УДК 681.3.06

ОСОБЕННОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ ВЫЧИСЛЕНИЯ КОНТРОЛЬНОЙ СУММЫ CRC32 НА ПРИМЕРЕ PKZIP, WINZIP, ETHERNET

Е.А. Мыцко, А.Н. Мальчуков

Томский политехнический университет
E-mail: 7evgen7@sibmail.com; jgs@tpu.ru

Обсуждаются алгоритмы вычисления контрольной суммы CRC32. Проанализированы описанные в литературе алгоритмы. Из имеющихся в свободном доступе исходных кодов восстановлен алгоритм вычисления контрольной суммы CRC32, который применяется на практике (в PKZIP, WinZIP, ETHERNET). На примере показано, что алгоритм, описанный в литературе и применяемый на практике, дают разные контрольные суммы.

Ключевые слова:
Контрольная сумма, циклический избыточный код, табличный алгоритм, матричный алгоритм, CRC32.

Key words:
Check sum, cyclic redundancy code, table-driven algorithm, matrix-driven algorithm, CRC32.

Мыцко Евгений Алексе-евич, студент кафедры вычислительной техники Института кибернетики ТПУ. E-mail: 7evgen7@sibmail.com
Область научных интересов: программная и аппаратная реализации алгоритмов вычисления контрольной суммы, тестирование и сравнение по быстродействию алгоритмов вычисления CRC.

Мальчуков Андрей Никола-евич, канд. техн. наук, доцент кафедры вычислительной техники Института киберне-тики ТПУ.
E-mail: jgs@tpu.ru
Область научных интересов: помехоустойчивое кодирова-ние, полиномиальные коды, системы проектирования помехоустойчивых полиномиальных кодов, алгоритмы поиска образую-щих полиномов, быстродейст-вующие алгоритмы кодирова-ния и декодирования данных полиномиальными кодами.

Контрольная сумма – некоторое значение, рассчитанное по набору данных путём применения определённого алгоритма и используемое для проверки целостности данных при их передаче. Алгоритм вычисления контрольной суммы (англ. Cyclic redundancy code, CRC – циклический избыточный код) – способ цифровой идентификации некоторой последовательности данных, который заключается в вычислении контрольного значения её циклического избыточного кода [1].

Существуют разные алгоритмы вычисления контрольной суммы и способы их реализации. Стандартный алгоритм подразумевает побитное вычисление контрольной суммы путём деления полинома, представляющего данные, на образующий полином, используемый для вычисления CRC в различных протоколах передачи данных (Ethernet, Bluetooth) и архиваторах (PKZIP, Winzip). Также существует табличный [1] алгоритм, ускоряющий расчёт контрольной суммы, за счёт побайтного вычисления. Далее речь пойдёт об особенностях программной реализации алгоритмов вычисления CRC, используемых в PKZIP, Winzip, Ethernet.

Классическая реализация алгоритма вычисления контрольной суммы

Основой для данного алгоритма является полиномиальная арифметика. Классическая реализация расчёта контрольной суммы заключается в побитовом делении полинома, представляющего данные (файл) на образующий полином. Образующий полином вычисляется в зависимости от сферы применения алгоритма. В данном случае это будет полином 104C11DB7, используемый в PKZIP, Winzip, Ethernet [2].

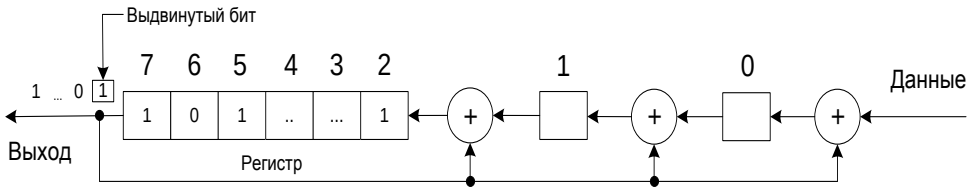

Рис. 1. Схематичное представление работы классического алгоритма

Классический алгоритм можно реализовать с помощью последовательного итерационного сдвига данных в регистре на один бит [3]. В результате проделанных операций в регистре будет находиться контрольная сумма (рис. 1).

Рассмотрим данный алгоритм подробнее. На начальном этапе регистр содержит нулевое значение. На первом шаге происходит сдвиг данных в регистре на один бит от младших разрядов к старшим и добавление к младшим разрядам регистра бита из данных (файла). На втором шаге проверяется выдвинутый бит из старшего разряда регистра. Если бит равен нулю, то осуществляется переход на четвёртый шаг. Если бит равен единице, то осуществляется переход на третий шаг. На третьем шаге выдвинутый бит складывается по модулю два с разрядами регистра 1, 2 и 3, и результат записывается в соответствующие ячейки регистра. На четвёртом шаге проверяется, все ли биты прошли через регистр. Если данное условие ложно, то осуществляется переход на первый шаг. Если условие истинно, то в регистре содержится контрольная сумма. На этом работа алгоритма завершается.

Основным недостатком данного алгоритма является то, что операции выполняются побитно, что значительно замедляет его работу. Поэтому можно применить ускоренные алгоритмы вычисления контрольной суммы, оперируя с блоками бит – байтами. Далее будут рассматриваться различные реализации табличных алгоритмов.

Прямой табличный алгоритм

Прямой табличный алгоритм (как и обратный) основывается на том, что при вычислении контрольной суммы можно оперировать байтами, а также с таблицей, рассчитанной на основе образующего полинома.

Ниже, на рис. 2 схематично представлен прямой табличный алгоритм.

В данном алгоритме используется 4-х байтный регистр, в который будет записываться контрольная сумма.

Рассмотрим прямой табличный алгоритм подробнее. На начальном этапе в регистре содержится значение FFFFFFFF в шестнадцатеричной системе счисления.

На первом шаге осуществляется сдвиг содержимого регистра на один байт влево (от младших байт к старшим) и добавление байта из файла. Данные из файла считываются последовательным сдвигом, начиная со старших байт и заканчивая младшим байтом.

На втором шаге, выдвинутый из регистра байт, используется в качестве индекса в таблице 32-битных значений. На третьем шаге осуществляется сложение по модулю два 32-битного значения таблицы с содержимым регистра. Результат записывается обратно в регистр. На четвёртом шаге проверяется, все ли байты прошли через регистр. Если условие ложно, осуществляется переход на шаг 1. В случае истинности данного условия, в регистре будет содержаться контрольная сумма. На этом работа алгоритма завершается.

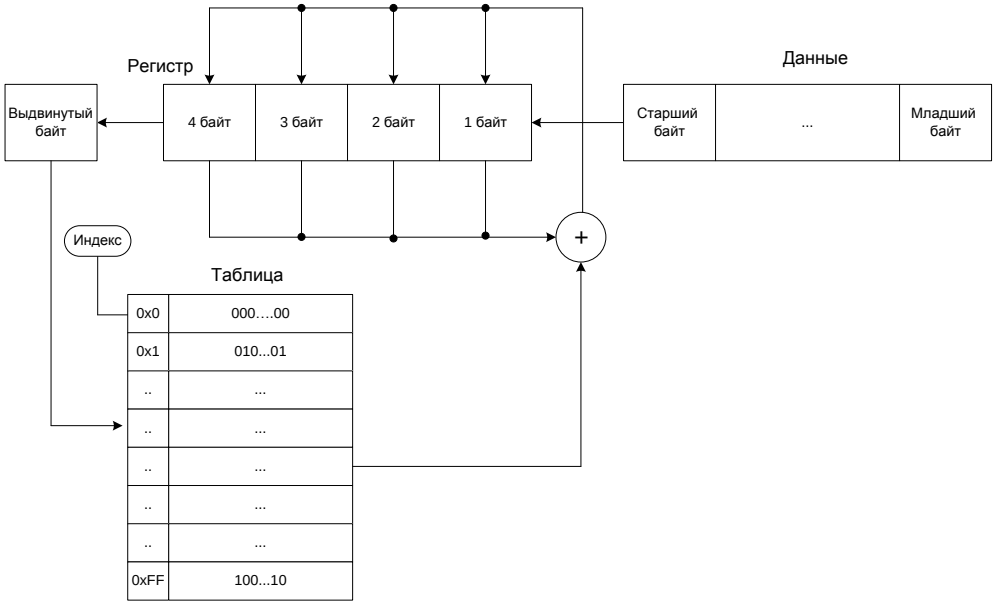

Рис. 2. Схематичное представление прямого табличного алгоритма

Описание данного алгоритма широко встречается в литературе [1]. Контрольная сумма, вычисляемая по прямому табличному алгоритму, не отличается от контрольной суммой, вычисляемой по классическому побитному алгоритму (при условии одинаковых образующих полиномах). Особенностью данного алгоритма является то, что данные считываются побайтно, что значительно ускоряет работу.

Однако на практике (в архиваторах PKZIP, WinZIP и протоколе ETHERNET) [4] используется другой табличный алгоритм, отличающийся от описанного выше. В нём считывание сообщения идёт с младших байт (начала файла) и используется изменённая таблица 32-х битных значений, что, как будет показано далее, приводит к несовместимости результатов вычисления контрольной суммы. В литературе эта модификация алгоритма не встречается. По имеющимся в свободном доступе исходным кодам вычисления контрольной суммы CRC32, совместимой с CRC32 в PKZIP, WinZIP и ETHERNET, нами был восстановлен алгоритм вычисления контрольной суммы CRC32 и назван обратным табличным алгоритмом.

Обратный табличный алгоритм

Основное отличие данного алгоритма от прямого табличного в том, что данные считываются с младших байт файла [4] и используется изменённая таблица 32-х битных значений. Изменения в таблице заключаются в том, что 32-х битные значения таблицы и их положение зеркально отражаются. На рис. 3 схематично представлен обратный табличный алгоритм.

Рассмотрим обратный табличный алгоритм подробнее. На начальном этапе, как и в прямом табличном алгоритме, регистр содержит значение FFFFFFFFh. На первом шаге происходит сдвиг содержимого регистра вправо на один байт (от старших байт к младшим). Данные из файла считываются последовательным сдвигом, начиная с младших байт. На втором шаге выдвинутый байт складывается по модулю два с новым байтом из данных.

На третьем шаге результат сложения используется в качестве индекса 32-битного значения в таблице. На четвёртом шаге осуществляется сложение по модулю два содержимого регистра с табличным 32-битным значением.

На пятом шаге проверяется, все ли байты файла были считаны. Если данное условие ложно, то осуществляется переход на первый шаг. В случае истинности данного условия, в регистре будет содержаться контрольная сумма. На этом работа алгоритма завершается.

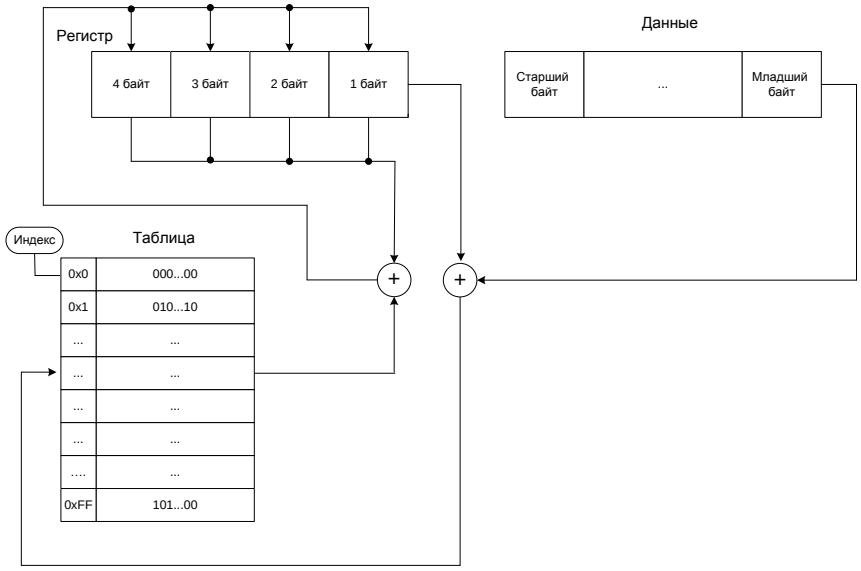

Рис. 3. Схематичное представление обратного табличного алгоритма

Именно этот алгоритм используется на практике при расчёте контрольной суммы CRC32 в архиваторах PKZIP, WinZIP и протоколе ETHERNET [4]. Особенностью данного алгоритма является то, что файл данных считывается с младших байт, что является более удобным. Однако, как уже было упомянуто в описании прямого табличного алгоритма, в этом случае имеются расхождения в результатах расчёта контрольной суммы по сравнению с обычным делением. Для текстового файла, содержащего набор символов 1234567890, по прямому табличному алгоритму контрольная сумма CRC32 равна 6929ACD0, а для обратного табличного алгоритма – 261DAEE5. Для одного и того же файла (picture.jpg), содержащего изображение в формате JPEG по прямому табличному алгоритму контрольная сумма CRC32 равна 569B681, а при обратном табличном алгоритме – FF134C9B.

Заключение

Из имеющихся в свободном доступе исходных кодов восстановлен алгоритм вычисления контрольной суммы CRC32, который используется на практике: в протоколах передачи данных (например, Ethernet, Bluetooth), архиваторах (например, Pkzip, Winzip). На примерах показано, что в результате применения табличных алгоритмов (описанного в литературе [1] и алгоритма, используемого на практике [4]) получаются разные контрольные суммы CRC32 (с одним и тем же образующим полиномом). Это связано с изменённым порядком считывания байтов из файла в алгоритмах. Возможность считывания файла с младших или старших байт требует разного подхода к реализации алгоритма. Рекомендуется при расчёте CRC уточнять, какой алгоритм используется, чтобы избежать разногласий при получении результатов.

СПИСОК ЛИТЕРАТУРЫ

1. Ross N.W. A Painless Guide to CRC Error Detection Algorithms. // Dr Ross Williams. 1993. URL: http://www.ross.net/crc/download/crc_v3.txt (дата обращения: 01.05.2011).
2. Koopman P. 32-Bit Cyclic Redundancy Codes for Internet Applications // Intern. Conf. on Dependable Systems and Networks (DSN). – Washington, July 2002. – Washington, DC, 2002. – P. 459–468.
3. Темников Ф.Е., Афонин В.А., Дмитриев В.И. Теоретические основы информационной техники. – 2-е изд., испр. и доп. – М.: Энергия, 1979. – 512 с.
4. 32 bit Cyclic Redundancy Check Source Code for C++. // Create Window Website. 2011. URL: http://www.createwindow.com/programming/crc32/ (дата обращения: 01.05.2011).

Поступила 14.09.2011.