1 có 1058 bit 1, tức là tổng số các bit 1 bằng 1058.

- Chuyển 1058 thành số nhị phân có chiều dài là: **10000100110.**

- Triển khai số nhị phân này thành đa thức đặc trưng sau đây: $\mathbf{x^{10} + x^5 + x^2 + x^1}$.

- Nhân đa thức đặc trưng với $\mathbf{x^8}$ được:

$\mathbf{x^{18} + x^{13} + x^{10} + x^9}$ ,chia cho đa thức sinh $\mathbf{x^8 + x^2 + x + x^0}$.

$$
\begin{array}{l|l}
x^{18}+x^{13}+\quad+x^{10}+x^9 & x^8+x^2+x^1+x^0 \\
\underline{x^{18}+\quad+x^{12}+x^{11}+x^{10}} & \\
\quad x^{13}+x^{12}+x^{11}+\quad+x^9 & x^{10}+x^5+x^4+x^3+x^1 \\
\quad \underline{x^{13}+\qquad\qquad+x^7+x^6+x^5} & \\
\qquad x^{12}+x^{11}+\quad+x^9+x^7+x^6+x^5 & \\
\qquad \underline{x^{12}+\qquad\qquad\quad+x^6+x^5+x^4} & \\
\qquad\quad x^{11}+\quad+x^9+x^7+\qquad+x^4 & \\
\qquad\quad \underline{x^{11}+\qquad\qquad\quad+x^5+x^4+x^3} & \\
\qquad\qquad x^9+x^7+\quad+x^5+\qquad x^3 & \\
\qquad\qquad x^9+x^3+x^2+x^1 & \\
\qquad\qquad\quad \underline{\qquad\qquad\qquad\qquad} & \\
\qquad\qquad\quad x^7+x^5+x^2+x^1 = 10100110 \text{ (số dư)} &
\end{array}
$$

# TÀI LIỆU THAM KHẢO

1. Giáo trình ghép kênh số – Học viện BCVT Hồ Chí Minh
2. Wed : tailieu.vn
3. Tài liệu ghép kênh PDH & SDH – Học viện BCVT