

问答式远动规约 CRC 的软件实现

陈剑云 刘方中 杨江波

(电气工程系)

摘 要

问答式远动规约规定远动报文数据以异步方式传输,因而报文的检错校验工作需用软件计算来完成。本文通过对远动规约中几种不同报文格式的分析,推导出对应报文的软件校验算法,计算出相应的余数表,并介绍在远动系统中实现的方法及软件结构。

关键词:远动;问答式规约;CRC

0 引 言

由于微型计算机在远动系统中的应用,使问答式远动系统得到了迅速的发展。为了促进我国问答式远动系统的研究、制造和应用,水电部地区电网调度自动化专业组制定了一套问答式远动规约^①。该规约是以英国西屋公司的 polling 规约为蓝本而制定的,如何在国产远动系统中应用好这套规约,有许多工作值得研究与探讨,而其中 CRC (Cyclic Redundancy check 循环冗余校验) 的软件实现是其中的基本问题之一。

问答式远动系统实际上是一种特殊形式的计算机网络,它的通讯方式类似计算机网络。问答式远动规约中规定,主站与 RTU (Remote Terminal Unit 远方终端单元) 之间进行通讯的数据报文格式采用类似计算机网络 HDLC (高级数据链路控制) 规程的格式,不过数据要求以异步方式传送。数据以异步方式传送可以降低通道的成本,但却要付出额外的代价用于 CRC 的实现。

众所周知,HDLC 规程是高级同步数据链路控制规程,报文的 CRC 校验工作有专门的硬件(如 Z80 SIO、Intel 8273、Mc6854 等芯片)自动完成。而在异步方式下,CRC 校验工作没有对应的硬件帮助,因而只有通过软件计算来完成。CRC 的软件实现要求时空复杂性愈小愈好,比较有效的算法是软件查表算法,软件查表算法是把一些复杂的运算事先计算出来,然后做成表存放在内存中。因此在实际的 CRC 运算过程中,只需查几次表,做些简单的运算就能完成。查表减轻了运算的复杂度,但却增加了内存空间的开销。软件查表算法可以推出多种形式,如何选用合适的算法,使得时间复杂性与空间复杂性相适应,在实际应用中是非常

本文于 1992 年 4 月 20 日收到

重要的。

问答式远动规约按CRC长度分有三种报文格式,即CRC—16 校验码的报文格式、8 位校验码的报文格式以及无校验码的报文格式,如图1所示。下面将分别对前两种需校验的报文进行分析,导出其算法与余数表。

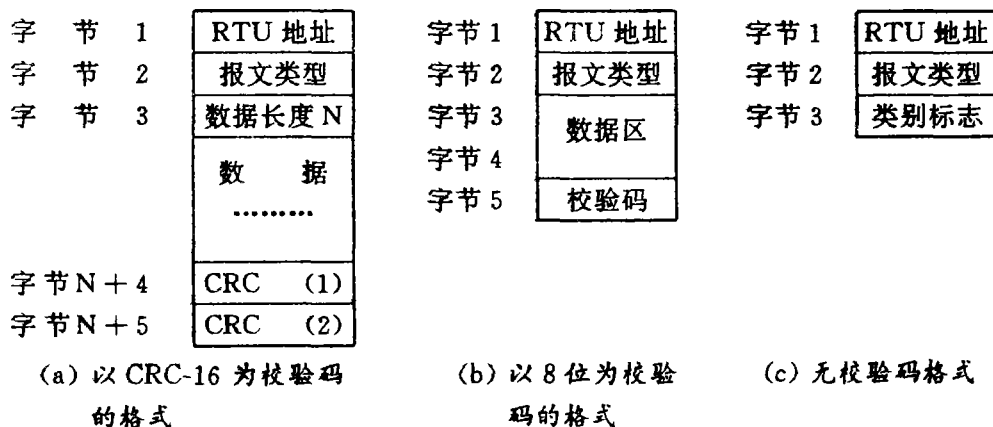


图1 问答式远动规约报文格式

1 CRC-16 校验的实现原理

1.1 CRC-16 校验码的报文格式

如图1(a)所示,这个格式是规约中最主要格式,用得最多。它从结构上类似HDLC-CRC格式,所不同的是它的校验码生成多项式是 $g_1(x) = x^{16} + x^{15} + x^2 + 1$ 。(HDLC-CRC的生成多项式为 $g(x) = x^{16} + x^{12} + x^5 + 1$)^{*}。信息码 $f(x)$ 的传送顺序为:从字节1的 b_0 位开始,直到数据区的最后字节 $N+3$ 的 b_7 位。令 $f(x)$ 除以 $g_1(x)$ 所得的余式设为 $r_1(x) = b_0x^{15} + b_1x^{14} + \dots + b_{14}x + b_{15}$, $r_1(x)$ 即为校验码。CRC排列如下:

$$\text{CRC}(1); b_7 \quad b_6 \quad b_5 \quad b_4 \quad b_3 \quad b_2 \quad b_1 \quad b_0$$

$$\text{CRC}(2); b_{15} \quad b_{14} \quad b_{13} \quad b_{12} \quad b_{11} \quad b_{10} \quad b_9 \quad b_8$$

1.2 软件查表算法的推导

把一帧数据报文的信息位(总长 $N+3$ 字节),按两字节分成一组,构成 m 组(若总长为奇数,则假想增加一全为零的字节放在报文开头处),则

$$f(x) = R_m(x) \cdot x^{16 \cdot m} + R_{m-1}(x) \cdot x^{16 \cdot (m-1)} + \dots + R_1(x) \cdot x^{16 \cdot 1} + R_0(x) \cdot x^{16 \cdot 0} \quad (1)$$

$$\text{式中} \quad R_i(x) = b'_{H0}x^{15} + b'_{H1}x^{14} + \dots + b'_{H7}x^8 + b'_{L0}x^7 + \dots + b'_{L7} \quad (2)$$

$R_i(x)$ 为第 i 组数据写成的多项式, b'_{Hj} 为高字节的第 j 位数据。 b'_{Lj} 为低字节的第 j 位数据。

* 文献[1]中把 $g_1(x) = x^{16} + x^{15} + x^2 + 1$ 误为HDLC标准CRC校验码的生成多项式。

式 (1) 又可写成

$$f(x) = x^{16} \cdot [R_1(x) + x^{16} \cdot [R_2(x) + x^{16} \cdot [\dots x^{16} [R_{m-1}(x) + x^{16} \cdot R_m(x)]] \dots]] \quad (3)$$

令 $f(x)$ 除以 $g_1(x)$ 得余式 $s(x)$

$$s(x) = x^{16} \odot [R_1(x) + x^{16} \odot [R_2(x) + x^{16} \odot [\dots x^{16} \odot [R_{m-1}(x) + x^{16} \odot R_m(x)]] \dots]] \quad (4)$$

式中 $\odot$ 为模 $g_1(x)$ 乘号^[2], 它的意义即为两项相乘后, 除以 $g_1(x)$, 取所得的余数做为其结果。

$$\begin{aligned} \text{令} \quad s_m(x) &= R_m(x) \\ s_{m-1}(x) &= R_{m-1}(x) + x^{16} \odot s_m(x) \end{aligned}$$

则有递推公式

$$\begin{aligned} s_{i-1}(x) &= R_{i-1}(x) + x^{16} \odot s_i(x) \\ \text{其中 } i &= m, \dots, 2 \end{aligned} \quad (5)$$

所得余式

$$s(x) = x^{16} \odot s_1(x) \quad (6)$$

$$\text{令} \quad T_i(x) = x^{16} \odot s_i(x) \quad (7)$$

则 (5) 式变为

$$\begin{aligned} s_{i-1}(x) &= R_{i-1}(x) + T_i(x) \\ i &= m, \dots, 2 \end{aligned} \quad (8)$$

余式为

$$s(x) = T_1(x) \quad (9)$$

1.3 分段余数表的生成

求余数的关键在于求取 T_i , 而求 T_i 的最快最好的办法是查表, 即预先做出 $s_i(x)$ 与 $T_i(x)$ 的关系表, 给定一个 $s_i(x)$, 通过查表而直接得到 T_i , 因而可以免去大量的计算, 节省时间。

$s_i(x)$ 是一个 15 次的多项式, 若用 $s_i(x)$ 直接生成余数表, 每个余数占用两个字节, 则表长为 $2^{16} \times 2$ 字节, 表所需存储空间太大, 无法实现。可行的办法是对 $s_i(x)$ 进行分段, 每段分别生成余数表, 在计算过程中, 分段查表, 然后叠加生成 $T_i(x)$ 。分段后运算量有所增加, 但内存占用量会大大地减少。

若考虑 $s_i(x)$ 的每一位为一段, 则分成 16 段, 共有 16 个表, 占用内存 $16 \times 2^1 \times 2 = 64$ 字节, 运算中查表 16 次; 若 $s_i(x)$ 四位为一段, 共有 4 段, 4 个表, 占用内存为 $4 \times 2^4 \times 2 = 128$ 字节, 查表 4 次; 若 $s_i(x)$ 分成两段, 每段 8 位, 由表占用内存为 $2 \times 2^8 \times 2 = 1024$ 字节, 查

表两次。从内存占用量与运算时间综合考虑,以分成四段为宜。下面推导四段求余法的公式与余数表。

$$s_i(x) = x^{12} \cdot s_{i1}(x) + x^8 \cdot s_{i2}(x) + x^4 \cdot s_{i3}(x) + s_{i4}(x) \quad (10)$$

$s_{i1}(x)$, $s_{i2}(x)$, $s_{i3}(x)$, $s_{i4}(x)$ 皆为3次多项式。

则式(7)有

$$\begin{aligned} T_i(x) &= x^{16} \odot s_i(x) \\ &= x^{28} \odot s_{i1}(x) + x^{24} \odot s_{i2}(x) + x^{20} \odot s_{i3}(x) + x^{16} \odot s_{i4}(x) \end{aligned}$$

令

$$\begin{cases} T_{i1}(x) = x^{28} \odot s_{i1}(x) \\ T_{i2}(x) = x^{24} \odot s_{i2}(x) \\ T_{i3}(x) = x^{20} \odot s_{i3}(x) \\ T_{i4}(x) = x^{16} \odot s_{i4}(x) \end{cases} \quad (11)$$

则

$$T_i(x) = T_{i1}(x) + T_{i2}(x) + T_{i3}(x) + T_{i4}(x) \quad (12)$$

以 $s_{ij}(x)$ 做为自变量, $T_{ij}(x)$ 做为因变量, 通过(11)式计算, 可得出余数表1。

2 8位校验的实现原理

2.1 8位校验码的报文格式

如图1(b)所示, 这种格式仅仅用来向RTU传送询问报文(05H)和重复报文(1AH)。校验码的最高位 b_7 为所有信息字节 b_i 的异或结果; 校验码的其余 b_6-b_0 以生成多项式 $g_2(x) = x^7 + x^6 + x + 1$ 去除字节所有信息字节的 b_6-b_0 所构成的信息码 $f(x)$, 而 $f(x)$ 的顺序为: 从字节1的 b_0 开始, 依次至 b_6 , 直至最后一字节的 b_6 为止。所得余数为

$$r_2(x) = b_0x^6 + b_1x^5 + \cdots + b_5x + b_6$$

排列顺序为

$$(b_7) \quad b_6 \quad b_5 \quad b_4 \quad b_3 \quad b_2 \quad b_1 \quad b_0$$

2.2 算法推导

校验码的 b_7 位直接取信息字节做异或运算, 取结果的 b_7 位即可生成。而校验码的 b_6-b_0 位, 仍用软件查表方法生成。

设报文信息位共有 m 个字节, 每个字节屏蔽 b_7 位后, 只剩7位, 则信息码 $f(x)$ 可以表示成

$$f(x) = x^{7m}R_m(x) + x^{7(m-1)} \cdot R_{m-1}(x) + \cdots + x^7 \cdot R_1(x) \quad (13)$$

$$\text{式中 } R_i(x) = b_0^i x^6 + b_1^i x^5 + \cdots + b_5^i x + b_6^i \quad (14)$$

余 数 表 表 1

S_{ij}	T_{i2}	T_{i1}	T_{i4}	T_{i3}
0	00 00	00 00	00 00	00 00
1	01 90	0D C0	C1 C0	01 CC
2	01 60	19 C0	81 C1	01 D8
3	00 F0	14 00	40 01	00 14
4	02 C0	31 C0	01 C3	01 F0
5	03 50	3C 00	C0 03	00 3C
6	03 A0	28 00	80 02	00 28
7	02 30	25 C0	41 C2	01 E4
8	07 C0	61 C0	01 C6	01 A0
9	06 50	6C 00	C0 06	00 6C
A	06 A0	78 00	80 07	00 78
B	07 30	75 C0	41 C7	01 B4
C	05 00	50 00	00 05	00 50
D	04 90	5D C0	C1 C5	01 9C
E	04 60	49 C0	81 C4	01 88
F	05 F0	44 00	40 04	00 44

b_j^i 为信息位的第 i 字节第 j 位。

用类似 1.2 节中的方法, 可得以下递推公式

$$s_{i-1}(x) = R_{i-1}(x) + T_i(x) \quad (15)$$

$$i = m, \dots, 2$$

$$T_i(x) = x^7 \odot s_i(x) \quad (16)$$

$$\text{余式} \quad s(x) = T_1(x) \quad (17)$$

2.3 余数表的生成

此处 $s_i(x)$ 为 6 次多项式, 若不分段, 直接用 $s_i(x)$ 生成余数表, 每个余数占用一个字节, 则表的总长为 $2^7 \times 1 = 128$ 字节, 这个表占用的空间不大, 所以不需再进行分段。

由式 (16) 可计算出其余数表如表 2 所示。

7 位 余 数 表

表 2

T S ₆ -S ₄	S ₃ -S ₀															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	02	42	40	21	23	63	61	12	10	50	52	32	31	71	73
1	0A	08	48	4A	2B	29	69	6B	18	1A	5A	58	39	3B	7B	79
2	05	07	47	45	24	26	66	64	17	15	55	57	36	34	74	76
3	0F	0D	4D	4F	2E	2C	6C	6E	1D	1F	5F	5D	3C	3E	7E	7C
4	43	41	01	03	62	60	20	22	51	53	13	11	70	72	32	30
5	49	4B	0B	09	68	6A	2A	28	5B	59	19	1B	7A	78	38	3A
6	46	44	04	06	67	65	25	27	54	56	16	14	75	77	37	35
7	4C	4E	0E	0C	6D	6F	2F	2D	5E	5C	1C	1E	7F	7D	3D	3F

3 CRC 校验的软件实现

CRC 校验工作实际上包含两个方面的内容, 其一是发送报文前, 根据信息位生成校验码。另外是收到报文后, 进行校验计算, 检查是否有错误。两者的工作原理与实现过程基本上相同。校验报文时, 只需像生成校验码的过程一样, 把生成的校验码与收到校验码的内容比较, 判其是否一致来判断正误。

图 2 给出了校验码生成的软件 PAD (Problem Analysis Diagram) 图。

4 结 束 语

CRC 的软件查表算法可以推出其它的形式, 推导过程的差异, 直接影响余数表占用内存的大小以及数据内容。文献 [3] 中 CRC-16 的校验码计算采用按字节分成一组的方法, 其余数表所用的内存较本文中的少一半, 但其算法稍嫌复杂; 另外其 8 位校验码没有采用规约中规定的生成多项式 $g(x) = x^7 + x^6 + x + 1$, 而是采用 $g(x) = x^7 + x^4 + 1$ 。本文严格按照规约的要求, 推导出相应的算法, 编制相应的软件, 并得以在新开发的远动系统中应用, 实验证明其内存占有量与运算速度都是适宜的。

另外值得一提的是, 由于通讯数据位以低位开始传送, 与传统习惯相反, 因而余数表中的数据需反序生成, 给运算增加了很大的难度。在分析与探讨中, 作者用 C 语言编制了一套通用程序, 对于不同的生成多项式可自动生成相应的余数表。

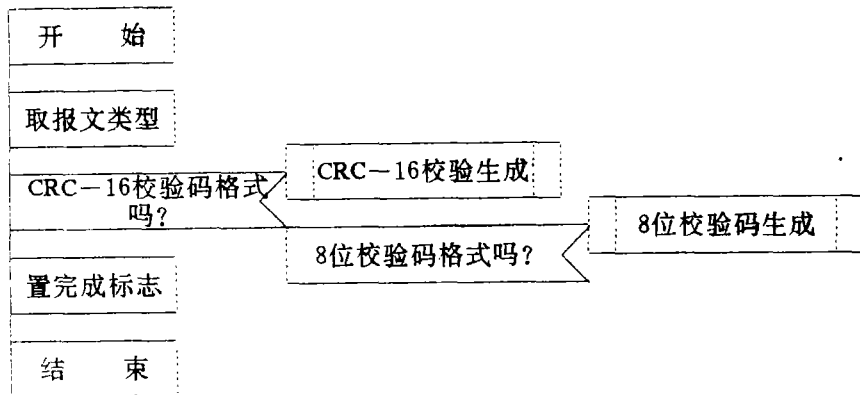


图2 (a)主程序模块

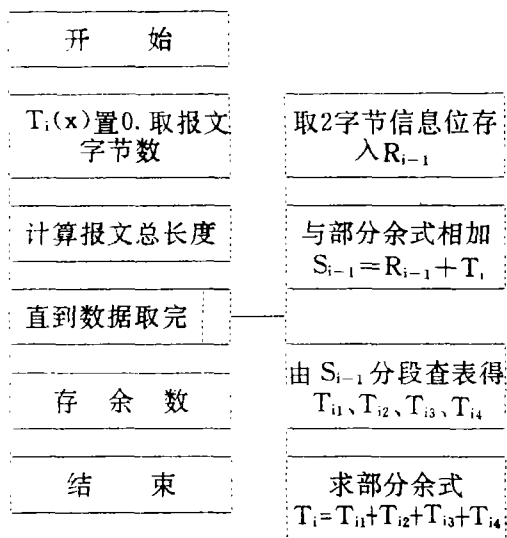


图2 (b)CRC-16校验码生成模块

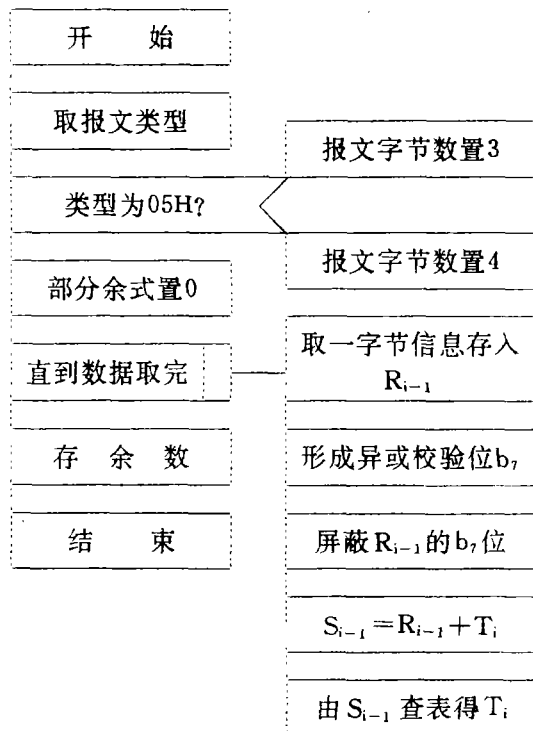


图2 (c)8位校验码生成模块

参 考 文 献

- [1] 水电部地区电网调度自动化专业组. 问答式远动规约(试行). 地区电网调度自动化技术文件汇编. 南京: 电力系统自动化杂志社, 1987
- [2] 刘贯宇. 电力系统远动技术. 北京: 水利电力出版社, 1986: 238~244
- [3] 李炎灵. N4F系列RTU综述(下). 电力自动化设备, 1991; (1)
- [4] 马锦林. 软件工程引论. 南京: 南京大学出版社, 1987

The Realization of CRC in the Telecontrol Polling Protocol

Chen Jianyun Liu Fangzhong Yang Jiangbo

Abstract

According to the telecontrol polling communication protocol, which the messages are transmitted by using asynchronous transmission protocol, the cyclic redundancy ckeck (CRC) must be calculated by software. In this paper, the algorithms for two kinds of messages cyclic redundancy check are deduced respectively, and the relative residuc tables are given. The method of CRC realization and the archriture of the software in a remote control system are introduced.

Key words: Telecontrol; Polling Protocol; CRC