УДК 681.3.06

ИССЛЕДОВАНИЕ АППАРАТНЫХ РЕАЛИЗАЦИЙ ТАБЛИЧНОГО И МАТРИЧНОГО АЛГОРИТМОВ ВЫЧИСЛЕНИЯ CRC32

Е.А. Мыцко, А.Н. Мальчуков

Томский политехнический университет
E-mail: EvgenRus70@mail.ru, jgs@tpu.ru

Приведено описание аппаратных реализаций матричного и табличного алгоритмов вычисления контрольной суммы CRC32 на программируемых логических интегральных схемах Cyclone фирмы Altera макета SDK-6.1. Показаны особенности аппаратной реализации на примере описания блоков вычисления CRC32 и работоспособность спроектированных устройств на конкретных примерах. Проведено исследование алгоритмов на основе сравнения блоков вычисления CRC32 по занимаемым логическим ячейкам и временным задержкам.

Ключевые слова:
Контрольная сумма, табличный алгоритм, матричный алгоритм, CRC32, аппаратная реализация.

Key words:
Check sum, table-driven algorithm, matrix-driven algorithm, CRC32, hardware implementation.

В литературе [1] можно встретить описание алгоритмов вычисления контрольной суммы CRC32, адаптированных для программной реализации. Однако данные алгоритмы можно адаптировать под аппаратные средства для контроля целостности данных при передаче на различные устройства. В данном случае рассматривается аппаратная реализация CRC32 на примере программируемых логических интегральных схем (ПЛИС) Cyclone фирмы Altera [2] макета SDK-6.1 [3].

Особенности аппаратной реализации алгоритмов

В настоящее время для создания цифровых устройств широко используются ПЛИС. Сферами их применения обычно являются цифровая обработка сигналов, цифровая видео-аудио аппаратура, высокоскоростная передача данных, криптография, коммутаторы между системами с различной логикой и напряжением питания, проектирование и прототипирование ASIC, реализация нейрочипов [4].

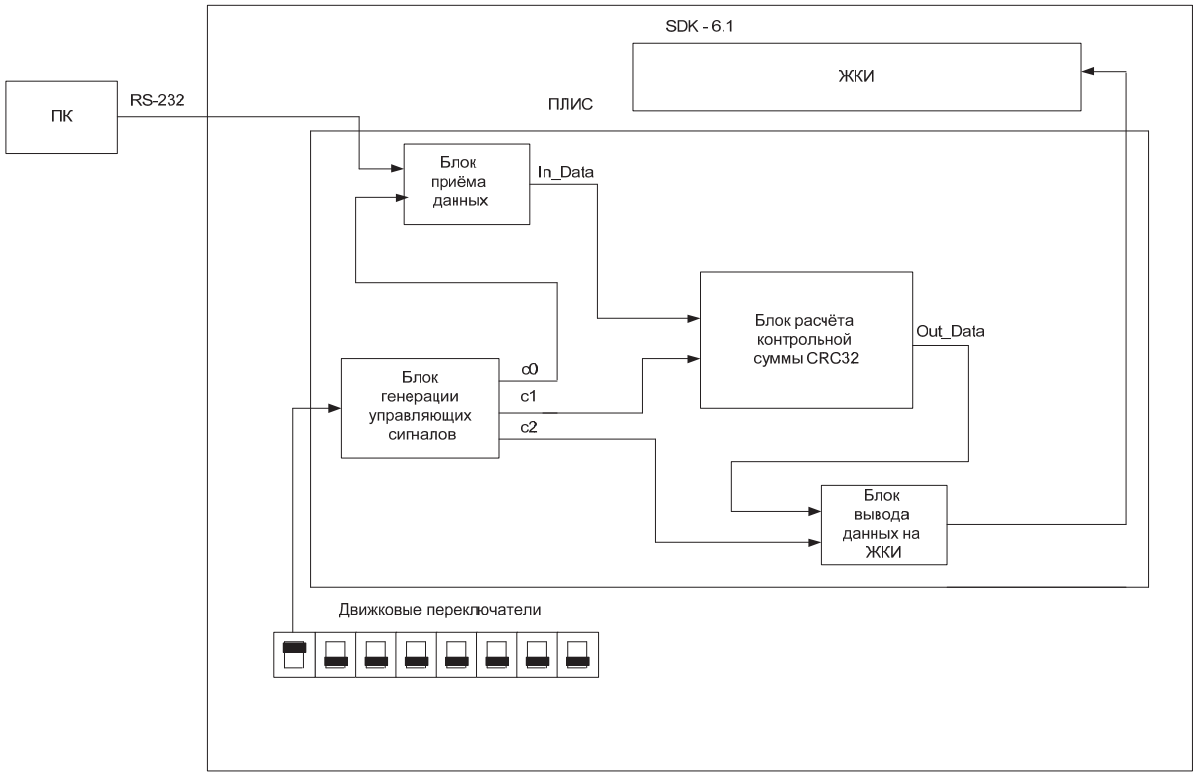

Рис. 1. *Структурная схема устройства расчёта контрольной суммы CRC32*

На кафедре вычислительной техники Томского политехнического университета студенты обучаются разработке устройств на базе современных ПЛИС фирмы Altera [2] в рамках дисциплины «Схемотехника ЭВМ». Для этих целей используются учебные стенды SDK 6.1 [3] на базе Altera Cyclone EP1C3T144, имеющие движковые переключатели, светодиоды, кнопки, двухстрочный жидкокристаллический индикатор (ЖКИ) и последовательный порт. В связи с этим для аппаратной реализации алгоритмов вычисления контрольной суммы CRC32 выбран макет SDK-6.1.

Ввод данных в макет для расчёта контрольной суммы CRC32 производится с персонального компьютера (ПК). Передача данных с ПК на макет осуществляется по последовательному интерфейсу RS-232, используя терминал для передачи данных (term_1b.exe). CRC32 рассчитывается для блока данных и записывается в регистр. Итоговая рассчитанная контрольная сумма отображается на жидкокристаллической индикации (ЖКИ) макета SDK-6.1 при переводе движкового переключателя № 0 (крайний слева) в верхнее положение. На структурной схеме устройства (рис. 1) 4 основных блока. В блоке приёма данных осуществляется приём последовательности битов данных от ПК по интерфейсу RS-232, определение старт- и стопбитов, а также контроль единичного уровня сигнала при передаче. Блок расчёта контрольной суммы вычисляет контрольную сумму CRC32 для данных, поступивших с блока приёма, и записывает её в регистр. Блок вывода на ЖКИ осуществляет перевод значения контрольной суммы в шестнадцатеричную форму и выводит её на дисплей SDK-6.1. При этом во время расчёта CRC32 блок генерации управляющих импульсов осуществляет синхронизацию всех блоков схемы.

Табличный алгоритм

На основе структурной схемы, используя блочно-ориентированный подход, в среде Quartus II спроектирована функциональная схема устройства расчёта контрольной суммы CRC32 с использованием языка описания аппаратуры VHDL [5].

Основным блоком в функциональной схеме является блок расчёта CRC32 «CRC32_vhdl» (рис. 2).

Рис. 2. *Блок расчёта CRC32 табличным алгоритмом*

Блок расчёта CRC32 (рис. 2) содержит 4 входа и 2 выхода. Вход CLK служит для подачи управляющего синхроимпульса частотой 40 МГц на блок. IN_BYTE [7..0] является входным байтом данных для расчёта контрольной суммы. Управляющий сигнал SEND_RECEIVE устанавливается с помощью движкового переключателя, что позволяет задать режим расчёта CRC32 или выдачи результата на ЖКИ. При нулевом уровне сигнала SEND_RECEIVE (нижнем положении переключателя) осуществляется приём данных по байту с последующим расчётом контрольной суммы и записью её в регистр. При единичном уровне сигнала SEND_RECEIVE (верхнем положении переключателя) рассчитанная контрольная сумма поступает на выход CRC32 [31..0] для выдачи её на ЖКИ SDK-6.1. Вход CALC служит для получения сигнала расчёта CRC от блока «main_vhdl» для принятого байта. Выходной сигнал SEND служит для управления блоком преобразования контрольной суммы из символьного представления в шестнадцатеричное.

Матричный алгоритм

Функциональная схема и блок расчёта CRC32 для однобайтового матричного алгоритма выглядят аналогично табличному алгоритму. Отличия заключаются только в описании работы блока вычисления CRC32 на языке VHLD.

Матричный двухбайтовый алгоритм является модификацией однобайтового матричного алгоритма. Для расчёта CRC32 необходимо формировать данные по 2 байта и передавать их на блок расчёта контрольной суммы. Соответственно, блок расчёта CRC32 при аппаратной реализации будет отличаться визуально (рис. 3) и по функциональному описанию.

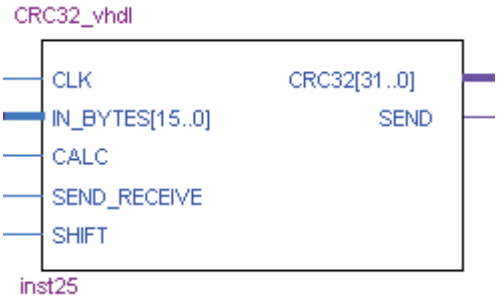

Рис. 3. *Блок расчёта CRC32 матричным двухбайтовым алгоритмом*

Данный блок имеет шестнадцатиразрядный информационный вход IN_BYTES [15..0] и в отличие от приведённого выше блока для табличного и матричного однобайтового алгоритма имеет дополнительный вход SHIFT, который служит для задания режима однобайтового (при единичном уровне сигнала) или двухбайтового (при нулевом уровне) расчёта CRC32. Это связано с тем, что объём данных в байтах не всегда кратен 2, поэтому для оставшихся байтов нужно применять однобайтовую схему расчёта.

В программной реализации особенность матричного четырёхбайтового алгоритма заключа-

лась в том, что сдвиг данных осуществлялся блоками по 4 байта. Поэтому функциональная схема для аппаратной реализации будет отличаться от описанных ранее алгоритмов. Изменения заключаются в том, что из последовательной передачи данных по порту RS-232 нужно формировать блоки данных по 4 байта, для которых будет рассчитываться контрольная сумма. Таким образом, была спроектирована функциональная схема, реализующая расчёт CRC32 данным алгоритмом.

Блок расчёта CRC32 четырёхбайтовым матричным алгоритмом (рис. 4) содержит 5 входов и 2 выхода. Вход CLK, как и ранее, служит для подачи на блок управляющего синхроимпульса частотой 40 МГц. IN_BYTES [31..0] является информационным тридцатидвухразрядным входом для расчёта CRC32. Двухразрядный вход SHIFT [1..0] выполняет функцию выбора способа расчёта CRC32 в зависимости от кратности набора данных. Пока на информационный вход поступают блоки по 4 байта, данный входной сигнал имеет значение «00». Если же на информационный вход поступает неполный блок данных 1, 2 или 3 байта, то на входе формируется соответствующий двухразрядный сигнал расчёта по однобайтовой («01»), двухбайтовой («10») или трёхбайтовой («11») схеме, что необходимо для расчёта CRC32 при наборах данных не кратных четырём.

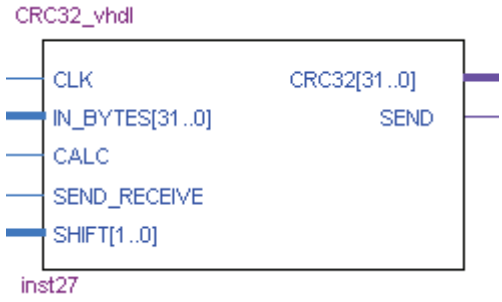

Рис. 4. *Блок расчёта CRC32 четырёхбайтовым матричным алгоритмом*

Примеры расчёта контрольной суммы CRC32

Далее приведены некоторые примеры расчёта CRC32 с применением SDK-6.1. Для расчёта CRC32 был выбран файл CRCfile.exe, который передавался на макет через последовательный порт (рис. 5). Результат расчёта контрольной суммы CRC32 для файла CRCfile.exe на макете SDK-6.1 приведён на рис. 6.

На примере архиватора WinRAR, в котором используется алгоритм CRC32, можно удостовериться в правильности расчёта контрольной суммы (рис. 7).

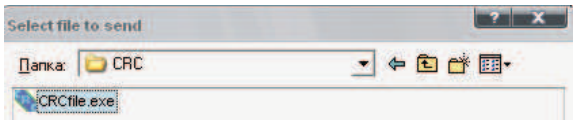

Рис. 5. *Выбор файла для передачи на SDK-6.1*

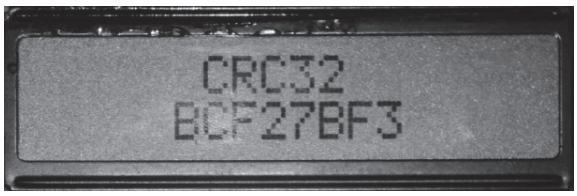

Рис. 6. *ЖКИ SDK-6.1 с контрольной суммой CRC32 для файла «CRCfile.exe»*

Исследование аппаратной реализации алгоритмов

Для проектирования схем в среде Quartus II фирмы Altera, используя блочно-ориентированный подход, применяют два типа логики: комбинаторную и регистровую. Логическая ячейка (Logic Cell) – общий термин для основных блоков микросхем, на которых реализуется проект. Логическая ячейка состоит из элемента регистровой логики (триггера) и универсального четырехвходового логического элемента LUT (Look Up Table), который может быть запрограммирован на реализацию любой четырехвходовой логической функции. Именно на этих элементах реализуется вся комбинаторная логика проекта в ПЛИС. Если нужна функция больше 4-х входов, то используется два таких элемента, больше 7 входов – три ячейки. Регистр (триггер) может быть сконфигурирован как latch (защёлка), D, T, RS, JK триггер или зашунтирован для реализации только комбинаторной логики [6].

В среде проектирования Quartus II для каждого блока функциональной схемы приводится ряд значений, таких как Logic Cells (общее количество ячеек для элемента), LC Resisters (общее количество регистров, входящих в ячейку), LUT-Only LCs (количество элементов только комбинаторной логики, входящих в ячейку), Register-Only LCs (количество элементов только регистровой логики, входящих в ячейку), LUT/Register LCs (количество элементов, использующих как комбинаторную, так и регистровую логику).

Для исследования аппаратной реализации на основе полученных данных была составлена сводная таблица занимаемых логических ячеек для блока вычисления контрольной суммы «CRC32_vhdl» в зависимости от алгоритма (табл. 1).

Как видно из табл. 1, значение Logic Cells состоит из суммы значений LC Registers и LUT-Only LCs, а значение LC Registers, в свою очередь, состоит из суммы значений Register-Only LCs и LUT/Register LCs.

Имя	Размер	Сжат	Тип	Изменён	CRC32
..			Папка		
CRCfile.exe	151 552	60 772	Приложение	15.01.2000 12:20	BCF27BF3

Рис. 7. *CRC32 для файла CRCfile.exe в архиваторе WinRAR*

Таблица 1. *Количество занимаемых логических ячеек для блока «CRC32_vhdl»*

Алгоритм/лог. ячейки	Количество занимаемых ячеек, шт.				
	Logic Cells	LC Registers	LUT-Only LCs	Register-Only LCs	LUT/Register LCs
Табличный	125	65	60	0	65
МС 1 байт	125	65	60	0	65
МС 2 байта	169	65	105	0	65
МС 4 байта	280	66	214	0	66

По полученным данным была составлена таблица изменений количества занимаемых логических ячеек в процентах для аппаратных реализаций различных вариантов матричного алгоритма относительно табличного (табл. 2).

Таблица 2. *Изменение количества логических ячеек относительно табличного алгоритма*

Алгоритм/лог. ячейки	Изменение количества логических ячеек, %				
	Logic Cells	LC Registers	LUT-Only LCs	Register-Only LCs	LUT/Register LCs
МС 1 байт	0	0	0	0	0
МС 2 байта	+35	0	+73	0	0
МС 4 байта	+124	+1,5	+256	0	+1,5

Из табл. 2 видно, что реализация матричного однобайтового алгоритма в данном случае требует столько же логических ячеек, как и для реализации табличного алгоритма.

При реализации двухбайтового матричного алгоритма общее количество логических ячеек увеличивается на 35 %, а количество элементов комбинаторной логики – на 73 %. Для реализации четырёхбайтового матричного алгоритма потребовалось на 256 % больше элементов комбинаторной логики. Данные изменения связаны с особенностью аппаратной реализации матричных алгоритмов. При реализации двухбайтового и четырёхбайтового алгоритмов увеличивается разрядность блока (с 8 до 16 и 32 бит соответственно), который вычисляет контрольную сумму CRC32. Также на блоке расчёта CRC32 появляется дополнительный вход для вычисления контрольной суммы в случаях, когда объём данных не кратен требуемому при реализации определённого варианта матричного алгоритма. Данные факторы приводят к увеличению количества LUT элементов, требуемых для построения блока вычисления CRC32.

Преимущество многобайтовых матричных алгоритмов заключается в уменьшении числа тактов, требуемых для расчёта CRC32. Так, если при передаче данных доступны блоки по 4 байта, то для расчёта CRC32 четырёхбайтовым матричным алгоритмом требуется в 4 раза меньше тактов, чем по табличному алгоритму или матричному однобайтовому алгоритму.

Помимо количества логических ячеек, занимаемых блоком вычисления CRC32, также исследованы временные задержки вычислительных блоков для каждого алгоритма.

На основе полученных данных составлена таблица временных задержек блоков вычисления CRC32 для каждого алгоритма (табл. 3). При этом учитывалось, что если в системе доступно сразу по 2 байта для обработки за такт, то задержка блоков однобайтовых алгоритмов увеличивается в 2 раза. При доступных 4-х байтах для обработки за такт задержка блока двухбайтового алгоритма увеличивается в 2 раза, а для однобайтовых – в 4 раза.

Таблица 3. *Временные задержки для блока вычисления CRC32*

Алгоритм	Задержка при передаче, нс		
	последовательной побайтной	по 2 байта	по 4 байта
Табличный	7,308	14,616	29,232
Матричный 1 байт	7,309	14,618	29,236
Матричный 2 байта	8,627	8,627	17,254
Матричный 4 байта	11,025	11,025	11,025

На основе табл. 3 была составлена таблица ускорений блоков вычисления CRC32 матричного алгоритма относительно табличного (табл. 4).

Таблица 4. *Ускорение блока вычисления CRC32 матричного алгоритма относительно табличного*

Матричный алгоритм	Ускорение при передаче, %		
	последовательной побайтной	по 2 байта	по 4 байта
1 байт	0	0	0
2 байта	−18	+69	+69
4 байта	−50	+32	+165

Как видно из табл. 4, блок вычисления CRC32 матричного однобайтового алгоритма имеет такую же задержку данных, как и для табличного алгоритма, в то время как блок вычисления матричного четырёхбайтового алгоритма отстаёт по скорости от табличного на 50 % при последовательной побайтной передаче данных. В случаях, когда при передаче за цикл доступно больше одного байта, многобайтовые алгоритмы опережают по скорости однобайтовые.

Таблица 5. *Значения времени расчёта CRC32 при аппаратной реализации*

Алгоритм	Значения времени расчёта CRC32 при аппаратной реализации, с					
	Объём использованных файлов, Мб					
	10	210	410	610	810	1010
Табличный	0,076	1,609	3,141	4,674	6,207	7,739
Матричный (сдвиг 1 байт)	0,076	1,609	3,141	4,674	6,207	7,739
Матричный (сдвиг 2 байта)	0,045	0,949	1,854	2,759	3,663	4,568
Матричный (сдвиг 4 байта)	0,028	0,606	1,184	1,762	2,341	2,919

В табл. 5 представлены значения времени расчета для файлов различного объема при асинхронной передаче, исходя из задержки блока вычисления.

Таблица 6. *Значения времени расчёта CRC32 при аппаратной реализации*

Алгоритм	Среднее значение времени расчёта CRC32, с (Доверительные интервалы, %)					
	Объём использованных файлов, Мб					
	10	210	410	610	810	1010
Табличный	0,057 (±1,4)	1,194 (±0,217)	2,299 (±0,203)	3,227 (±0,162)	4,334 (±0,211)	6,184 (±0,155)
Матричный (сдвиг 1 байт)	0,125 (±0,732)	2,81 (±0,112)	5,399 (±0,067)	7,602 (±0,166)	10,23 (±0,161)	14,557 (±0,048)
Матричный (сдвиг 2 байта)	0,092 (±3,882)	2,084 (±0,133)	4,02 (±0,09)	5,645 (±0,105)	7,566 (±0,097)	10,835 (±0,072)
Матричный (сдвиг 4 байта)	0,08 (±0,827)	1,729 (±0,172)	3,32 (±0,011)	4,694 (±0,314)	6,272 (±0,188)	8,965 (±0,064)

На основании значений времени расчета контрольной суммы при аппаратной реализации и значений, полученных при исследовании программных реализаций (табл. 6) [7], была составлена табл. 7, в которой отражена разница по времени расчета CRC32 для программной и аппаратной реализаций.

Таблица 7. *Разница по времени расчета CRC32 для программной и аппаратной реализаций*

Алгоритм	Разница по времени расчета CRC32 для программной и аппаратной реализаций, с					
	Объём использованных файлов, Мб					
	10	210	410	610	810	1010
Табличный	−0,019	−0,415	−0,842	−1,447	−1,873	−1,555
Матричный (сдвиг 1 байт)	0,049	1,201	2,258	2,928	4,023	6,818
Матричный (сдвиг 2 байта)	0,047	1,135	2,166	2,886	3,903	6,267
Матричный (сдвиг 4 байта)	0,052	1,123	2,136	2,932	3,931	6,046

Из табл. 7 видно, что для табличного алгоритма время расчета в программной реализации меньше, чем в аппаратной. Для остальных вариантов алгоритма аппаратная реализации опережает по скорости вычисления программную приблизительно на одну и ту же величину, о чем говорит положительная разница по времени вычисления.

Заключение

На основе алгоритмов расчёта контрольной суммы CRC32, описанных в литературе [1], были спроектированы устройства для аппаратных реализаций алгоритмов на ПЛИС Cyclone. При проектировании функциональной схемы использовался блочно-ориентированный подход (BBD). Было установлено, что для однобайтовых алгоритмов (табличный и матричный) количество задействованных логических ячеек для блока вычисления CRC32 одинаково. Для двухбайтового и четырёхбайтового алгоритмов блок вычисления CRC32 имеет существенные отличия от блоков однобайтовых алгоритмов. Для их реализации потребовалось значительно больше логических ячеек (на 35 % для двухбайтового и на 124 % для четырёхбайтового алгоритма).

По результатам анализа временных задержек блоков вычисления CRC32 можно сказать, что при аппаратной реализации однобайтовые алгоритмы имеют одинаковую скорость вычисления, в то время как у двухбайтового матричного алгоритма задержка блока вычисления на 18 % больше, чем у табличного, а четырёхбайтовый алгоритм отстаёт по скорости на 50 %. Однако при доступном блоке данных, который можно обрабатывать за такт объёмом в 4 байта, матричный четырёхбайтовый алгоритм значительно опережает по скорости (на 165 %) однобайтовые алгоритмы.

Данные аппаратные средства с использованием полученных конфигураций с реализованными алгоритмами расчёта CRC32 позволяют осуществлять контроль целостности данных при передаче по последовательному порту.

Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение 14.B37.21.0457.

СПИСОК ЛИТЕРАТУРЫ

1. Ross N.W. A Painless Guide to CRC Error Detection Algorithms. 1993. URL: http://www.ross.net/crc/download/crc_v3.txt (дата обращения: 01.05.2011).
2. EP1C3T144C8 datasheet // DATASHEET.SU. 2007. URL: http://pdf3.datasheet.su/ Altera/ EP1C3T144C8.pdf (дата обращения: 06.08.2012).
3. Учебный лабораторный стенд SDK-6.1 // Embedded systems. URL: http://embedded.ifmo.ru/index.php/support/sdk-61 (дата обращения: 06.08.2012).
4. Логовский А. Технология ПЛИС и ее применение для создания нейрочипов // Издательство «Открытые системы». 2012. URL: http://www.osp.ru/os/ 2000/10/178242/ (дата обращения: 06.08.2012).
5. Бибило П.Н. Основы языка VHDL. 3-е изд., доп. – М.: Изд-во ЛКИ, 2007. – 328 с.
6. Мяльк Р., Шестаков В. Технология проектирования цифровых устройств на ПЛИС. – СПб.: Изд-во ГУАП, 2005. – 75 с.
7. Мыцко Е.А., Мальчуков А.Н. Исследование программных реализаций алгоритмов вычисления CRC совместимых с PKZip, WinZip, Ethernet // Известия Томского политехнического университета. – 2013. – Т. 322. – № 5. – С. 170–175.

Поступила 09.01.2013 г.