DOI：10.13879/j.issn.1000-7458.2024-08.24034

扫码浏览下载

欧标应答器报文编码算法优化与实现

卓　鹏，刘江沙，方志刚

摘　要：针对欧标应答器报文编码耗时不确定问题，在分析应答器报文格式和编码算法流程的基础上，从2个重点方向对算法的多个步骤进行优化提升：一是提高计算效率，如通过查表法提高加扰、校验位、汉明距的计算效率；通过一个整型数存储2个11 bit字，从而有效减少移位操作，并且基于此使得有效字检查与移位操作可并行执行；借助整型数实现一次执行获得4 bit的漏取样操作等。二是避免大量无效计算，如调整同步偏离解析条件检查的左右移位执行顺序；通过贪心算法检查连续有效字；并行执行加扰与部分同步偏离解析条件有效字检查；并行执行校验位计算与字母表条件检查等。采用盐通高铁应答器报文进行性能测试，验证基于本编码优化算法的通用CPU代替专用编码单元的可行性，从而降低设备成本，提高系统可靠性。

关键词：欧标应答器；报文；编码；查表法；循环冗余校验码
中图分类号：U284.48　　**文献标识码：**A

Optimization and Implementation of Eurobalise Message Encoding Algorithm

ZHUO Peng, LIU Jiangsha, FANG Zhigang

Abstract: To address the ancertainty of the encoding time-consuming of the Eurobalise message, the encoding algorithm process is strategically refined on the basis of the analysis on the balise message format and the encoding algorithm process. This involves optimizing and improving multiple algorithm steps in two key areas: On the one hand, enhancing computational efficiency — such as by utilizing look-up tables to improve computational efficiency of scrambling, check bits and Hamming distance, storing two 11-bit words in an integer number to effectively reduce shift operations and enable parallel execution of valid word check and shift operation, and employing integer numbers to achieve 4-bit under-sampling operation in one execution; On the other hand,

卓　鹏：北京交大微联科技有限公司　工程师　100043　北京
刘江沙：北京交大微联科技有限公司　工程师　100043　北京
方志刚：北京交大微联科技有限公司　工程师　100043　北京
收稿日期：2024-02-02
引用格式：卓鹏，刘江沙，方志刚. 欧标应答器报文编码算法优化与实现[J]. 铁道通信信号，2024，60(8)：34-40.
Citation: ZHUO Peng, LIU Jiangsha, FANG Zhigang. Optimization and Implementation of Eurobalise Message Encoding Algorithm[J]. Railway Signalling & Communication, 2024, 60(8): 34-40.

minimizing invalid calculations—by adjusting the left-right shift execution order of Off-Synch-Parsing condition test, checking consecutive valid words using a greedy algorithm, executing scrambling and partial Off-Synch- Parsing condition test for valid word in parallel, and performing check bit calculations and alphabet condition checks in parallel. Performance testing on the Yancheng-Nantong High-speed Railway's balise message confirms the viability of replacing the special encoding unit with a general CPU based on the optimized algorithm, leading to reduced equipment costs and improved system reliability.

Key words: Eurobalise; Telegram; Encoding; Look-up table method; Cyclic Redundancy Check (CRC)

我国国铁和城市轨道交通广泛使用的欧标应答器（以下简称“应答器”），其报文的编码规范引自欧洲铁路运输联盟制定的《FFFIS for Eurobalise》[1]（SUBSET-036），目前该规范已于2023年更新到基线4。国内对应的标准是《应答器传输系统技术条件》[2]（TB/T 3485—2017）。应答器报文编码方式的特点决定了其耗时具有不确定性，在极端情况下还存在不可编的情况[1]。较早研究编码算法实现的文献［3］认为，在加扰位确定之后，长报文前928 bit的校验位计算结果就可以确定，这部分可以在确定额外修正位之前先行计算，从而减少大量重复计算的工作，同时提出在10 bit到11 bit的逆变换过程中采用折半查找的方式；文献［4］构造反向查询的字母表，通过查表法直接判断当前字是否为有效的合法字，相比二分法查找能够明显节省时间。对于应答器报文编码比较耗时问题，较多的做法是使用FPGA算法[5-7]，也有基于C语言实现编码算法的[8-10]，但均未提出实质性的优化策略。

基于应答器设备的广泛应用，其编码效率的提升有助于降低各项社会经济成本，本文在前述研究的基础上，采用C语言，通过查表法、优化11 bit字数据存储、并行执行、利用贪心算法检查连续有效字、优化修正约束条件检查等措施，使应答器报文编码算法的性能有显著提升。

1 报文格式

《FFFIS for Eurobalise》（SUBSET-036）规定的应答器传输报文分为长报文和短报文。2种格式的报文均由5部分组成：整形后的数据、控制位、加扰位、额外修正位、校验位[2]。应答器报文格式见表1。

长报文和短报文仅整形后的数据不一样。长报文的原始用户数据报文是830 bit，通过对其进行加扰和整形操作，变为913 bit的整形数据，加上后面4个部分的数据，共1 023 bit，通常以$b_{n-1},b_{n-2},\cdots,b_1,b_0$表示，$n$=1 023。短报文的原始用户数据报文长210 bit，经过加扰和整形变为231 bit，加上后面的4个部分的数据，共341 bit。短报文的编码算法与长报文无本质区别，且我国轨道交通行业广泛使用长报文，故本文以长报文为例展开论述。

表1 应答器报文格式 bit

整形后的数据	控制位	加扰位	额外修正位	校验位
83×11=913 21×11=231	3	12	10	85

长报文的1 023 bit按一组11 bit可划分为93组，称每一组为一个字。原始用户数据报文经加扰和整形后的数据913 bit可划分为83组字，控制位目前固定取001b[2]，加上加扰位的高8 bit，组成第84组字。加扰位的低4 bit加上额外修正位的高7 bit组成第85组字。额外修正位的低3 bit加上85 bit的校验位，组成第86～93组字。1 023 bit需满足以下4个修正约束检查条件[2]。

条件1：字母表检查。即从0～（2^{11}—1）整数里挑出特定的1 024个数，组成一张字母表。要求报文里的每一个字均在字母表中，也称之为有效字。

条件2：同步偏离解析检查。即对1 023 bit报文进行循环移位，移位后再重新划分成93组字，在左移1 bit或右移1 bit时，新划分的93组字里连续有效字的个数不能超过2个；其他移位情况（由于移位11 bit又回到了自身，左移10 bit等效于右移1 bit，故其他移位也就是左移2～9 bit）下，新划分的93组字里连续有效字的个数不能超过10个。注意连续有效字的检查是对93组字进行循环检查。

条件3：非周期条件是否满足要求检查（仅限长报文）。设i为11的倍数，$b_{i-1},\cdots,b_{i-22}$与$b_{i-341-1},\cdots,b_{i-341-22}$的汉明距至少为3，即93组字中任意连

续2组字与间隔31组之后的2组连续字的总汉明距至少为3。在此基础上对左移或右移1～3 bit后的新报文，也要求与原始报文按上述间隔对应的汉明距至少为2。

条件4：漏取样检查。漏取样系数为2时，报文序列变为$b_{n-2},b_{n-4},\cdots,b_1,b_{n-1},b_{n-3},\cdots,b_2,b_0$。要求新的报文序列，以及对该报文循环左移1～10 bit后的新报文，循环检查其连续有效字均不能超过30个。漏取样系数还要求取4、8、16时，也要相应满足连续有效字不能超过30个。

2　编码算法

TB/T 3485—2017中应答器报文编码流程见图1。

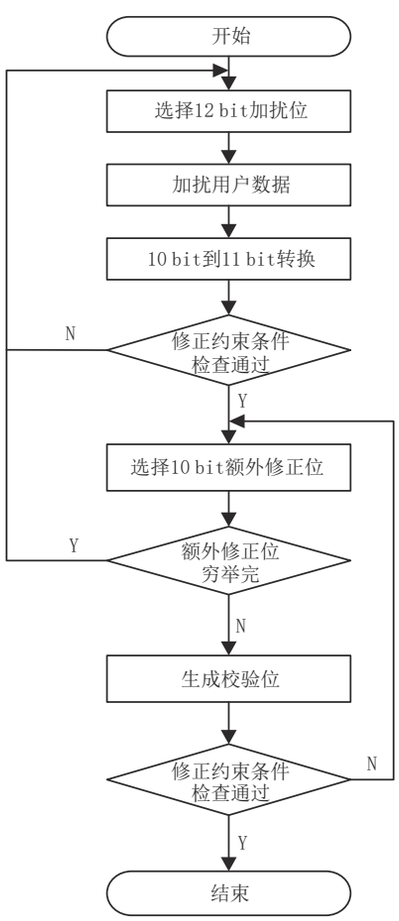

图1　应答器报文编码流程

Step 1　选择12 bit加扰位。根据文献［3］、文献［4］的分析，选定加扰位后，对第84组进行字母表条件检查，这样所有待尝试的加扰位数量就从4 096个降到2 512个。代码实现时，循环所有待尝试的加扰位，只需在循环体内首先执行第84组字的有效性检查即可。

Step 2　加扰用户数据。又可细分为3步：

Step 2.1　用所有用户位的函数替换第1组10 bit用户位，即将原始用户数据报文按10 bit一组进行划分，共划分为83组，再将83组数据按整型数相加取低10 bit替换掉第1组10 bit用户位；

Step 2.2　用12 bit加扰位B计算32 bit整数S，即$S=(2\,801\,775\,573\times B)\ \mathrm{mod}\ 2^{32}$；

Step 2.3　使用初始状态为S的32bit线性反馈移位寄存器进行加扰运算，加扰多项式为$h(x)=x^{32}+x^{31}+x^{30}+x^{29}+x^{27}+x^{25}+1$。加扰与常规的CRC计算无本质区别，只是采用的多项式不同，并需注意数据流高低位的问题。

Step 3　10 bit到11 bit的转换。依据文献［2］给定的1 024个字，实现所有10 bit整型数变为对应的11 bit整型数。以10 bit数作为下标，查文献［2］给定表即可获取对应的11 bit字。

Step 4　检查修正约束条件。对目前已经确定的前928 bit，尽可能地进行候选报文修正约束条件检查。显然整形后的数据，共计83组自动满足条件1；第84组已经在Step 1执行检查通过，第85组及之后需确定额外修正位才能确定，故此处无需再执行条件1检查。通过对某测试站预置的137条有源应答器报文进行全部加扰位尝试，发现单独执行条件2、条件3、条件4的淘汰率分别为86%、1%、0%。由于执行条件3和条件4检查比较耗时，而其淘汰率又非常低，故本步骤仅执行条件2。这也与文献［10］中的编码算法流程图描述一致。最终经过条件2淘汰之后，平均每条报文剩余352个加扰位可选。

Step 5　选择10 bit额外修正位。根据文献［3］、文献［4］的分析，额外修正位确定后，可以先执行第85组字的条件1检查。代码实现时，循环所有额外修正位，只需在循环体内首先执行第85组字的检查即可。通过对多个车站的预置有源应答器报文及盐通高铁全部正线无源应答器报文的测试统计，发现本步骤可淘汰一半的额外修正位。

Step 6　生成校验位。对目前已经得到的应答器报文的前938 bit的信息位，采用BCH编码方式计算85 bit的校验位，相应的计算多项式见文献［2］的附录A。根据文献［3］、文献［4］的分析，校验位的计算可以分为3步进行：

Step 6.1 先对前928 bit计算，得到BCH码的中间结果，记为*bch_temp*，这一步可提前至Step 5之前进行计算；

Step 6.2 在*bch_temp*的基础上加入计算10 bit的额外修正位，得到最终的*bch_final*；

Step 6.3 在*bch_final*基础上线性加上长格式报文对应的75位多项式$g_L(x)$[2]，得到最终需要的校验位。

根据Step 5的结论，本步骤的第2步、第3步，在尝试一个加扰位的所有可选额外修正位时，平均要计算512次。故*bch_temp*的计算放至Step 5之前可显著减少计算时间[3-4]。

Step 7 修正约束条件检查。首先执行条件1检查，只需执行最后8组字的检查。随后依次执行条件2、条件3、条件4的检查。针对某3个车站的所有预置有源应答器报文及盐通高铁的所有无源应答器报文进行测试，获取所有可通过检查的合法传输报文。各条件的淘汰率统计情况见表2。

表2 修正约束检查条件淘汰率 %

报文	条件1	条件2	条件3	条件4
同济测试站137条有源报文	99.61	30.82	1.33	0
甬金嵊州站86条有源报文	99.61	30.96	1.29	0
南通动车所236条有源报文	99.61	30.73	1.35	0
盐通高铁729条无源报文	99.61	30.66	1.30	0

由表2可知，条件1淘汰率最高，一个加扰位下的全部1 024个额外修正位，在条件1检查（含第85组）后，平均仅2个额外修正位得以通过。条件2淘汰率是在条件1淘汰之后的结果上再淘汰约30%，同样的条件3、条件4也是在前述条件淘汰之后的结果上再进行淘汰。由表2可知，条件1、条件2、条件3、条件4依次执行是最有效的淘汰检查方式。

3 优化策略

3.1 查表法计算加扰和校验位

通过对编码算法的研究，可以看到加扰和校验位的计算都是典型的线性反馈移位计算，这部分可以改用成熟的查表法以提高效率。典型的查表法均是一次查表完成8 bit即1个字节的计算。

针对加扰计算，由于需要先将用户数据按10 bit一组进行划分处理，故可以创建一次查表计算10 bit的表格，避免10 bit一组的数据再转换为8 bit一组的字节数据流，且后续Step 3也可以直接使用10 bit查表法的计算结果数据，直接查表转换成11 bit的字。8 bit查表法对应256个表值，10 bit查表法对应1 024个表值。由于解扰只是加扰的逆运算，故10 bit查表法也可以用于加快解扰运算。

校验位计算*bch_temp*时，928 bit刚好是8的倍数，采用典型的8 bit查表法；剩余的10 bit额外修正位，采用10 bit查表法。

文献［11］在应答器解码（译码）中提到采用字节型算法的查表法。对于编码中的校验位计算，对应的表值为85 bit，这可以用9个字节数组或3个整型数组进行存储，显然3个整型数组的效率优于9个字节数组。表值的生成方式与加扰查表法的表值生成相同，85 bit的表值生成和32 bit的表值生成，在逻辑和形式上也相同。

3.2 优化11 bit字存储方式

对于修正约束条件检查中的条件2、条件3、条件4均需针对11 bit字执行大量移位操作的情况，可定义一个全局整型数组变量，每个整型存储2个11 bit字，从而可有效减少移位操作。即常规的11 bit字数组为第1组$b_{1022},b_{1021},\cdots,b_{1012}$；第2组$b_{1011},b_{1010},\cdots,b_{1001}$；…；最后一组$b_{10},b_9,\cdots,b_0$。可以调整为第1组$b_{1022},b_{1021},\cdots,b_{1012},b_{1011},b_{1010},\cdots,b_{1001}$；第2组$b_{1011},b_{1010},\cdots,b_{1001},b_{1000},b_{999},\cdots,b_{990}$；…；最后一组$b_{10},b_9,\cdots,b_0,b_{1022},b_{1021},\cdots,b_{1012}$。这样对11 bit字的左移1～10的移位操作均只需2步操作即可，即先移位再取低11 bit。

由此在执行条件2、条件3、条件4的检查时，不需要先完成所有字的移位操作，而是在需要使用时才执行移位。这样在检查执行过程中失败时，即可避免后续不必要的移位操作，从而比常规思路节省大量时间。

3.3 优化条件2移位执行顺序

Step 4的修正约束条件检查执行条件2时，按常规实现思路，循环左移1 bit，执行一次有效字检查，然后再循环左移1 bit，再执行有效字检查，一共执行10轮。但右移1 bit和左移1 bit的连续有效字不能超过2个，比其他移位不能超过10个显然更易满足，故应当优先执行检查。通过对某测试站137条报文所有可能加扰位的淘汰情况进行统计发现，左移1 bit和右移1 bit谁先执行都能淘汰

17万多的候选，说明左移1 bit和右移1 bit谁先执行不影响效率。由数据的随机性，同样的也能推测左移n位和右移n位谁先执行不影响效率。左移1 bit加上右移1 bit的淘汰量占了总淘汰量的88%。

对于左移2～9 bit，记录3种执行顺序的结果，见表3。表3中，顺序1依次左移2～9 bit，左移9 bit即右移2 bit，淘汰量很大，却被最后执行，故效率会大受影响。顺序2依次左右移位2 bit，左右移位3 bit，左右移位4 bit，左右移位5 bit。顺序3依次左右移位5 bit，左右移位4 bit，左右移位3 bit，左右移位2 bit。对比顺序2和顺序3，可以看到，左右移位2 bit总是比左右移位3～5 bit淘汰更多候选，再结合顺序1的淘汰情况，可以说明顺序2是最佳的执行顺序，更能有效节省时间。

Step 7和Step 4中的条件2检查均适用本优化策略。

3.4　贪心算法检查连续有效字

条件2需要判断连续有效字不超过2个或10个，条件4需要判断连续有效字不超过30个。由于这些判断多数都是不满足的情况，因此没必要采取逐个递增检查是否连续满足的方式，可以先假定当前位置开始的连续n个字满足要求，然后反查是否真的逐个满足，不满足时，则继续以当前位置，假定后续的连续n个字满足要求，继续反查。当需要继续向前跳而剩余长度不足时，即不存在连续满足的字，则通过检查；当反查到第n个时，说明存在满足的连续字，可立即结束检查。由此就可以在存在大量非法字时，几乎以n为步长往前跳的方式快速完成检查，理想情况下理论上效率可提升n倍。连续合法字检查的贪心算法见图2。

3.5　并行执行加扰与条件2有效字检查

由于条件2检查的左移1 bit和右移1bit的连续有效字不能超过2个，即使采用贪心算法提升的极限也就是3倍。而通过调整Step 2、Step 3、Step 4的处理思路，可以不用等到所有的加扰都计算出来之后，再执行Step 3的转换，也不用等到Step 3的所有转换都完成后再进入Step 4的修正约束条件检查。即依据前面的10 bit查表法，计算出1个10 bit的加扰结果，就可以立即进行10 bit到11 bit的转换，然后再执行相应的左移1 bit和右移1 bit，无需贪心算法，简单顺序计数，一旦满足超过2个，就可以立即停止当前加扰位的尝试，从而有效减少后续的加扰计算，以及10 bit到11 bit的转换计算。通过对极端报文的测试，83组加扰计算提前结束的策略性能优于左右移位1 bit的贪心算法。

3.6　并行执行校验位计算与条件1有效字检查

前面算法已经提到校验位分3步进行，第2步和第3步针对所有可能的候选额外修正位，平均要计算512次。常规思路是在得到最终的85 bit校验位之后，再与低3 bit的额外修正位组成8组11 bit字，逐个进行有效字的检查。然而第2步的计算可以用10 bit查表法一步计算到位，第3步是与固定的$g_L(x)$执行线性加法运算。这两步本质上都是对85 bit的数据做线性加法运算，而线性加法运算满足结合律，故可以将第1步的结果先行与第3步$g_L(x)$进行线性加法运算，而第2步的计算方式

表3　条件2淘汰加扰位情况　　个

执行顺序	左移1 bit	右移1 bit	左移2 bit	右移2 bit	左移3 bit	右移3 bit	左移4 bit	右移4 bit	左移5 bit	右移5 bit
顺序1	173 698	87 677	15 482	10 886	454	362	1 357	1 206	2 454	2 277
顺序2	173 698	87 677	15 482	12 360	359	375	1 101	1 005	1 943	1 853
顺序3	173 698	87 677	13 582	10 886	471	449	1 642	1 529	3 033	2 886

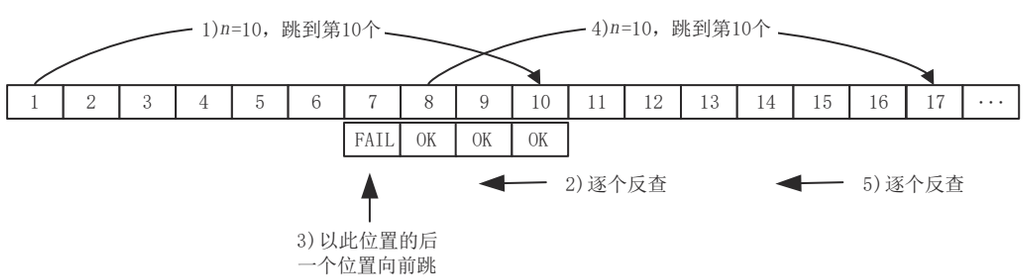

图2　连续合法字检查的贪心算法

其实又可以拆成8组11 bit字（第1组8 bit，其余均为11 bit）分别进行计算，相应的10 bit查表表值也可以改用8组短整型数组进行存储。如此大量计算可提前到额外修正位确定之前计算。在额外修正位确定后，只需简单计算出每一个11 bit字，且每计算出来一个11 bit字就可以立即执行字母表的检查，而不是等到所有校验位（8组字）都计算出来再执行字母表的检查，可显著提高计算效率。

通过对盐通高铁729条无源应答器报文的测试统计，发现第86～93组字按顺序逐个检查有效字的淘汰情况与按逆序逐个检查淘汰的情况一致。即第1个执行的有效字检查总是淘汰1/2，第2个执行的有效字检查淘汰1/4，后续均接近淘汰剩余的1/2。基于此，再结合第86组字还需要与额外修正位的低3 bit进行拼接，在具体实现时，采用逆序执行检查的方式，可进一步减少计算。

3.7 优化Step 7的修正约束条件检查

在Step 7的条件1的有效字检查和校验位并行计算基础上，继续分析优化条件2、条件3、条件4的实现。

1）在Step 4里已经对前84组数据进行了条件2的检查，故在Step 7的条件2检查中可以利用这一结果，在左移1 bit和右移1 bit时，只需检查第82～93组和第1～3组是否存在连续有效字超过2个。在其他移位时，检查第72～93组和第1～11组是否存在连续有效字超过10个。这比常规重新执行全部93组字的条件2检查，可节省三分之二的时间。

2）条件3主要是大量的汉明距计算，即2个11 bit字的汉明距计算，可以借鉴文献［12］中解法五的查表法。只需提前构建11 bit字的汉明距表值，由于最大距离为11，故可用字节数组存储，共需2 048个字节。

3）对条件4漏取样的操作进行分析，可以看到漏取样系数为2时，报文序列变为$b_{n-2},b_{n-4},\cdots,b_1,b_{n-1},b_{n-3},\cdots,b_2,b_0$。由于比特位是循环检查的，故可以将$b_{n-1},b_{n-3},\cdots,b_2,b_0$放至$b_{n-2},b_{n-4},\cdots,b_1$前面，这样前半部分刚好是64个字节，后半部分的最后一个比特位留空。这样可以大大方便代码实现。而漏取样为4的序列，又刚好可以在原先漏取样为2的序列基础上再进行漏取样为2的操作。这样所有漏取样的操作就都可以很方便地用一个循环实现。

在具体实现漏取样的代码时，常规思路是将比特逐个挑出来，组成新的序列。通过分析漏取样的操作特点，再结合一个由4个字节组成的整型数，设计出一次挑出4 bit的快速漏取样操作策略，从而在理论上可比常规思路节省75%的时间。以大端模式CPU（数据的低位保存在内存的高地址，数据的高位保存在内存的低地址）为例，操作流程示意见图3。

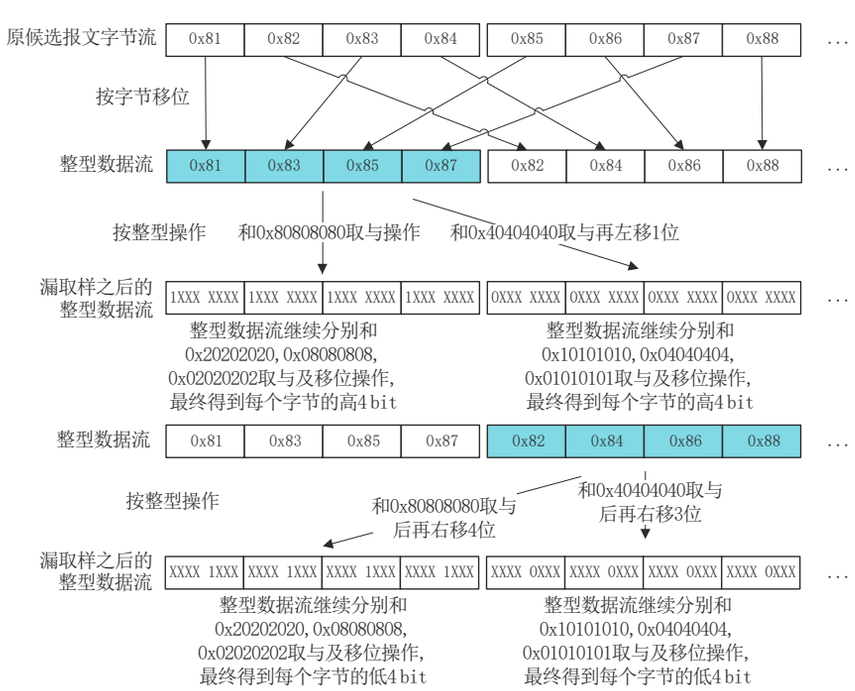

图3 快速漏取样操作流程示意

4　性能测试

将优化后的程序在一台与文献［4］相同CPU主频（2.93 GHz）的PC机上进行测试，对盐通高铁所有729条无源应答器报文获取其对应的首条合法传输报文，有11条报文均只需8 μs，并且仅3条超过了56 μs，分别为58、61、81 μs。与文献［4］给出的获取首条合法传输报文2～6 ms相比，提升了2个数量级。再对盐通高铁729条无源应答器报文获取其全部可能的合法传输报文，最快的一条仅需6.4 ms便可得到全部409条合法报文，最慢的一条也仅用了8.4 ms便得到了全部524条合法报文。

将优化后的应答器报文编码算法程序在STM32F207系列的芯片上进行测试，获取首条合法报文的编码耗时为1～4 ms，与文献［5］采用FPGA实现编码的耗时相当。

5　结论

本文通过分析应答器报文编码算法的各个步骤，采取7个策略对编码算法进行优化。

1）优化之后计算效率得到极大提升，对于现场运用的无源报文，完全可以依据文献［4］提出的优选报文策略，从所有修正约束条件检查通过的合法报文中，选取评价指标最好的报文作为最终的传输报文。

2）对于有源应答器报文，目前各厂家的列控中心均采用专用的编码单元执行实时编码工作，该编码单元如果采用的是FPGA芯片，则可以替换为价格更低的通用CPU芯片。

3）对于安全主控运算能力较强的列控中心设备，可尝试取消专用的编码单元，直接在安全主控运算单元中利用本文的算法实现编码。不仅可降低设备成本、简化设备构成、减少故障点，一定程度上也可提高设备可靠性。

参　考　文　献

［1］ ERTMS/ETCS-Class 4. FFFIS for Eurobalise: SUBSET-036,V4.0.0[S].2023.

［2］ 国家铁路局.应答器传输系统技术条件:TB/T 3485—2017[S].北京:中国铁道出版社,2018.

［3］ 李晓宇,王安.FFFIS编解码算法与实现[J].微处理机,2007,28(6):67-69,72.
LI Xiaoyu, WANG An. FFFIS Coding/Decoding Algorithm and Implementation[J]. Microprocessors, 2007,28(6):67-69,72.

［4］ 张艳宁,赵会兵,全宏宇,等.应答器报文优选及快速编码方法的研究[J].铁道学报,2015,37(2):52-57.
ZHANG Yanning, ZHAO Huibing, QUAN Hongyu, et al. Research on Optimization and Fast Coding Method of Balise Telegram[J]. Journal of the China Railway Society, 2015,37(2):52-57.

［5］ 北京和利时系统工程有限公司.高速铁路实时生成应答器报文的系统和方法:201110421704.2[P].2012-05-02.

［6］ 彭奇.基于FFFIS编码策略和FPGA的应答器报文信号源设计[D].成都:电子科技大学,2013.

［7］ 代萌.简谈列控系统应答器编码策略和编码装置[J].铁路通信信号工程技术,2017,14(3):14-17.
DAI Meng. Balise Coding Strategy and Coding Device of Train Control Systems[J]. Railway Signalling & Communication Engineering, 2017,14(3):14-17.

［8］ 温强.基于FFFIS编码的列控中心应答器报文生成软件优化设计[J].电声技术,2022,46(1):96-101.
WEN Qiang. Design on Message Generation of the TCC Balise Based on FFFIS Optimize Strategy[J]. Audio Engineering, 2022,46(1):96-101.

［9］ 万敏华.应器数据报文编解码的研究与实现[D].成都:西南交通大学,2010.

［10］ 孟令韬,张新明,管伟军,等.应答器报文实时组帧技术的研究[J].铁道通信信号,2014,50(9):11-13.
MENG Lingtao, ZHANG Xinming, GUAN Weijun, et al. Research of Balise Real-time Framing Technology [J]. Railway Signalling & Communication, 2014,50(9):11-13.

［11］ 朱卫华,张奇.应答器报文硬件和软件CRC校验对比分析[J].铁道通信信号,2014,50(9):60-63.
ZHU Weihua, ZHANG Qi. Comparison Between Software CRC Verification and Hardware CRC Verification to Balise Telegram[J]. Railway Signalling & Communication, 2014,50(9):60-63.

［12］ 邹欣,李东,陈远,等.编程之美:微软技术面试心得[M].北京:电子工业出版社,2008.

（责任编辑：吕书丽）