Расширение возможностей CRC-контроля с введением принципа добавления остатков

П. Н. Ерлыков

Петербургский государственный университет путей сообщения Императора Александра I

petrerlikov@mail.ru

Н. С. Ерлыков

Петербургский государственный университет путей сообщения Императора Александра I

nikolaysergeevich.yerlikov@mail.ru

Ю. Я. Меремсон

Петербургский государственный университет путей сообщения Императора Александра I

meremson@list.ru

Аннотация. **В статье рассмотрены различные способы добавления остатков при CRC-контроле. Дано доказательство тождественности CRC-контроля при различных способах добавления остатков. Введено понятие четной и нечетной обработки алгоритмом CRC**

Ключевые слова: CRC-алгоритм, добавление остатков, полином, остаток, делимое, делитель, займ

I. Постановка задачи

Основные принципы CRC-контроля рассмотрены в [1, 2, 3]. При обосновании принципов «выровненного сообщения» [4], появилась догадка о том, что вместо добавления к остатку числа нулей, равного ширине полинома, можно добавлять любое двоичное число с разрядностью, равной ширине полинома.

Добавленная к любому двоичному числу с младшей стороны комбинация единиц и нулей, равная ширине полинома, при обработке того же числа четное число раз алгоритмом CRC «возвращается» на месте добавления.

Проведя ряд опытов, мы убедились в правильности догадки. Осталось малое – доказать правомерность такого предположения.

II. Вариант доказательства

Доказательство проведем на основе примера, приведенного в [5].

В табл. I приведены остатки от деления алгоритмом CRC-4, равного 19.

Если мы дважды обработаем алгоритмом CRC число 560 с добавлением единицы, то добавленная единица возвращается.

Если мы дважды обработаем алгоритмом CRC число 560 с добавлением числа 14, то добавленное число 14 возвращается.

Это свойство удобно применять при CRC-контроле.

Во-первых, мы сможем отойти от добавления нулей связанного с контролем по нулевому остатку недостатком.

Во-вторых, что более важно, мы можем применять свои специальные коды остатков, препятствуя их незаконному добавлению и, тем самым, препятствуя несанкционированному использованию системы приема-передачи.

ТАБЛИЦА I.

Исходное	Делимое	Частное	Остаток
560	718	37	15
561	717	37	14
562	716	37	13
563	715	37	12
564	714	37	11
565	713	37	10
566	712	37	9
567	711	37	8
568	710	37	7
569	709	37	6
570	708	37	5
571	707	37	4
572	706	37	3
573	705	37	2
574	704	37	1

При ширине полинома, равной четырем, количество таких остатков, отличных от нуля, невелико, оно равно 15-ти. Однако, при реально применяемых в современных системах полиномах с шириной, равной 16-ти, это количество возрастает до числа 2^{16}=65535.

Для усложнения несанкционированного доступа можно даже вносить изменения в сами заданные коды, что может быть полезно при передаче специальной информации.

Здесь же заметим, что независимо от того, добавляем мы нули или добавляем иное число, в алгоритм приема-передачи нужно ввести такую опцию, которая говорила бы о том, что при приеме-передаче само преобразование происходило.

Иначе возможен вариант, когда вместо возвращения добавленного остатка после якобы выполненных преобразований, самих преобразований не было, а добавленный остаток просто сохранился при передаче.

В этом случае результат контроля – фиктивен.

III. Переход от остатков без добавления нулей к специально добавляемым остаткам

Для начала выберем последовательность выполнения обоснования правильности нашей догадки.

Как мы считаем, начинать нужно с того, что как при обработке алгоритмом CRC числа с искусственно добавленным остатком (нечетная обработка), так и при обработке алгоритмом CRC числа с промежуточным остатком (четная обработка) частное получается одним и тем же. Это следует из того, что четное и нечетное числа

отличаются друг от друга по величине не больше чем на число 16 при полиноме делителе, равном 19-ти.

Здесь у читателя может возникнуть естественный вопрос: «Раз делимое равно 19-ти, то и остатков, с учетом нулевого, тоже должно быть 19. И определить это можно, деля достаточное количество чисел, идущих подряд».

Дело в том, что делимое в расчетах алгоритмов CRC, является не исходной, а результирующей величиной, получаемой вследствие приведенных расчетов, в которых исходными числами являются сами остатки, участвующие в определении делимого отброшенными займами. Именно поэтому делимых, вызывающих названные числа остатков, при расчетах не возникает.

Для обоснования правильности нашей догадки о возврате добавленных остатков вернемся к тому же способу, который мы применили для обоснования правильности возврата нулевого остатка.

Одновременно замечаем, что отброшенные займы, вызванные вычитанием единицы из нуля в младших разрядах, не совпадают в нечетных и четных алгоритмах CRC.

Получается, что если отбрасываемые займы при нечетном алгоритме CRC вызваны вычитанием единицы из нуля в одних из названных четырех разрядах, то при четном алгоритме CRC такие займы вызваны вычитанием единицы из нуля – в остальных из этих четырех разрядов.

Что касается «выровненного» сообщения, то понятно, что оно является частным случаем такого добавленного остатка и, естественно, возвращение нулевого остатка является так же частным случаем возвращения нулевого остатка, разрядность которого, повторяем, равна ширине используемого полинома.

По этой причине, если будет обосновано «четное возвращение» добавленных остатков, то тем самым будет обосновано и «нечетное возвращение» добавленного остатка, составляющего число нулей, равное ширине полинома.

Для того чтобы начать какое-нибудь опробование, рассмотрим в качестве исходного число 561, пользуясь источником [5]. В таблице I приведена обработка числа 561 алгоритмом CRC с применением полинома 19.

Замечаем, что остаток равен числу 1110(14), то есть четырнадцати.

Давайте заменим четыре младшие двоичные цифры у исходного числа 561, равные величине 0001(1) на величину 1110(14). Теперь получили вместо числа 1000110001(561) число 1000111110, то есть число 574.

ТАБЛИЦА II.

Шаги	9	8	7	6	5	4	3	2	1	0	Степени 2-х
1	1	0	0	0	1	1	0	0	0	1	Число 574
	1	0	0	1	1						Полином 19
				1	0	1	0	0			XOR. Займ 128
2				1	0	0	1	1			Полином 19
						1	1	1	0	1	XOR. Займ 16 и 8
3						1	0	0	1	1	Полином 19
							1	1	1	0	XOR. Займ 14. Остаток 4
Расчет для числа 561					1	0	0	1	0	1	«Частное» 37 Делимое 561+128+16+8+4= 717 717:19=37 с остатком 14 19*37=703 703+14=717

Выполним с полученным числом такую же операцию по алгоритму CRC. В табл. II приведена обработка числа 574 алгоритмом CRC с применением полинома 19.

ТАБЛИЦА III.

Шаги	9	8	7	6	5	4	3	2	1	0	Степени 2-х
1	1	0	0	0	1	1	1	1	1	0	Число 574
	1	0	0	1	1						Полином 19
				1	0	1	1	1			XOR. Займ 128
2				1	0	0	1	1			Полином 19
						1	0	0	1	0	XOR
3						1	0	0	1	1	Полином 19
							0	0	0	1	XOR. Займ 2. Остаток 1
Расчет для числа 574					1	0	0	1	0	1	«Частное» 37 Делимое 574+128+2=704 704:19=37 с остатком 1 19*37=703 703+1=704

С удовлетворением отмечаем, что, как мы и ожидали, новый остаток стал равняться четырёхразрядной младшей части первоначального числа 561.

Займемся попытками доказательства справедливости наших ожиданий.

При нечетном применении алгоритма CRC из-за нулей в разрядах: 2, 3 и 4 возникают займы 4, 8 и 16 и в сумме займ числа 28. Одновременно выполняется подготовка займа, равного числу 2.

В итоге, число, увеличенное в результате применения нечетного алгоритма CRC на число, запланировано уменьшить на число 13, так как +13-(28-2)=13. Эта операция реализуется при четном применении алгоритма CRC.

К пояснению возврата заданного остатка при применении четных алгоритмов CRC можно подходить двумя способами.

Первый способ, более очевидный берет за основу расчет фактических делимых и их изменение при изменении суммы отброшенных займов, возникающих из-за вычитания единицы из нуля в любом из четырех младших разрядов.

Второй способ менее очевидный, о более оригинальный, основан на требуемом изменении числа при реальном учете тех же самых займов, о которых сказано в первом способе.

Попробуем применить первый способ для исходного числа, равного числу 561.

При нечетном применении алгоритма CRC к числу 561, равному числу 560 с добавленным остатком, равным единице, возникают следующие из перечисленных, займы: 4, 8 и 16, составляющие сумму

28. Делимое выросло до числа 717, а остаток стал равным 14-ти, что в двоичном выражении равно 1110, на что обращаем особое внимание. Понимаем, что при четном применении алгоритма CRC к числу 560 с остатком 14 из названных четырех займов возникнет лишь займ из-за вычитания единицы из нуля в первом разряде. Этот займ имеет вес, равный двум. Значит делимое уменьшится на величину, равную разности чисел 28 и 2, то есть 26. С другой стороны, это же делимое уже увеличено на величину 13, равную разности остатков 14-ти и единицы. Значит делимое в итоге уменьшится на величину 13 и станет равным числу 704, что соответствует действительности. Так как частное остается прежним, то уменьшение делимого приводит к уменьшению остатка на ту же величину, то есть на число 13. А именно 14-13=1, что и требовалось показать.

На этом примере мы заметили, что нечетное применение алгоритма CRC подготавливает к возврату заданного остатка при четном применении того же алгоритма.

Материал, рассмотренный на примерах использования полиномов шириной, равной четырем, несложно представлять для полиномов другой ширины, например, 8, 16 и др.

Оставляя возникший вопрос в стороне, покажем, как используя найденные нами закономерности, можно подходить к оценке алгоритмов CRC.

Попытка пояснения алгоритма возврата специальных остатков или иначе: попытка объяснения того, что нечетный алгоритм выделяет промежуточный остаток, а четный алгоритм возвращает специальный остаток.

Прежде всего нужно понять, что специальный остаток из-за добавления 4-х разрядов увеличивает передаваемое число в 16 раз. Кроме того, сам специальный остаток добавляется к передаваемому числу. Например, мы хотим передать на приемный пункт число семь. В качестве специального остатка примем число 12. Это значит, что мы теперь имеем дело с числом 7*16+12=124, а не с числом 7!

Если число 7 само являлось четырехразрядным остатком: 0111, то для поиска остатка у числа 124 нужно применить алгоритм CRC.

IV. Другая попытка пояснения алгоритма возврата спецостатков на приемном пункте

В качестве обоснования возврата «спецостатков» на приемном пункте можно привести два довода, каждый из которых может считаться достаточным. Заметим, что в основе каждого из этих доводов лежат правила используемой полиномиальной арифметики.

Рассмотрим первый из этих доводов.

Таким доводом является равенство частных от делений, выполненных как на передающем пункте, так и на приемном пункте. Равенство этих частных вызвано одинаковой старшей частью делимого без учета четырех разрядной младшей части. К четырехразрядной младшей части относятся: добавленный на пункте передачи «спецостаток» и полученный там же «средний» остаток. Максимальная разница между этими остатками может составлять число 15. Максимально на эту величину могут отличаться друг от друга: делимое на передающем пункте от делимого на приемном пункте. Изменяемая в таких пределах величина делимого не вызывает изменения величины частного. А так как частное от деления как на пункте передачи, так и на пункте приема одно и то же, значит и «спецостаток» должен повториться на пункте приема.

Рассмотрим второй из этих доводов.

В основу этого довода заключено то, что старшая единица полинома – делителя должна непременно вычитаться из старшей единицы очередного уменьшаемого. А это приводит к тому, что младшие 4 единицы делимого независимо от того, принадлежат ли они «спецостатку» или «среднему» остатку, никак в этом процессе не участвуют. Отсюда следует вывод: так как в сочетаниях указанных единиц участвует только делимое (без участия остатков), то, как количество единиц частного, так и расположение этих единиц одинаковое: как на передающем пункте, так и на приемном пункте. В итоге частные на обоих пунктах равны друг другу. А это последнее может быть лишь в случае возврата спецостатка на приемном пункте.

V. Вариант заключения

Все рассмотренные в работе варианты анализа алгоритмов CRC выполнены с учётом отброшенных займов при использовании поразрядной операции XOR на каждом шаге вычисления.

Учет отброшенных займов показывает, что алгоритм CRC не является какой-то особенной CRC арифметикой, как об этом сказано в работе [1], а представляет собой использование известной схемы деления Горнера, в частности, с той особенностью, что за счет вынужденного снижения на единицу двоичной разрядности остатка, количество вариантов остатков в зависимости от используемого полинома несколько уменьшено.

Напомним, что в нашем случае использование полинома, равного 19-ти, количество возможных остатков (с учетом нулевого) снижено с 19-ти до 16-ти.

Несмотря на то, что в данной работе все варианты были выполнены с использованием лишь одного полинома, такие же расчеты и такой же анализ может быть выполнен с применением любого из применяемых и возможных для применения полиномов.

Такой подход открывает для разработчика более широкий взгляд на применение CRC-контроля и предоставляет ему более широкие возможности для такого контроля.

Следует заметить, как применение любого вида контроля, так и применение CRC-контроля не является универсальным.

Наиболее успешно такой контроль находит применение там, где требуется высокая скорость контроля при большом объеме и при высокой скорости передачи информации.

Однако, в тех случаях, когда большой скорости передачи информации не требуется, передаваемый объем данных – невелик, но ошибочный прием данных может иметь катастрофические последствия, следует применять другие виды контроля: неоднократное дублирование с видоизменением дублируемой

информации, использование мажоритарных приемов при передаче и другие способы, опирающиеся не на сложные компьютерные технологии, а на простые, реализуемые аппаратно логические схемы, требуемую надежность которых можно многократно повышать простыми приемами.

Список литературы

[1] Ross N. Williams. Элементарное руководство по CRC-алгоритмам обнаружения ошибок. Текст электронный

[2] Таненбаум Э., Уэзеролл Д. Т.18 Компьютерные сети. 5-е изд. СПб.: Питер. 2012. 960 с.: ил

[3] Теория передачи сигналов на железнодорожном транспорте: учеб. для ВУЗов ж.-д. трансп. / Г.В. Горелов, А.Ф. Фомин, А.А. Волков, В.К. Котов. 2001. 415 с.

[4] П.Н. Ерлыков, Н.С. Ерлыков, Ю.Я. Меремсон. Определение остатков при различных видах CRC-контроля // СПбНТОРЭС, труды ежегодной НТК. 2025. С.193-195.

[5] П.Н. Ерлыков, Н.С. Ерлыков. Правомерность использования схемы Горнера при CRC-контроле // V Бетанкуровский международный форум. Сборник трудов в двух томах. Том 1. 2023. С. 261-265.

[6] Глоссарий простых телекоммуникационных терминов (ГПТТ). https://www.itu.int/rec/T - REC-G.704/en. Версия 17.02.2021 15.29. Текст электронный.