



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
«КУРЧАТОВСКИЙ ИНСТИТУТ»
Институт физики высоких энергий имени А.А. Логунова
Национального исследовательского центра
«Курчатовский институт»

Препринт 2017-8

Е.В. Волков, А.В. Ивашин, В.В. Календарёв, В.Д. Матвеев,
В.П. Сугоняев, Ю.А. Хохлов, А.А. Шумаков

Автономный модуль амплитудно-цифрового преобразования
ADC-32АТС с оцифровкой формы импульса

2017

Аннотация

Волков Е.В. и др. Автономный модуль амплитудно-цифрового преобразования ADC-32ATC с оцифровкой формы импульса: Препринт НИЦ «Курчатовский институт» - ИФВЭ 2017-8. – Протвино, 2017. – 58 с., 16 рис., 5 табл., библиогр.: 30.

В работе представлен 32-канальный модуль амплитудно-цифрового преобразования ADC-32ATC, предназначенный для оцифровки быстрых импульсных аналоговых сигналов. В нём используются микросхемы конвейерных 8-канальных 12-разрядных АЦП AD9222 с быстродействующей схемой выборки-хранения на входе, с высокоскоростным последовательным выходом (от 480 до 780 Мбит/с) и тактовой частотой оцифровки от 40 до 65 МГц. Модуль в комплекте с формирователем входных аналоговых сигналов ориентирован на различные детекторы частиц в составе установок, работающих на ускорителях с циклическим выводом пучка в условиях умеренного радиационного фона. Модуль обладает внутренней буферной памятью 64 МБ, интерфейсом Gigabit Ethernet 1000BASE-T, имеет мёртвое время не более 2.0 – 2.5 мкс. Описаны архитектура модуля, его функции и характеристики.

Abstract

Volkov E. et al. Autonomous analog-to-digital conversion module ADC-32ATC with pulse shape digitizing: NRC «Kurchatov Institute» - IHEP Preprint 2017-8. – Protvino, 2017. – p. 58, figs. 16, tables 5, refs.: 30.

This paper presents a 32-channel analog-to-digital conversion module ADC-32ATC which is intended for digitization of fast analog pulses. For A/D conversion presented module use AD9222 integrated circuit: 8-channel 12-bit pipelined ADC with fast sample-and-hold circuit, high speed serial output links (from 480 to 780 Mbit/s) and from 40 to 65 MHz digitizing clock. The module, equipped with shaper for input analog signals, is intended for various particles detectors in high energy physics experiments operated in moderate ionizing radiation environment at accelerators with cyclic beam ejection. The module has 64 MB internal buffer memory, Gigabit Ethernet 1000BASE-T interface, dead time no more than 2.0 – 2.5 μ s. The module architecture and its characteristics are described.

© Федеральное государственное бюджетное учреждение
«Институт физики высоких энергий имени А.А. Логанова
Национального исследовательского центра
«Курчатовский институт», 2017

Введение

В состав системы сбора данных (ССД) [1] эксперимента ВЕС на ускорительном комплексе У-70 входит большая подсистема амплитудного анализа (около 1600 каналов) на основе интегрирующих АЦП (ИАЦП) ЛЭ-71 [2], в основном — для электромагнитного калориметра [3], а также для черенковских счетчиков и счетчиков мишенной охранной системы. В качестве перспективной замены ЛЭ-71, а также для других применений, разработан 32-канальный автономный¹ модуль амплитудно-цифрового преобразования ADC-32ATC с оцифровкой и записью в событие мгновенного значения напряжения входного сигнала в нескольких точках по времени (в дальнейшем эти оцифрованные значения будут называться выборками, а АЦП с таким принципом работы ВАЦП (SADC) — выборочные АЦП (sampling ADC)). Также разработано несколько вариантов формирователей для согласования/нормализации входных сигналов. Формирователи, далее в настоящей работе называемые также усилителями-формирователями (УФ), применяются для преобразования исходного сигнала от некоторого источника (ФЭУ, генератор и т.п.) в дифференциальный сигнал для подачи непосредственно на схему выборки-хранения АЦП с амплитудой в диапазоне ± 1 В. Также формирователи обеспечивают необходимое растяжение сигнала по времени, для того, чтобы он был покрыт хотя бы несколькими существенно ненулевыми выборками АЦП. Особенностью применяемых в модуле микросхем АЦП является небуферизованная схема выборки-хранения на входе, которая в момент отпираания создаёт заметные импульсные помехи на источник сигнала, поэтому ещё одной целью выходного дифференциального драйвера на выходе формирователя является гашение этих помех за время, равное половине периода тактового сигнала АЦП (подробнее см. в [4]).

Принципиальным отличием описываемого в данной статье АЦП является регистрация последовательности выборок — то есть оцифрованной осциллограммы вход-

¹Под автономностью имеется ввиду то, что модуль не требует для своего функционирования какой-либо магистрально-модульной системы.

ного сигнала, а не одного числа, пропорционального интегралу входного сигнала, как это происходит в интегрирующих АЦП. Это позволяет путём фита осциллограммы («offline», при реконструкции событий) измерять не только амплитуду, но и время прихода входного импульса. Фит осциллограммы также может быть применён для извлечения индивидуальных параметров нескольких наложенных друг на друга импульсов. Эти особенности являются мощным средством для борьбы с несинхронными с изучаемым событием шумовыми импульсами, которые могут быть вызваны паразитной ионной обратной связью в ФЭУ (последимпульсы от предшествующих событий), наложением соседних по времени событий (pile-up). В некоторых случаях открываются возможности различения сигналов разной природы по их форме. Поскольку пьедестал может быть извлечён в каждом событии, то может быть пособытийно учтён его дрейф. Возможна оцифровка сигналов, выходящих по длительности за пределы временного окна измерения (аналог ворот интегрирования), без существенной потери точности. Ценой некоторой потери точности достигается эффективное расширение за пределы динамического диапазона (путём фитирования «обрубленных» сверху импульсов).

С точки зрения удобства эксплуатации серьёзным преимуществом является принципиальная возможность в ВАЦП (в отличие от ИАЦП) организовать цифровую поддержку данных на основе регистров сдвига, которую к тому же можно регулировать в широких пределах прямо в процессе работы. Задержка (порядка нескольких сотен наносекунд) необходима для того, чтобы триггерная система установки успела выработать триггерный сигнал, запускающий приём события. В интегрирующих АЦП для этой цели является обязательным применение длинных (несколько десятков метров) коаксиальных кабелей для подачи входных аналоговых сигналов с присущими им недостатками (повышенные электромагнитные наводки, массогабаритные параметры системы). Также эти кабели вносят заметный вклад в общую стоимость системы ($\sim \$5 - 10/\text{канал}$).

Модуль ADC-32ATC выполнен на современной элементной базе как «надетекторный», полнофункциональный, с автономной системой питания, с программным управлением и расчетом на применение в составе модернизированной ССД ВЕС. Это дает ряд дополнительных преимуществ по сравнению с устаревшей системой на базе ЛЭ-71: кратное уменьшение мёртвого времени, большую гибкость расширения системы, облегчение конфигурирования и настройки, расширение возможностей аппаратной обработки данных, повышение диагностируемости, применение промышленного канала передачи данных (Gigabit Ethernet), минимизацию кабельных соединений, и снижение массогабаритных параметров. Для удобства механического размещения модулей возможно (но не обязательно) использование каркаса стандарта «Евромеханика». Для того, чтобы физически подключить модуль к ССД, требуется всего одно питание (от +8 до +12 В, допустимы пульсации ~ 100 мВ), 2 синхросигнала GATE и START (дифференциальные в уровнях LVDS), один кабель для Gigabit Ethernet 1000BASE-T.

Стоимость модуля ADC-32ATC без учёта формирователей при изготовлении от 50 штук (комплектация, печатные платы, автоматизированный монтаж) составляет

около \$27/канал.

Данная работа описывает в основном конструкцию модуля ADC-32ATC. Помимо задач, возможностей и характеристик модуля, весьма подробно описываются интерфейс микроконтроллера с ПЛИС, форматы данных и всех реализованных команд управления, поэтому данная публикация также может служить пособием для пользователя. Результаты количественного тестирования аналоговых характеристик комбинации ADC-32ATC и формирователя (интегральная и дифференциальная нелинейности и т.п.), а также описание реального опыта работы с модулем в эксперименте ВЕС будут приведены в отдельной публикации.

Далее в тексте приставки мега- и кило- применительно к байтам и размеру микросхем памяти (в байтах или битах) имеют двоичный смысл (2^{20} и 2^{10}), для остальных величин (включая биты, если они не используются для указания размера памяти) — обычный десятичный (10^6 и 10^3).

Работа имеет следующую структуру. В разделе 1 обсуждаются различные схемы запуска (триггера) экспериментальной установки применительно к ИАЦП и ВАЦП, вводится ключевое понятие фазы триггера. Общее описание модуля и процесс его функционирования в рамках ССД эксперимента ВЕС представлены в разделе 2. Архитектура модуля и детализация схемы приведены в разделе 3. В разделе 4 описаны режимы работы модуля x2 и x4 с чередованием АЦП по времени, которые позволяют достичь, соответственно, вдвое и вчетверо большей эффективной частоты оцифровки за счёт пропорционального снижения количества каналов в модуле. В разделе 5 описан высокоточный вспомогательный время-цифровой преобразователь, используемый для измерения фазы триггера. Более подробное описание функций микроконтроллера и ПЛИС в составе модуля даны в разделах 6 и 7, соответственно. В разделе 8 описаны отладочные средства модуля. В разделе 9 обсуждаются вопросы радиационной стойкости модуля. Далее следует краткое заключение. Справочные данные по элементной базе, форматам команд и данных вынесены в приложения. В завершение приведён список ссылок.

1. Сравнение систем запуска (триггера) применительно к ИАЦП и ВАЦП. Понятие фазы триггера

Системы распределения триггерного сигнала (если такое понятие как триггер вообще присутствует в экспериментальной установке) можно разделить на 2 класса (см. рис. 1):

- Асинхронные, когда для запуска регистрации события используется одиночный импульс, один из фронтов которого точно синхронизирован по времени с принимаемым событием (например, вырабатывается схемой совпадений сигналов с пучковых сцинтилляционных счётчиков). Различные приборы используют этот фронт в качестве точки отсчёта для измерения времён регистрируемых ими сигналов (во время-цифровых преобразователях) или для правильной установки по времени ворот для интегрирования сигналов в интегрирующих АЦП.

- Синхронные, когда по всей установке распространяется периодический тактовый сигнал, фронты которого и используются в качестве точных временных отметок. При этом собственно сигнал триггера лишь обозначает конкретный фронт, который прибор должен использовать в качестве точки отсчёта времени. Часто в подобных системах вместе с триггером передаётся также некоторое сообщение, которое идентифицирует триггер, но для наших целей это не важно. Вариантом подобной системы являются также системы, где тактовый сигнал и триггерный физически объединены. Для этого применяется скремблирование или специальное избыточное кодирование типа 8B10B, в результате закодированный сигнал обладает балансом по числу нулей и единиц в среднем, а также всегда содержит достаточное число изменений сигнала в единицу времени при любой передаваемой по каналу информации. Тогда тактовый сигнал извлекается с помощью схемы ФАПЧ (PLL), а триггерные сообщения – это полезная информация. Подобная схема весьма полезна при передаче триггера по оптическому кабелю² (в качестве примера реального применения можно привести эксперимент COMPASS [6]). Ещё одним достоинством синхронных систем является возможность фильтрации джиттера тактового сигнала с помощью схем ФАПЧ для повышения точности синхронизации.

Асинхронные триггерные системы являются естественными при применении интегрирующих АЦП, в то время как для ВАЦП более удобными являются синхронные системы, поскольку для их функционирования необходим периодический тактовый сигнал (по крайней мере при той реализации, что используется в модуле ADC-32ATC). С точки зрения типов физических экспериментов, асинхронные системы более естественны для экспериментов на фиксированной мишени (где обычно есть возможность поставить сцинтилляционные счётчики в пучок и получить тем самым синхронный с событием сигнал запуска). Синхронные системы естественны для коллайдерных экспериментов: там счётчики в пучок поставить нельзя, зато есть периодический сигнал пересечения банчей, синхронно с которым происходят события, этот сигнал естественно использовать в качестве тактового в триггерной системе. Тем не менее, синхронные системы получили распространение и в экспериментах с фиксированной мишенью (см. например уже упомянутый эксперимент COMPASS [6]), поскольку обладают определёнными преимуществами, некоторые из которых приведены выше.

В настоящий момент большинство экспериментов ИФВЭ (в том числе эксперимент ВЕС) используют асинхронные триггерные системы, поэтому для обеспечения возможности работы с такими системами при разработке модуля ADC-32ATC было принято решение использовать для тактирования микросхем АЦП в каждом экзем-

²Баланс по числу единиц и нулей позволяет легко реализовать схему автоматической регулировки усиления (АРУ) в оптическом приёмнике, поскольку амплитуда извлекается простой подачей сигнала на фильтр нижних частот. Схема АРУ является неотъемлемой частью практически любого оптоволоконного приёмника, поскольку разброс по затуханию применяемых оптических кабелей (а значит и разброс мощности оптического сигнала на входе приёмника) может быть очень велик. Подробный обзор этих вопросов можно найти в [5].

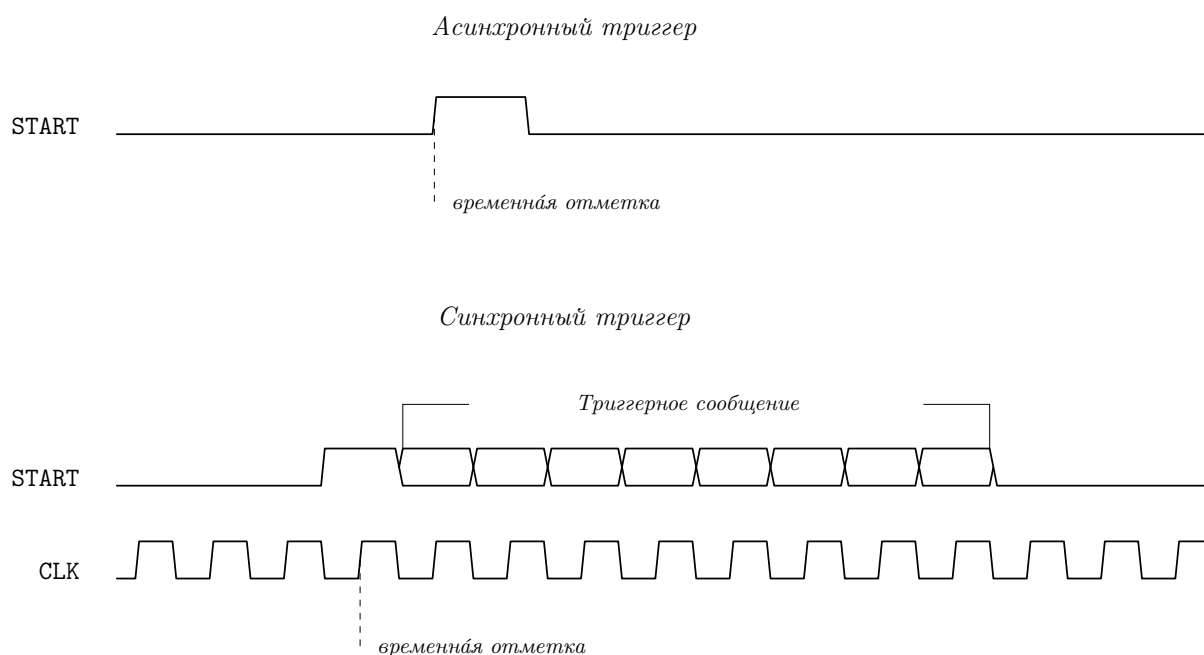


Рис. 1. Два различных типа триггерных систем. Для синхронных систем показан один из возможных вариантов реализации.

пляр модуля свой собственный внутренний тактовый генератор. При этом возникает проблема: прежде, чем поступить в качестве стартового сигнала для записи порции выборок в событие, внешний асинхронный триггерный сигнал должен быть засинхронизирован с внутренней тактовой частотой оцифровки, что вызывает частичную потерю информации о времени прихода триггерного сигнала. С точки зрения анализа информации, записанной в событие, это проявляется во флуктуациях времени прихода зарегистрированного по аналоговому входу модуля АЦП импульса (синхронного с триггером) относительно момента времени, соответствующему начальной выборке в событии. Ширина флуктуаций равна периоду тактового сигнала оцифровки, то есть для 40 МГц это будет 25 нс. Во многих ситуациях это слишком много для того, чтобы надёжно идентифицировать полезный сигнал по времени его прихода на фоне асинхронного с триггером шума. Хотелось бы уменьшить флуктуации до нескольких сотен пикосекунд (именно такой порядок величины имеет в эксперименте ВЕС джиттер триггерного сигнала). Для этого в модуле ADC-32ATC реализован высокоточный время-цифровой преобразователь (ВЦП) на основе вентильных линий задержки с отводами, с помощью которого измеряется так называемая фаза триггерного сигнала (см. рис. 2), с точностью до аддитивной константы равная интервалу времени между передним фронтом подаваемого на модуль триггерного сигнала и передним фронтом его синхронизированной (с внутренним тактовым сигналом оцифровки) версии. Для практического использования фазы триггера (как добавки к времени прихода сигнала, чтобы погасить вышеупомянутые искусственные флуктуации и получить узкий пик в распределении по времени для полезных сигналов) аддитивная константа не

важна — она только сдвигает пик по времени на некоторую постоянную величину. Поскольку положение пика всё равно определяется экспериментально по данным, то константа учитывается автоматически. Время-цифровой преобразователь для измерения фазы триггера реализован на ПЛИС и обеспечивает точность не хуже 100 пс в смысле среднеквадратичного отклонения (экспериментально полученное значение). Подробно этот ВЦП будет рассмотрен в разделе 5.

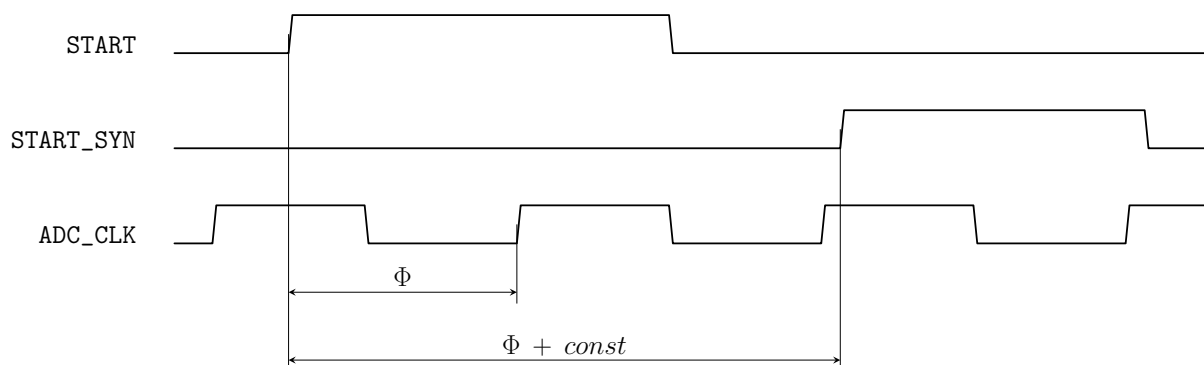


Рис. 2. Фаза триггера при использовании ВАЦП с триггерной системой асинхронного типа. По существу, фаза триггера есть временной интервал от переднего фронта триггерного сигнала START до ближайшего фронта тактовой частоты оцифровки. Но такое определение не годится, если вышеупомянутые фронты почти совпадают по времени. Поэтому практически лучше вместо ближайшего фронта тактового сигнала брать синхронизованный сигнал триггера START_SYN, тогда фаза измеряется с точностью до некоторой аддитивной константы.

Тем не менее, аппаратно в модуль ADC-32ATC добавлен дополнительный вход для подачи внешнего тактового сигнала в уровнях LVDS. Тогда остальные два синхросигнала (GATE и START) можно использовать для подачи триггерных сообщений. Внешний тактовый сигнал можно подать в качестве опорного для ФАПЧ в формирователе тактовых сигналов АЦП AD9516-0 [12] (вместо сигнала с внутреннего кварцевого генератора) и таким образом обеспечить тактирование АЦП очищенным от джиттера внешним тактовым сигналом. Так что в перспективе модуль ADC-32ATC может быть легко модифицирован для работы с триггерной системой синхронного типа.

2. Общее описание модуля ADC-32ATC

Модуль предназначен для цифровой обработки быстрых сигналов детекторов частиц в двухтактном рабочем цикле, определяемым работой ускорителя (см. рис. 3).

В 1-м такте — т.н. СБРОС³ — во время вывода частиц из ускорителя на установку на вход модуля поступает случайный (как правило, пуассоновский) поток аналоговых сигналов, которые подвергаются оцифровке, цифровой обработке, форматированию и буферизации. Во 2-м такте, во время подготовки ускорителя к следующему СБРОСу, буферизованные в течение СБРОСа данные передаются в компьютер ССД. На ускорителе У-70 типичная длительность цикла около 10 с, из них СБРОС занимает около 2 с. Для отладочных целей в модуле предусмотрена возможность внутренней генерации цикла с программируемой структурой.

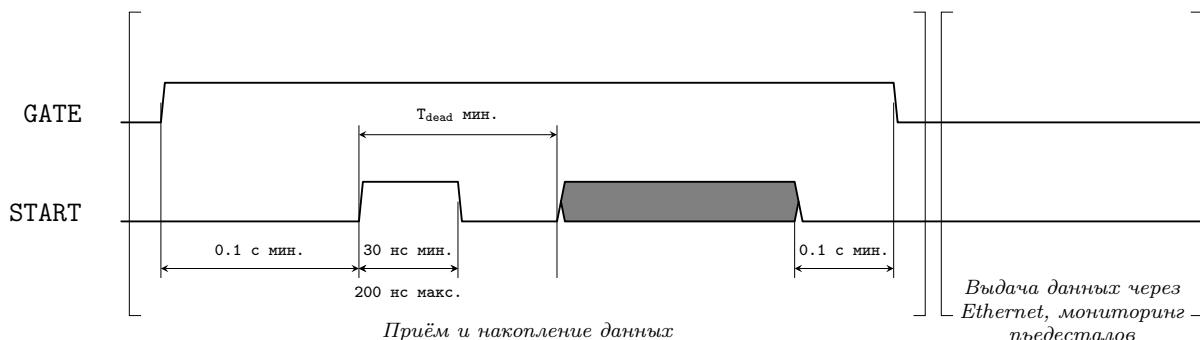


Рис. 3. Временная диаграмма работы модуля ADC-32АТС. Типичная длительность сигнала GATE в экспериментах на ускорителе У-70 составляет 2 – 3 секунды, период повторения GATE — около 10 секунд. Мёртвое время модуля T_{dead} составляет 2.0 – 2.5 мкс (см. раздел 7.5).

Основные функции модуля следующие:

- тактируемая оцифровка аналоговых сигналов, поступающих в течение СБРОСа по 32 каналам (возможно их индивидуальное программное маскирование);
- организация конвейерного потока оцифрованных данных и их отбор из потока по триггерному сигналу с заданной задержкой и длиной последовательности (количеством) выборок;
- обработка и форматирование всех данных модуля, относящихся к одному триггеру, в т.н. «событие»;
- буферизация событий за СБРОС и их передача в компьютер по каналам USB или Ethernet.

Общая структура модуля определяется его основными функциями.

Для оцифровки сигналов по 32-м каналам используются конвейерные АЦП с быстродействующей схемой выборки-хранения на входе с разрядностью 12 бит и

³Мы будем использовать заглавные буквы при написании слова СБРОС, если оно обозначает сброс частиц ускорителем или ассоциированную с ним порцию зарегистрированных данных. В противном случае (например сброс в цифровых схемах) будет использоваться написание строчными буквами.

тактовой частотой оцифровки (в зависимости от исполнения) 40 – 65 МГц. В дальнейшем характеристики, зависящие от тактовой частоты, приводятся для варианта 40 МГц.

Для оцифрованных данных каждого канала устроены конвейеры. Извлечение данных из конвейеров (формирование последовательности выборок) для дальнейшей обработки происходит по общему для всех каналов синхронизированному с конвейером триггерному сигналу, который вырабатывается из входного триггера, задержанного на заданное (программируемое) количество тактов оцифровки сигналов в диапазоне от 7 до 134. Помимо этого длина самих конвейеров может программироваться в диапазоне от 70 до 85 тактов индивидуально для каждого канала, что позволяет подавать на один модуль АЦП сигналы с разной задержкой относительно триггера.

Разрешение приема триггеров и оцифровки сигналов дается одним сигналом ворот (GATE) на всю длительность СБРОСа.

Для буферизации данных за СБРОС используется т.н. память СБРОСа размером 64 МБ (реализована вне ПЛИС), что достаточно для накопления информации не менее чем по 85000 триггерам. Организация потока цифровых данных на всех этапах в течение СБРОСа: конвейеризация, выборка, обработка и буферизация — реализуется в ПЛИС. В разработке загружаемого в ПЛИС проекта использовалась среда разработки QUARTUS-2 Version 9.1 [7]. В основной моде работы в составе ССД данные из памяти СБРОСа выдаются в паузах между СБРОСами в локальную сеть Ethernet. Для этого, а также для общего управления модулем и ряда вспомогательных функций, служит микроконтроллер (МК) модуля.

Для отладки и тестирования модуля предусмотрен альтернативный канал управления и выдачи данных по шине USB2.0 без использования МК (см. раздел 3).

3. Детализация схемы и архитектура ADC-32АТС

На рис. 4,5 приведены функциональная блок-схема и фотография модуля с пояснениями. Информация по электронным компонентам модуля приведена в Приложении А. Используемые АЦП имеют дифференциальные входы с динамическим диапазоном ± 1.0 В. Для восстановления с требуемой точностью из последовательности выборок с конечной тактовой частотой характеристик непрерывного входного сигнала необходима его формировка. Для согласования сигналов по полярности, амплитуде и форме служат специализированные формирователи (УФ), подсоединяемые к ADC-32АТС через многоконтактный соединитель. Через него же на формирователь с ADC-32АТС подается набор питаний (+5, -5, +3.3, +1.8 В), интерфейс SPI и три однопроводные линии интерфейсов 1-wire.

Базовыми архитектурными элементами, реализующими общее функционирование модуля ADC-32АТС, являются: набор входных и выходных сигналов управления, схема формирования тактовых частот для АЦП, интерфейс микросхем АЦП с УФ, интерфейсы ПЛИС с МК и контроллером USB, управление SDRAM памятью СБРОСа, контроллер шины SPI.

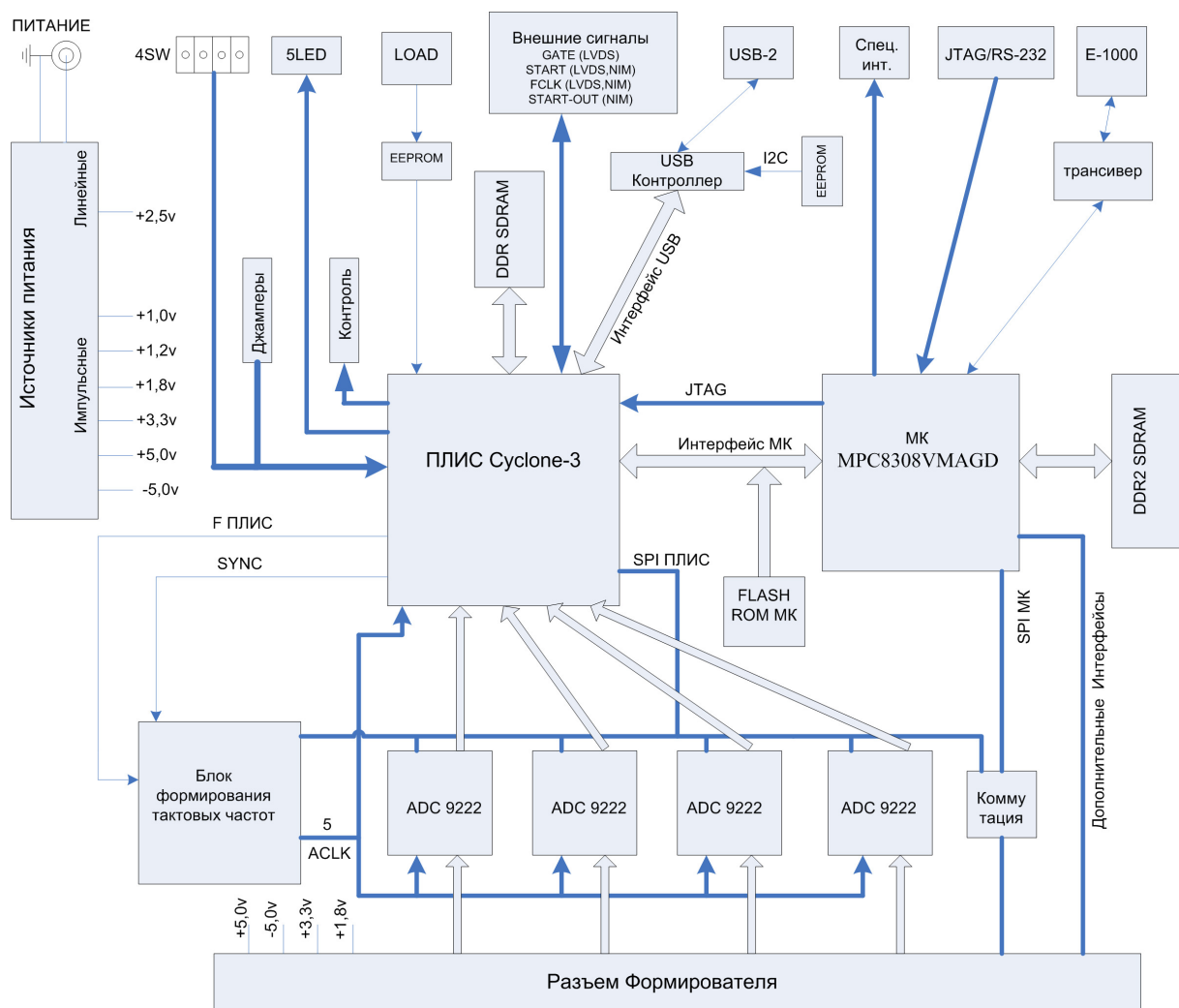


Рис. 4. Блок-схема модуля ADC-32ATC.

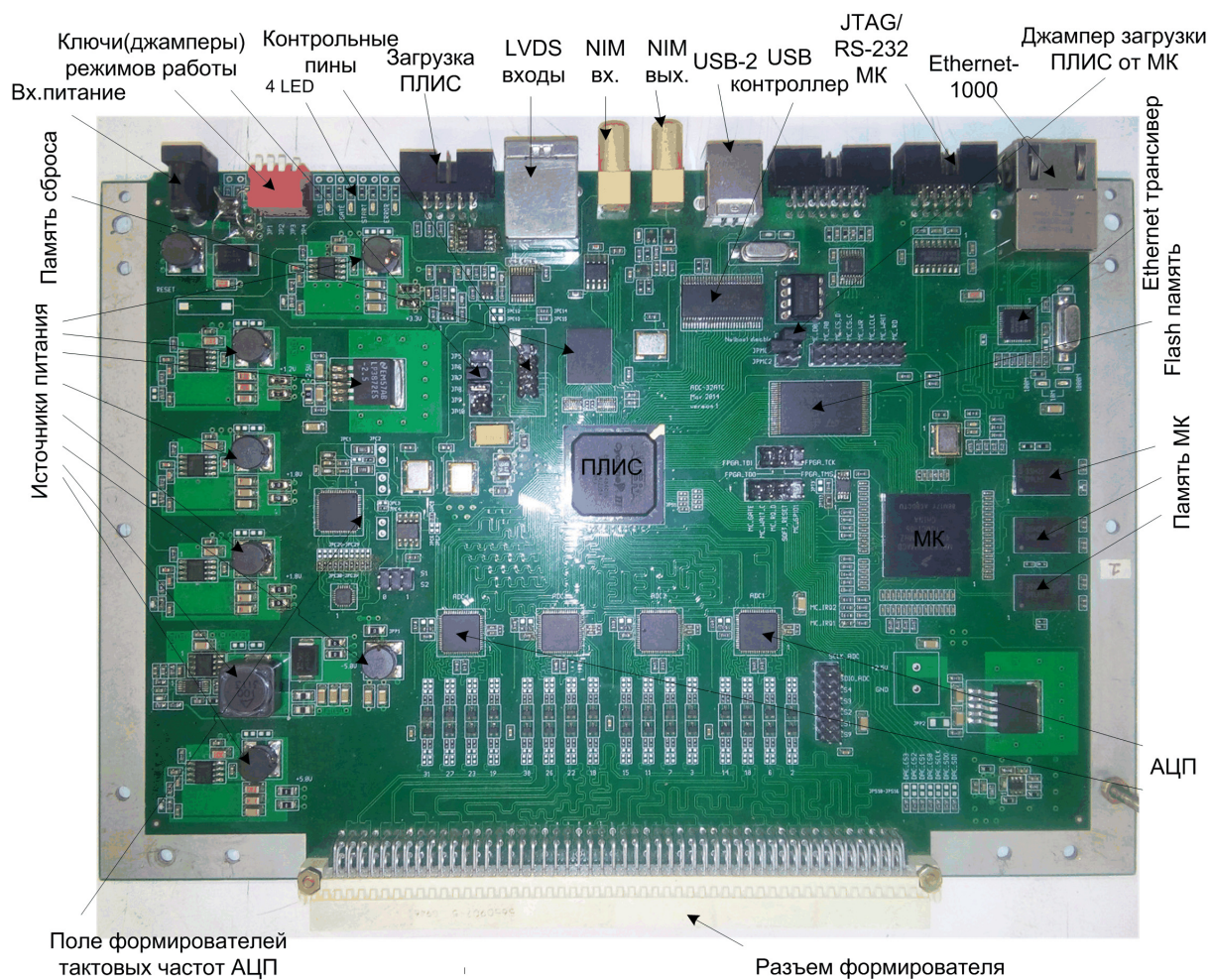


Рис. 5. Фотография модуля ADC-32ATC.

3.1. Внешние сигналы управления

Набор сигналов управления включает основные сигналы в уровнях LVDS и дополнительные сигналы в уровнях NIM (последние — для стендовых применений).

Основные сигналы включают в себя сигнал ворот СБРОСа GATE, сигнал триггера START и входную частоту для схемы формирования тактовых частот АЦП (для будущих возможных применений модуля с триггерными системами синхронного типа). Дополнительные — сигнал входной частоты для схемы формирования тактовых частот АЦП или дополнительного триггера START, и выходной сигнал для выдачи триггера START в режиме его внутреннего формирования (см. раздел 7).

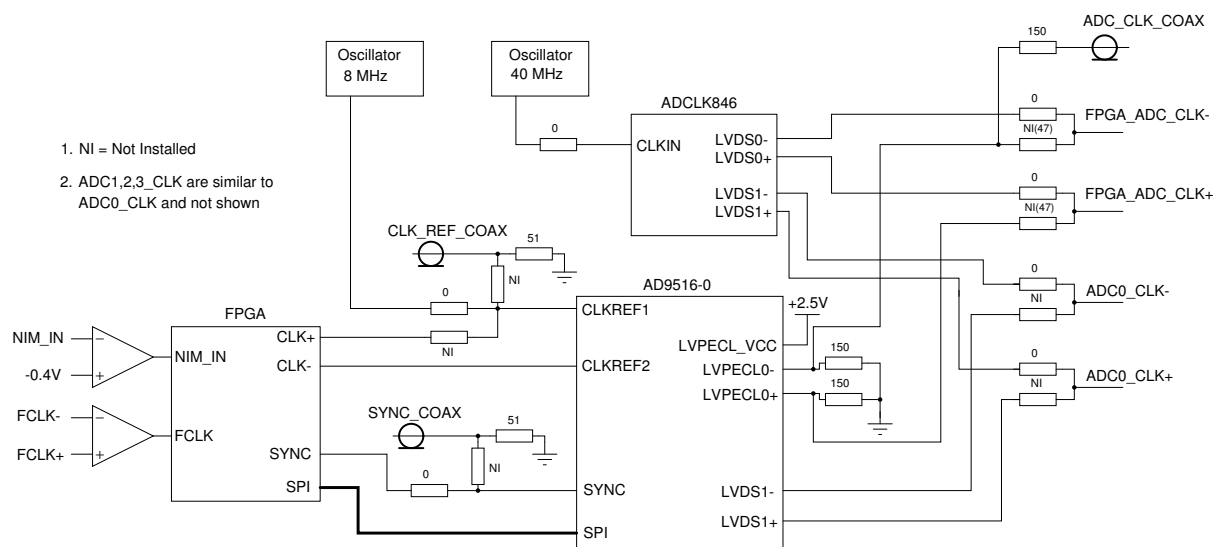


Рис. 6. Схема формирования тактовых частот для АЦП. Показана конфигурация для тактирования АЦП от кварцевого генератора 40 МГц с использованием простого разветвителя ADCLK846. Обозначение NI означает, что элемент не установлен, NI(47) — не установлен, но в случае установки элемента должен использоваться резистор с сопротивлением 47 Ом, а не перемычка 0 Ом. Сигналы ADC1,2,3_CLK генерируются аналогично ADC0_CLK и на схеме не показаны. Сигнал FPGA_ADC_CLK заводится обратно на ПЛИС (FPGA) и в текущей реализации логической схемы в ПЛИС никак не используется.

3.2. Схема формирования тактовых частот для АЦП

На рис. 6 показана схема формирования тактовых частот для АЦП, которая позволяет путем распайки резисторных перемычек и изменением режима работы логической схемы в ПЛИС выбрать один из 5-ти первичных источников тактового сигнала:

- внутренний 40 МГц кварцевый генератор;

- внутренний 8 МГц кварцевый генератор;
- внешний сигнал FCLK в уровнях LVDS из триггерного разъёма 8P8C модуля ADC-32ATC;
- внешний сигнал NIM_IN в уровнях NIM, который подаётся на разъём РЦ-00 модуля ADC-32ATC (этот разъём может использоваться не только для подачи внешней тактовой частоты, его назначение зависит от режима работы схемы в ПЛИС);
- внутренний сигнал частоты от ПЛИС.

Конечным формирователем 4-х тактовых частот для АЦП выступает простой разветвитель ADCLK846 или формирователь с ФАПЧ AD9516-0 [12], выбор также осуществляется распайкой соответствующих резисторных перемычек.

Внешние сигналы должны быть предварительно пропущены через ПЛИС (FPGA) для формирования синхронного с ними импульса SYNC. Это связано с тем, что ГУН (генератор управляемый напряжением, VCO) в AD9516-0 работает на высокой частоте (2.55 – 2.95 ГГц), поэтому даже если частоты опорного входного и выходного сигналов совпадают, то без подачи импульса SYNC их разность фаз будет хаотически меняться от включения к включению. Возможна подача на AD9615-0 от ПЛИС как дифференциального опорного тактового сигнала, так и двух различных однофазных. Со стороны AD9516-0 выбор режима осуществляется программированием по интерфейсу SPI.

Отладочные разъёмы с суффиксом COAX представляют собой 2.54 мм двухштыревые вилки, установленные вблизи микросхемы AD9516-0, и предназначены для прямого подключения к ним 50-омного коаксиального кабеля. С помощью разъёма ADC_CLK_COAX можно наблюдать выходной сигнал AD9516-0, с помощью CLK_REF_COAX и SYNC_COAX подать на AD9516-0 соответствующие сигналы от лабораторного генератора (предварительно отсоединив другие источники сигнала перепайкой показанных на схеме резисторных перемычек).

3.3. Интерфейс микросхем АЦП с УФ

Модуль ADC-32ATC рассчитан на применение в УФ операционных усилителей с двуполярным питанием ± 5 В, поэтому уровень сигналов на выходе УФ может превышать допустимый для АЦП AD9222 диапазон от -0.3 до $+2.0$ В. Хотя на аналоговых входах в AD9222 есть встроенная защита от перенапряжения в виде шунтирующих диодов и стабилитронов, она не предназначена для протекания длительных токов более ~ 10 мА. Существенное превышение указанного значения тока может приводить к разрушению внутренних проводников микросхемы из-за эффекта электромиграции. Поэтому модуль ADC-32ATC содержит дополнительную схему защиты, показанную на рис. 7. Также на входах АЦП предусмотрены посадочные места для элементов R1-7 и C1-3, которые могут использоваться для распайки входного фильтра. Сейчас из этих элементов распаяны только R1-4 номиналы которых указаны

на схеме. Сигналы IN+ и IN− выведены непосредственно на разъём для подключения УФ. При нормальной работе значение входного синфазного сигнала должно находиться в диапазоне от +0.7 до +0.8 В, дифференциального — в диапазоне ± 1 В.

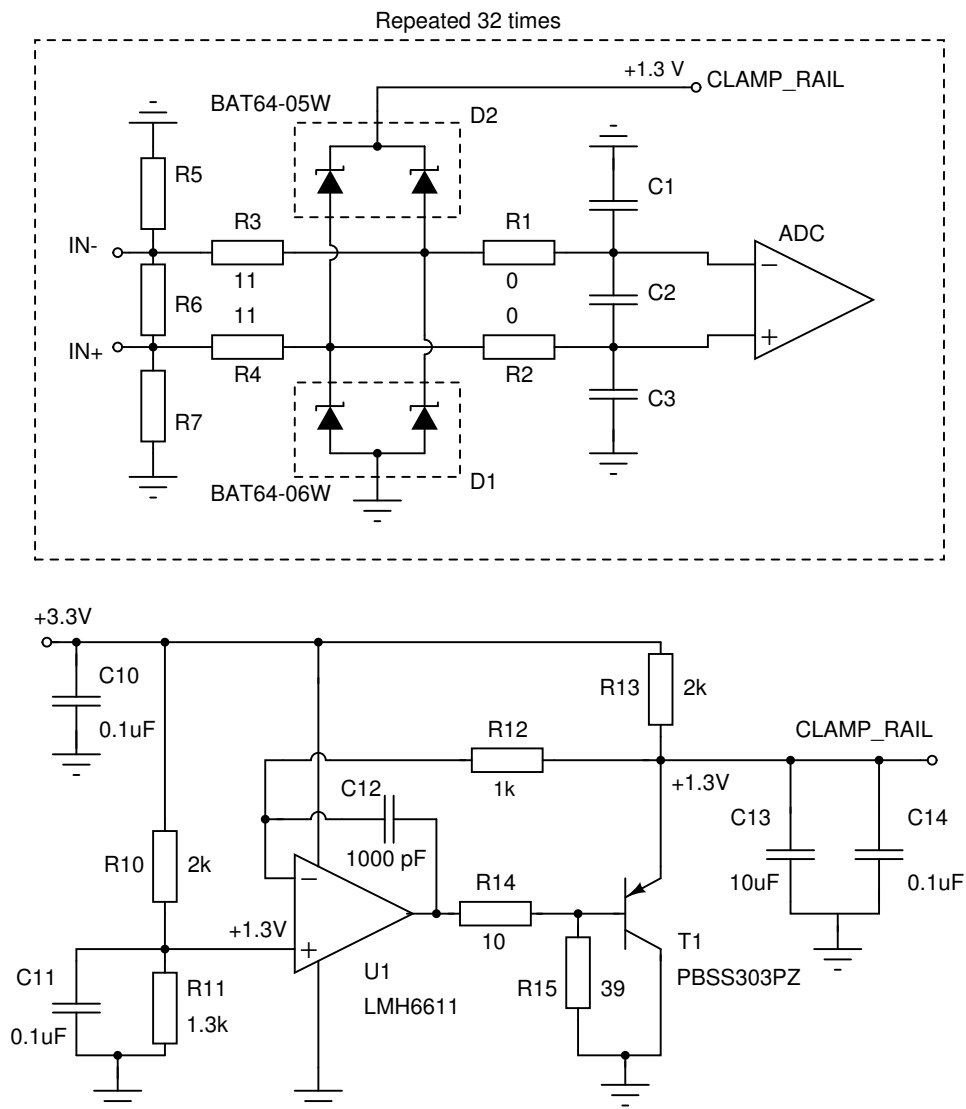


Рис. 7. Схема фильтрации входного аналогового сигнала и защиты от перенапряжения входов АЦП. Внизу показан стабилизатор для выработки требуемого схемой защиты напряжения +1.3 В. Стабилизатор оптимизирован для потребления большого тока и может потреблять до 10 А в пике ($t < 1$ мс) и 1.3 А в постоянном режиме. Особенностью стабилизатора является возможность потребления вышеуказанных токов без подачи питания (для этого установлен резистор R15), хотя напряжение на CLAMP_RAIL при этом не стабилизируется, оно находится в допустимом для работы диодной защиты диапазоне от +0.7 до +1.3 В. Элементы C12, R14 необходимы для частотной коррекции ОУ при работе на ёмкостную нагрузку.

3.4. Интерфейс ПЛИС с микроконтроллером

Интерфейс обеспечивает двухбайтовый канал передачи команд управления от МК в ПЛИС и двухбайтовый канал приема данных от ПЛИС в МК. Каналы имеют общую тактовую частоту (MC_LCLK) 66 МГц, вырабатываемую МК, и совместимую двунаправленную шину данных (MC_D). Сигналы интерфейса описаны в Приложении В.

3.5. Интерфейс ПЛИС с контроллером шины USB

Интерфейс является альтернативным интерфейсу с МК. Предназначается в основном для отладочных целей, но функционально полный и обеспечивает как управление модулем, так и выдачу данных. Контроллер USB работает в режиме USB2.0 (480 Мбит/с). В интерфейсе используется синхронная мода работы на вырабатываемой в ПЛИС тактовой частоте (IFCLK) 48 МГц. Для контроллера USB используется дефолтный дескриптор конфигурации, реализующий для двух используемых Endpoints буферы данных по 512 байт и BULK моду передачи данных. Дополнительная информация по интерфейсу приведена в Приложении В.

Форматы данных и управления для обоих интерфейсов ПЛИС совпадают, кроме кодов управления приборами на шине SPI (см. раздел 7). Выбор того или другого интерфейса в качестве рабочего осуществляется съемной перемычкой на плате модуля. Для тестовых целей в режиме выбора интерфейса с USB возможна программная передача канала управления от USB к МК и обратно.

3.6. Управление динамической памятью СБРОСа

Память СБРОСа реализована на отдельной микросхеме типа DDR SDRAM емкостью 64 МБ. Запись и чтение ведется по последовательным адресам (подобно FIFO). Возможны два режима переключения от записи к чтению данных: по окончании записи всех событий СБРОСа (режим памяти БУФЕР) или постраничное переключение с автоматическим обеспечением минимальной загрузки памяти на уровне 3-х страниц памяти (режим памяти БУФЕР-FIFO). Длина страницы памяти и выбор режима памяти определяются версией проекта в ПЛИС. Режим памяти БУФЕР-FIFO дает возможность ускорения приема данных в ССД, а также организации бесконечного потокового приема данных, но в циклическом режиме работы вводит ограничение на возможность формирования аппаратного заголовка СБРОСа, предшествующего данным. Поэтому для циклического режима, если чтение всего объема данных укладывается в паузу между СБРОСами, целесообразно использовать режим памяти БУФЕР.

3.7. Контроллер шины SPI

Для управления расположенными в модуле программируемыми (по шине SPI) микросхемами АЦП AD9222 и формирователем частот с ФАПЧ AD9516-0 в соответствии с AD рекомендациями [8] в ПЛИС реализован мастер-контроллер шины

SPI с функцией записи и чтения внутренних регистров устройств. Эта шина с помощью перемычек может быть расширена на формирователь аналоговых сигналов при расположении в нем приборов с шиной SPI фирмы AD. На случай применения в формирователе SPI приборов других производителей, или по другим причинам, на те же контакты соединителя с формирователем можно скоммутировать перепайкой резисторных перемычек SPI-шину МК (см. раздел 6.3).

4. Режимы x2 и x4 с чередованием АЦП по времени

Обычно в экспериментальной установке бывает потребность в увеличенной скорости оцифровки для небольшого числа каналов в наиболее загруженных частях детекторов. Например, в эксперименте ВЕС это пучковые черенковские счётчики и околопучковая часть электромагнитного калориметра. Поскольку число таких каналов мало, то с точки зрения затрачиваемых ресурсов и времени не имеет смысла разрабатывать отдельную разновидность модуля АЦП. Выгоднее использовать те же самые модули, что и в остальной части установки, но запустив отдельные микросхемы АЦП в режиме чередования по времени: один и тот же аналоговый сигнал подаётся на N каналов АЦП, которые тактируются со сдвигом по фазе на $2\pi/N$ (следующий относительно предыдущего). При этом эффективное число аналоговых каналов в модуле уменьшается в N раз.

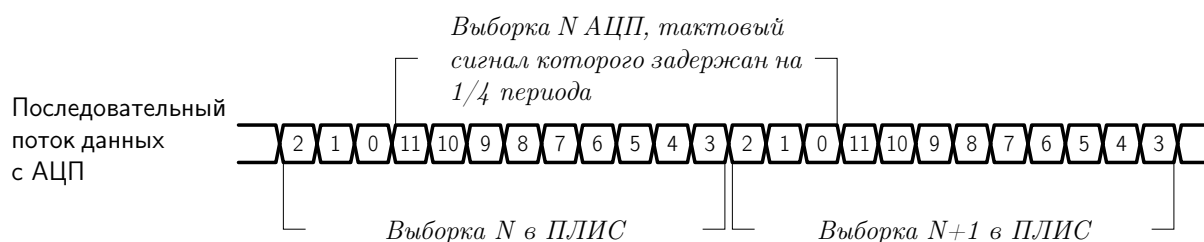


Рис. 8. Приём данных с высокоскоростного последовательного выхода 12-битного АЦП при условии, что тактовый сигнал АЦП задержан на четверть периода, а десериализатор в ПЛИС работает от исходного незадержанного тактового сигнала.

Поскольку модуль ADC-32АТС содержит 4 отдельные микросхемы АЦП, то в нём можно реализовать режимы x2 ($N = 2$) и x4 ($N = 4$). Такая возможность и была предусмотрена при разработке модуля. При этом сдвинутые по фазе тактовые сигналы оцифровки вырабатываются формирователем частот AD9516-0 [12]. В силу того, что применяются 12-битные АЦП с высокоскоростным последовательным выходом, для приёма данных с АЦП в ПЛИС в режимах x2 и x4 можно применить трюк, проиллюстрированный на рис. 8: так как 12кратно 4, то битовые ячейки от всех АЦП совпадают по времени, следовательно, все битовые потоки можно стробировать при приёме в ПЛИС единственным тактовым сигналом. При этом биты от реальной выборки для тех АЦП, которые работают с ненулевым сдвигом по фазе,

будут распределены по двум соседним выборкам в принятых данных. То есть в режимах x2 и x4 можно работать, используя обычную прошивку ПЛИС для режима x1, а потом программно преобразовывать принятые данные. Если последовательность выборок из данных для одного физического канала АЦП находится в массиве `sample[]` длиной `*nsamples`, то следующий код на языке C может быть использован для необходимого сдвига битовой последовательности на `bits` бит:

```
void shift_bitstream(int sample[],int *nsamples,int bits)
{
    *nsamples -= 1;

    for(int i = 0;i < *nsamples;i++)
        sample[i] = ((sample[i] << bits)&0xFFF) | (sample[i+1]>>(12-bits));
}
```

Недостатком программной коррекции является потеря последней выборки для тех АЦП, которые тактируются с ненулевым сдвигом по фазе. Поэтому более предпочтительной является аппаратная коррекция в ПЛИС, которую достаточно легко произвести уже после преобразования битовой последовательности в параллельный код (т.е. после десериализатора) в конвейерном режиме, используя тот же самый по существу алгоритм, что и приведённом выше коде на C. Текущая реализация проекта в ПЛИС поддерживает эту аппаратную коррекцию, при этом в заголовке СБРОСа (см. Приложение D) модуль устанавливает определённые флаги, сигнализирующие о работе в режиме чередования. В остальном формат данных не меняется (т.е. по-прежнему в событиях будут фигурировать 32 физических канала), но построитель события может использовать флаги из заголовка СБРОСа для программного объединения каналов АЦП, которые соответствуют одному аналоговому сигналу. Отсечение каналов по порогу срабатывания (см. раздел 7.4) в режимах x2 и x4 происходит следующим образом: если хотя бы один канал из объединяемой группы срабатывает, то в событие записывается вся группа. Если хотя бы один канал из объединяемой группы отключён (путём отправки соответствующей команды по каналу управления), то вся группа отключается (не записывается в событие). Реализованная логическая схема в ПЛИС может свободно переключаться между режимами работы x1, x2, x4 отсылкой соответствующей команды по каналу управления (см. Приложение C).

На рис. 9 показан реальный сигнал, оцифрованный в режиме x4 при тактировании отдельных микросхем АЦП частотой 40 МГц (т.е. с эффективной частотой оцифровки 160 МГц), для сравнения в том же масштабе по времени показан тот же сигнал, оцифрованный в режиме x1 (т.е. взяты выборки только с одного АЦП). При этом использовалась следующая схема для подачи аналогового сигнала: сигнал пассивно разветвлялся на 4 и каждый из этих 4 сигналов подавался на отдельный УФ. Эта схема плоха тем, что разброс в параметрах времяформирующих RC-цепей УФ приводит к тому, что каждый АЦП оцифровывает несколько отличающийся по форме сигнал. Этого можно избежать, используя специальные УФ с разветвлением уже сформированного сигнала при помощи 4 полностью дифференциальных повторителей. По-

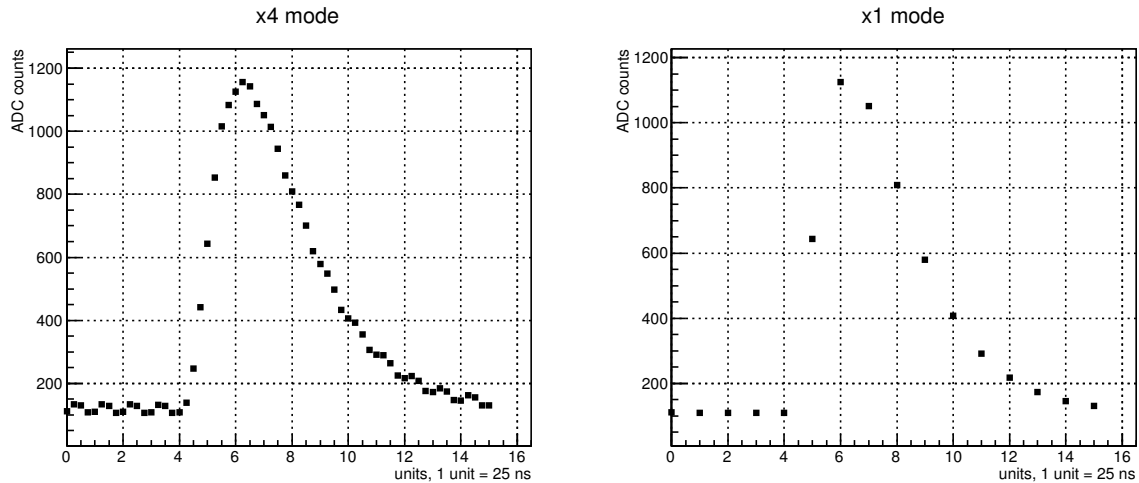


Рис. 9. Реальный импульс, оцифрованный в режимах x4 и x1. В режиме x4 явно видна модуляция с периодом 25 нс, вызванная неодинаковостью пьедесталов в используемых четырёх каналах АЦП.

вторители нужны для исключения взаимного влияния схем выборки-хранения от 4 объединяемых каналов АЦП. Тем не менее, останется вопрос о несколько отличающихся пьедесталах (этот эффект хорошо виден на рис. 9), коэффициенте усиления и неидеальной разности фаз тактовых сигналов АЦП. Все эти отличия планируется учесть при помощи калибровки прибора (на тестовых сигналах и прямо на реальных данных).

5. Высокоточный вспомогательный ВЦП для измерения фазы триггера

Для измерения фазы триггера (см. раздел 1) используется ВЦП на основе вентиляльной линии задержки с отводами (ЛЗО, tapped delay line). В таком ВЦП входной сигнал запускается в цепочку последовательно соединённых идентичных вентилях, к выходу каждого вентиля подсоединяется D-триггер, все D-триггеры тактируются периодическим тактовым сигналом. В результате на каждом положительном фронте тактового сигнала в D-триггеры запоминается состояние линии задержки, которое при наличии в ней сигнала имеет вид типа (сигнал запускается со стороны младших битов) 0000000001111111. Далее подсчитывается число единиц (в данном случае оно равно 8), которое пропорционально интервалу времени от переднего фронта входного сигнала до того фронта тактового сигнала, которым запоминалось состояние задержки в триггеры. Фронты тактового сигнала отсчитываются с помощью счётчика. Таким образом, комбинация счётчика и вышеописанной схемы измерения времени прихода сигнала относительно ближайшего фронта тактовой частоты на основе ЛЗО представляет собой полноценный ВЦП.

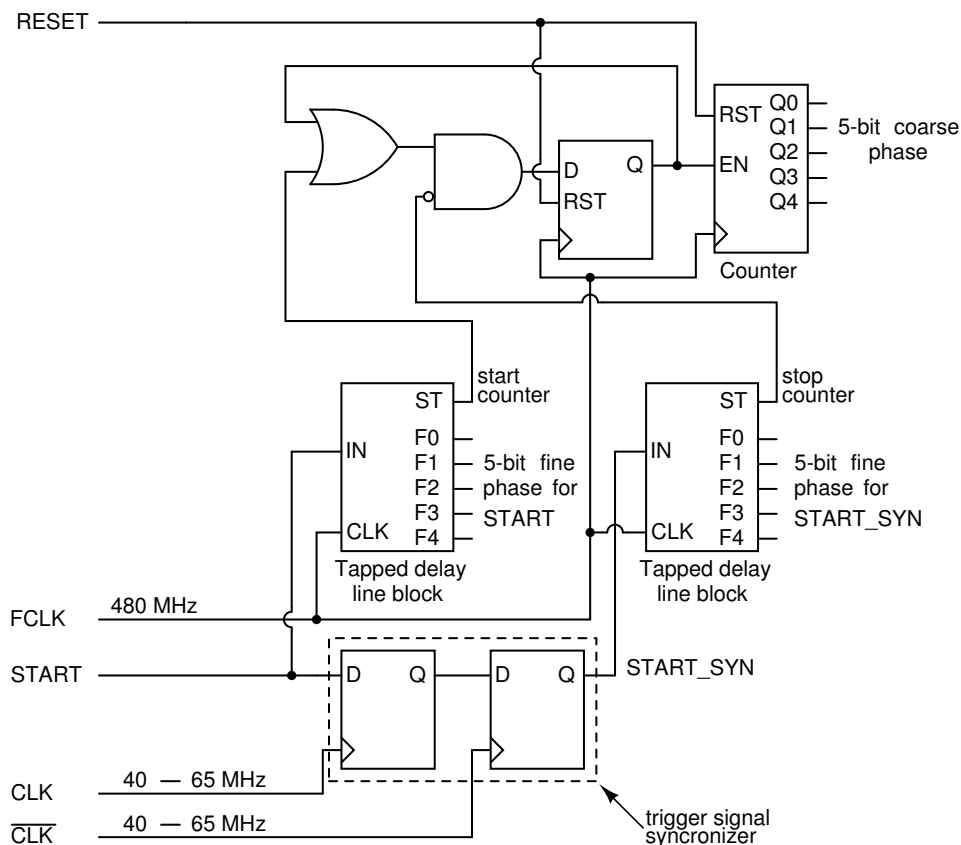


Рис. 10. Схема ВЦП для измерения фазы триггера. Сигнал START — это сигнал триггера, который подаётся на модуль ADC-32ATC, CLK — сигнал, синхронный с тактовым сигналом АЦП, которым тактируется основной конвейер данных от АЦП, $\overline{\text{CLK}}$ — инверсная версия CLK, RESET — сигнал сброса, который подаётся в некоторый момент до обработки события, FCLK — несинхронный с сигналом CLK быстрый тактовый сигнал. Через несколько десятков наносекунд после прихода START на трёх 5-битных выходах устанавливаются постоянные значения: грубая фаза Φ и две точные поправки к ней — ϕ_t (для сигнала START, «t» от слова «trigger») и ϕ_s (для сигнала START_SYN, «s» от словосочетания «synchronized trigger»). Эти три величины записываются как есть в заголовок события (см. Приложение D). Устройство блоков на основе вентильной линии задержки с отводами (tapped delay line) показано на рис. 11.

Создание однородной ЛЗО на ПЛИС типа FPGA представляет значительные трудности, поскольку, как правило, пользователь явно не управляет размещением и разводкой компонентов ПЛИС, на которых реализуется заданная пользователем логика. Более того, средства синтеза схем для ПЛИС производят агрессивную оптимизацию комбинационной логики; особенно это касается инструментария Quartus [7] для ПЛИС производства Altera/Intel: иногда оптимизация не выключается, даже если при задании схемы явно пользоваться примитивами, на которых физически реализуется комбинационная логика.

Тем не менее, существует подход [14], который практически применим к большинству ПЛИС: использование в качестве ЛЗО цепи переноса в сумматоре. Почти все производители ПЛИС предусматривают некоторые дополнительные к основной структуре ПЛИС аппаратные возможности для облегчения реализации сумматоров. Сумматор, синтезированный с применением этих дополнительных возможностей, всегда обладает фиксированной физической структурой внутренних цепей, и его можно использовать в качестве ЛЗО с предсказуемыми характеристиками.

Но даже при использовании для ЛЗО сумматоров необходимо предпринять дополнительные меры, которые в нашем случае (ПЛИС Cyclone III [15]) были такими:

- На входах сумматора необходимо задать постоянные сигналы 0 и 1 (при этом вход переноса будет входом ЛЗО, выходы сумматора – выходами (отводами) ЛЗО), но оптимизация приводит к удалению сумматора с постоянными значениями на входах в физически реализованной схеме. Поэтому мы подавали на входы некоторые реальные сигналы, которые во время работы принимали необходимые постоянные значения.
- Логические ячейки (LUT+триггер) упакованы в Cyclone III в LAB'ы (logic array block) по 16 штук. Для построения требуемого нам 63-битного сумматора необходимо 4 соседних по вертикали LAB. Для этого использовались ограничения на физическое расположение компонентов схемы внутри кристалла. К выходу сумматора в каждой логической ячейке подключался D-триггер, так что и здесь образовывалась структура с предсказуемыми задержками.

Реализованная нами схема ВЦП показана на рис. 10,11. Для уменьшения аддитивной константы фазы триггера (см. раздел 1) второй D-триггер синхронизатора сигнала START тактируется инверсным тактовым сигналом. Поскольку интервал между соседними сигналами START достаточно велик (не менее нескольких сотен наносекунд), то мы не стали реализовывать конвейерный кодер для преобразования выхода ЛЗО в отсчёты ВЦП, а ограничились полностью комбинационным.

В реальности задержки отдельных элементов ЛЗО отличаются друг от друга, также неизвестным является конкретное значение задержки элемента. Поэтому для использования данного ВЦП необходима калибровка, которая позволяет также учесть температурные изменения задержки. Калибровка ВЦП делается «offline» с использованием статистической обработки данных, при этом возможно использование как специальных тестовых данных, так и реальных. Для этого заметим, что если входной

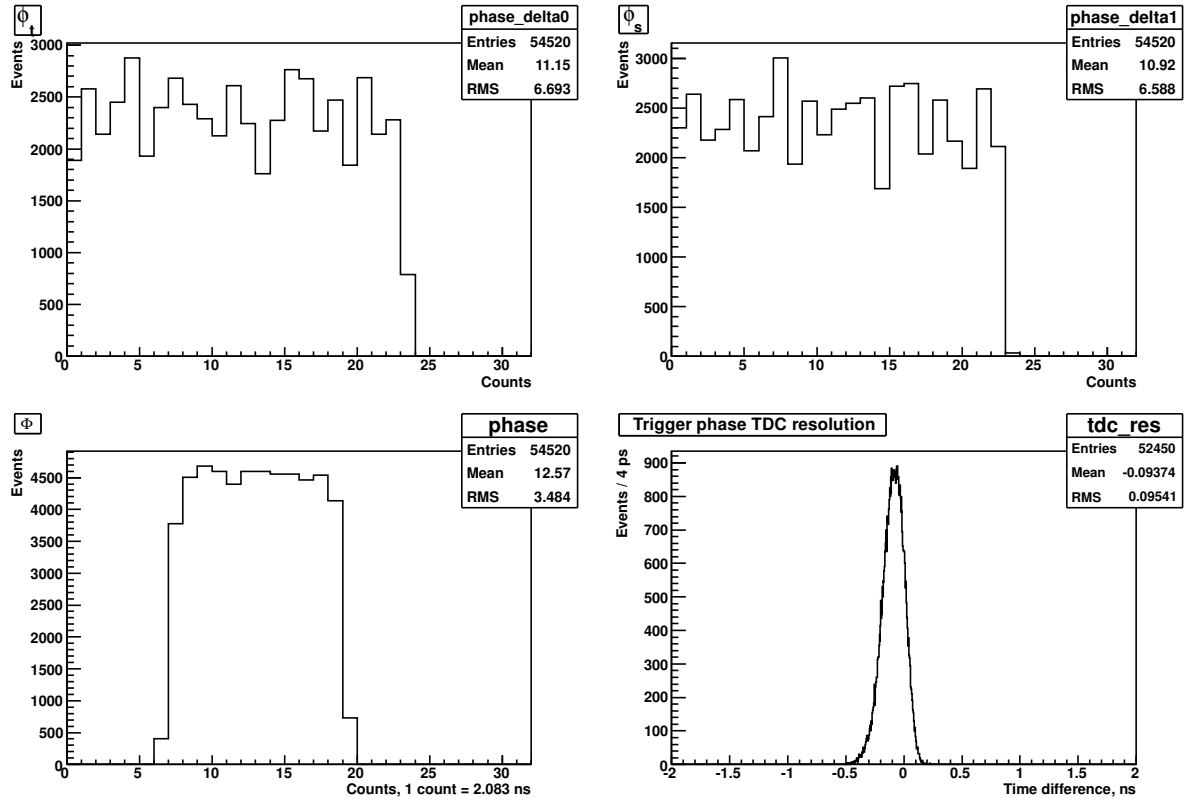


Рис. 12. Гистограммы для величин ϕ_t , ϕ_s , Φ из заголовка события и гистограмма с разностью фазы триггера, полученной с помощью ВЦП (из величин ϕ_t , ϕ_s , Φ) и с помощью ВАЦП (фитом массива выборок синхронного с триггером сигнала, поданного на один из аналоговых входов модуля ADC-32ATC). Видно, что разрешение ВЦП не хуже 100 пс.

сигнал распределён равномерно по времени (это хорошее предположение даже для сигнала START_SYN, поскольку частота 480 МГц несинхронизована с тактовой частотой АЦП), а задержки элементов ЛЗО одинаковы, то гистограмма величины ϕ будет плоской (здесь и далее мы будем называть любую из величин ϕ_t или ϕ_s как ϕ , если соответствующее утверждение применимо для любой из них). В реальности это, естественно, не так, и высота бина в гистограмме пропорциональна задержке соответствующего элемента ЛЗО (чем больше задержка — тем больше вероятность обнаружить перепад входного сигнала именно в этом элементе задержки). На рис. 12 показаны измеренные распределения для ϕ_s и ϕ_t для одного из модулей ADC-32ATC. По гистограмме вычисляется функция преобразования $F(\phi)$ следующим образом:

$$F(\phi) = \sum_{i=0}^{\phi} h_i / \sum_{i=0}^{31} h_i$$

где h_i — значение в бине i гистограммы.

Собственно тонкая фаза триггера Φ_{fine} в наносекундах вычисляется с использованием вышеописанных функций преобразования $F(\phi)$ по формуле:

$$\Phi_{fine} = 2.083 \cdot (\Phi - F_t(\phi_t) + F_s(\phi_s))$$

где 2.083 нс — период тактовой частоты 480 МГц.

На рис. 12 приведена гистограмма с разностью фазы триггера, полученной вышеописанным способом, и с помощью ВАЦП (фитом массива выборок синхронного с триггером сигнала, поданного на один из аналоговых входов модуля ADC-32ATC). Видно, что разрешение ВЦП не хуже 100 пс в смысле среднеквадратичного отклонения. При этом надо иметь ввиду, что при фите сигнала использовалось достаточно грубое приближение к реальной форме сигнала, так что полученную величину разрешения следует рассматривать как оценку сверху. На среднее гистограммы обращать внимание не надо, как уже упоминалось, ВЦП измеряет фазу триггера с точностью до некоторой аддитивной константы. Частота тактового сигнала АЦП при измерениях составляла 40 МГц. Более детально экспериментальное тестирование ВЦП для фазы триггера будет обсуждаться в упомянутой во Введении отдельной работе.

6. Микроконтроллерное управление и передача данных

6.1. Общее описание и характеристики

В модуле ADC-32ATC используется 32-разрядный МК MPC8308 [17] производства Freescale/NXP с архитектурой PowerPC, тактовой частотой 400 МГц и 256 МБ оперативной памяти DDR2-266 с 32-разрядной шиной данных (с поддержкой ECC). Для связи с внешним миром применяется Gigabit Ethernet 1000BASE-T на витой паре. Микроконтроллер подсоединён к ПЛИС посредством 16-разрядной шины eLB с тактовой частотой 66 МГц и нескольких сигналов GPIO. Один сигнал GPIO используется для подключения температурного датчика модуля ADC-32ATC по протоколу

1-wire. Ещё несколько сигналов GPIO используются для организации шины SPI и 3-х интерфейсов 1-wire для управления формирователем аналоговых сигналов. Интерфейс JTAG МК и порт RS-232 выведены для экономии места на единый разъём типа IDC-10.

Поддерживаются два режима загрузки МК при включении питания (выбор осуществляется съёмной перемычкой «Netboot disable»):

1. Локальный: образы ядра операционной системы (ОС) и корневой файловой системы (ФС) загружаются из NOR Flash объемом 8 МБ.
2. Сетевой: образы ядра ОС и корневой ФС загружаются по сети Ethernet с сервера (по протоколу tftp), при этом с сервера также загружаются дополнительный скрипт инициализации модуля и прошивка ПЛИС для её конфигурирования с использованием интерфейса JTAG. С этой целью сигналы JTAG ПЛИС подключены к выводам GPIO МК. Загрузка прошивки в ПЛИС по JTAG в формате Jam STAPL Byte-Code (jbc) осуществляется доработанной программой от Altera [20]. Время загрузки около 1 секунды. После загрузки прошивки исполняется shell-скрипт инициализации, который настраивает число выборок на канал, задержки триггера и сигналов и т.п. при помощи программы `adc-32atc-ctl` (см. раздел 6.5), а также запускает монитор пьедесталов, описанный в разделе 6.8. Сейчас скрипт инициализации является общим для всех модулей в системе, различия в инициализации отдельных модулей учитываются при помощи конструкции `if` по IP-адресу модуля.

Сетевой режим является штатным при работе модуля в составе ССД установки. Он позволяет иметь актуальные версии ПО и проекта ПЛИС для всех экземпляров модулей (которых может быть несколько сотен), для этого достаточно просто обновить файлы на сервере. Локальный режим предназначен для тестовых (стендовых) применений при работе с единственным экземпляром модуля, когда развёртывание инфраструктуры для сетевой загрузки нецелесообразно.

Синхронизация системного времени на модуле ADC-32ATC осуществляется по протоколу NTP, для этого на МК при загрузке запускается стандартный демон `ntpd`. Сейчас для синхронизации времени используется тот же сервер, что и для сетевой загрузки, адрес которого задаётся в параметрах загрузчика `u-boot`.

Как непосредственно на МК, так и для разработки программного обеспечения (ПО) для него применяется свободное ПО, в частности:

1. `urjtag` 0.10 [21] для первичного программирования загрузочной NOR Flash с использованием JTAG в стандартном режиме периферийного сканирования для установки/опроса выводов МК;
2. `u-boot` 2014.07 [22] для загрузки ОС;
3. `Linux` 2.6.36 [23] в качестве ядра ОС;
4. `glibc` 2.9 [24] в качестве стандартной библиотеки C;

5. busybox 1.22.1 [25] в качестве пользовательского окружения;
6. сервер openssh 6.6p1 [26] для удалённого доступа на модуль по ssh;
7. buildroot 2014.08 [27] для генерации образа корневой файловой системы;
8. gcc 4.3.2 [28] + binutils 2.19.1 [29] в качестве кросс-компилятора, для его сборки применяется crosstool-ng 1.9.3 [30];
9. GNU/Linux в качестве ОС для разработки ПО.

Достигнуты следующие скоростные характеристики:

1. Скорость копирования данных из памяти МК в память МК (DDR2 SDRAM): 150 МБ/с (glibc memcpy()).
2. Скорость выдачи данных по Gigabit Ethernet с использованием протокола TCP (стандартный стек TCP/IP из ядра Linux, данные передаются из оперативной памяти МК, применяются кадры обычного размера (не более 1500 байт)): 36 МБ/с.
3. Скорость выдачи данных по Gigabit Ethernet в виде UDP-дейтаграмм с использованием низкоуровневого интерфейса PACKET MMIO ядра Linux (данные передаются из оперативной памяти МК и забираются сетевым контроллером для передачи прямо из пользовательского буфера, применяются большие кадры (jumbo frames) с полезной нагрузкой 8192 байт): 116 МБ/с.
4. Скорость приёма данных из буферной памяти ПЛИС (памяти СБРОСа) по каналу ПЛИС – МК в оперативную память МК с использованием механизма DMA: 60 МБ/с.
5. Скорость передачи данных из буферной памяти ПЛИС в Gigabit Ethernet по TCP, с кадрами обычного размера (не более 1500 байт) и с промежуточной буферизацией в оперативной памяти МК: 24 МБ/с. Это означает выдачу всей буферной памяти ПЛИС за 2.7 с.
6. Скорость передачи данных из буферной памяти ПЛИС в Gigabit Ethernet с передачей основного массива данных в виде UDP-дейтаграмм с использованием PACKET MMIO, больших кадров (~ 8 кБ) и с промежуточной буферизацией в оперативной памяти МК: не менее 40 МБ/с (оценка, полученная из п. 3,4).

6.2. Исследование пропускной способности МК

Было проведено исследование причин достаточно низкой скорости чтения из канала данных ПЛИС в память МК при помощи DMA (60 МБ/с). Пиковая скорость eLB составляет 92 МБ/с, здесь учтено, что пакетный цикл чтения для 16 16-битных слов занимает 20 тактов и UPM вставляет как минимум 2 такта между соседними

циклами, поэтому скорость получается $126 \cdot 16 / 22$ МБ/с. Для этого была измерена скорость чтения из канала данных ПЛИС в регистры процессора с последующим выкидыванием данных. При этом был включен кэш процессора для генерирования пакетных циклов чтения и использовалась процедура на ассемблере, которая производила чтение блоками по 2 линии кэша (64 байта) в 8 64-разрядных регистров FPU с предвыборкой следующих 2 линий кэша при помощи инструкции процессора `dcbt`. Была получена скорость 74 МБ/с. Разница между 92 МБ/с и 74 МБ/с эквивалентна дополнительным неиспользуемым 5 тактам шины eLB, которые могут объясняться затратами на арбитраж внутренней шины микроконтроллера CSB [18, 19] и особенностями работы контроллера eLB и интерфейса процессора с CSB. Если считанные данные не выбрасывать, а копировать в память МК, то скорость падает до 52 МБ/с. Это означает, что шина CSB блокируется на всё время выполнения цикла чтения на внешнем интерфейсе eLB, а не только на момент передачи считанных данных. Проблема, по-видимому в том, что контроллер eLB никогда не является мастером на CSB. Поэтому, как только процессор или контроллер DMA иницируют цикл чтения с eLB, то сразу захватывают шину данных CSB и ожидают данные от eLB, поскольку им неизвестно реальное время отклика контроллера eLB. Это сильно снижает пропускную способность CSB, в результате чтение по DMA eLB замедляет также скорость работы сетевого стека Linux (что видно по скорости выдачи данных буферной памяти ПЛИС 24 МБ/с против 36 МБ/с при отсутствии DMA на eLB).

Как видно из результатов, приведённых выше, скорость копирования из памяти в память 150 МБ/с. Это означает, что CSB обеспечивает пропускную способность при работе с памятью 300 МБ/с, поскольку данные при копировании передаются 2 раза по внутренней шине МК. Поэтому следует ожидать, что при применении для связи с ПЛИС присутствующего в MPC8308 интерфейса PCI Express скорее всего можно реально достигнуть скорости ~ 100 МБ/с передачи данных из буферной памяти ПЛИС в Ethernet. Это связано с тем, что контроллер PCI Express сам может осуществлять прямой доступ в память без отдельного внешнего контроллера DMA, который 2 раза гоняет данные по внутренней шине с ожиданием (и соответствующей блокировкой внутренней шины) завершения запросов на чтение от источника данных. Пропускная способность внешнего интерфейса PCI Express также более чем достаточна — 240 МБ/с. К сожалению, применяемая нами ПЛИС (Cyclone III) интерфейсом PCI Express не обладает, поэтому проверить это мы не смогли.

6.3. Интерфейсы МК – формирователь

На соединитель формирователя выведены 10 сигналов GPIO МК в уровнях LVCMOS3.3. Они могут свободно использоваться для управления формирователем. В актуальной версии программного обеспечения МК реализован протокол шины SPI и три однопроводных интерфейса 1-wire для чтения данных с температурных датчиков.

6.4. Мониторинг температурного режима модуля

Для измерения температуры печатной платы в модуле ADC-32ATC присутствует температурный датчик DS18B20, работающий по протоколу 1-wire. Датчик расположен недалеко от центра печатной платы и подсоединён к МК с помощью одного сигнала GPIO.

Мониторинг температуры весьма полезен при установке массива модулей в крейт, поскольку позволяет вовремя заметить проблемы с системой охлаждения крейта и избежать поломки модулей из-за перегрева.

Для считывания температуры с датчика был написан модуль ядра Linux, который также поддерживает чтение температурных датчиков DS18B20 по соответствующим трём линиям 1-wire интерфейса МК – формирователь. Модуль ядра потребовался по той причине, что в протоколе 1-wire присутствуют достаточно жёсткие временные ограничения на применяемые в протоколе импульсы, которые невозможно удовлетворить при работе в пространстве пользователя.

Достаточно большое количество дополнительных температурных датчиков может быть полезным при применении модуля ADC-32ATC для оцифровки сигналов с ячеистых лавинных фотодиодов (ЯЛФД, SiPM) для их температурного мониторинга.

6.5. Канал управления модулем ADC-32ATC

Управление модулем осуществляется по каналу управления интерфейса МК с ПЛИС (раздел 3.4 и Приложение В). Для взаимодействия с каналом управления ПЛИС разработана специальная программа `adc-32atc-ctl`, которая выполняется прямо на микроконтроллере. Она позволяет посылать команды как в «сыром» виде (последовательности 16-битных чисел), так и в высокоуровневом — в виде символьных строк, которые транслируются далее в последовательности 16-битных чисел. Также возможно выполнение операций чтения из канала управления. Программа предназначена для изменения режимов/параметров проекта в ПЛИС, а также для обращения (записи и чтения) по SPI ПЛИС (см. раздел 3.7) к регистрам микросхем АЦП и формирователя тактовых сигналов AD9516-0. Пример высокоуровневой команды записи регистра микросхемы АЦП через SPI:

```
adc-32atc-ctl -c 'adc0 spi-write 0x11 0x08'
```

Удалённый доступ к модулю ADC-32ATC для запуска программ на МК осуществляется по протоколу SSH.

Программа `adc-32atc-ctl` построена поверх библиотеки на языке C. Эта библиотека используется также в других инструментах, которым необходимо взаимодействие с ПЛИС (например, в описанном в разделе 6.8 мониторе пьедесталов). Значительная часть кода библиотеки автоматически генерируется из высокоуровневого декларативного описания команд канала управления ПЛИС, описанных в Приложении С. Для трансляции описания в C был разработан соответствующий транслятор на языке Scheme. Подобный подход позволяет существенно снизить трудозатраты, в том числе при дальнейшем развитии ПО, поскольку вместо трудоёмкого и потенциально

опасного с точки зрения внесения ошибок рефакторинга кода на С, все модификации осуществляются в достаточно компактном трансляторе без изменения высокоуровневой спецификации команд канала управления ПЛИС.

6.6. Драйвер считывания буферной памяти ПЛИС

Для обслуживания канала данных интерфейса МК с ПЛИС применяется специально разработанный драйвер ядра Linux. С драйвером ассоциированы два символических файла устройства: `/dev/adc-data-0` и `/dev/adc-data-1`, в первый посылаются данные от запроса, инициированного прерыванием `MC_IRQ1` (данные от внешних циклов и внутренних автоциклов), во второй — прерыванием `MC_IRQ2` (данные от командных СБРОСов, см. раздел 7). Если файл никем не открыт или процесс, открывший файл после вычитывания данных очередного СБРОСа, не просигнализировал драйверу вызовом `lseek()` о готовности читать новый СБРОС, то драйвер самостоятельно отрабатывает запрос ПЛИС, выбрасывая данные. Драйвер производит запись данных от ПЛИС прямо в передаваемый ему в системном вызове `read()` буфер пользовательского процесса при помощи прямого доступа к памяти (DMA).

6.7. Сервер выдачи данных по Ethernet

Передача данных в сеть сервером выдачи данных начинается сразу же после получения первой порции данных 512 кБ. Сервер представляет собой 2 работающих параллельно потока (`thread`), один читает память ПЛИС в буфер равного ей размера (64 МБ) в оперативной памяти МК, второй передаёт из этого буфера в сеть по протоколу ТСР. Поэтому время отработки запроса ПЛИС на чтение данных ограничено сверху некоторой величиной (около 2 секунд) вне зависимости от скорости работы сети и проблем в сети.

Для получения массива данных СБРОСа необходимо установить с модулем соединение ТСР на порт 5000. По концу СБРОСа модуль отправляет массив накопленных данных в установленное соединение и сразу же закрывает его. Это служит признаком конца передачи данных текущего СБРОСа, и можно вновь установить соединение ТСР для приема следующего. Текущий СБРОС помечается как переданный и повторно модулем не передается; сервер передачи на модуле будет ожидать поступления «свежих» данных. Сервер предназначен для обслуживания единственного потребителя данных.

6.8. Процесс мониторинга пьедесталов

Для мониторингования пьедесталов и использования их значений для аппаратного (в ПЛИС) отсеечения событий без сигналов (см. раздел 7.4) был написан демон `adc-32atc-peds`. Демон работает прямо на микроконтроллере. При запуске в аргументах командной строки можно задать тип отсеечения по порогу (по среднему по всем выборкам, по максимальной выборке в событии или вообще выключить отсеечение),

превышения порога над пьедесталом для каждого канала отдельно и порт, на котором будет работать TCP-сервер для выдачи параметров пьедесталов различным мониторирующим программам. Демон ожидает окончания сигнала GATE, захватывает блокировку канала управления, затем блокирует запуск циклических СБРОСов по внешнему сигналу GATE, настраивает и запускает внутренний командный СБРОС. Чтение данных осуществляется через `/dev/adc-data-1`. Для каждого канала рассчитывается среднее значение пьедестала по всем выборкам и по всем событиям и среднеквадратичное отклонение. Также рассчитываются средние значения выборок с определённым номером по всем событиям (для всех номеров). Для среднего выборки с определённым номером каждое значительное отклонение (на 5 и более стандартных отклонений от общего среднего по всем выборкам для данного канала) наращивает специальный счётчик. Общие средние округляются до целого числа по стандартным правилам, к ним добавляются превышения над пьедесталом, заданные аргументами командной строки при запуске. Полученные значения порогов для каждого канала записываются в ПЛИС. Далее разрешается запуск циклических СБРОСов по внешнему сигналу GATE и отпускается блокировка канала управления. После этого в каждое входящее TCP-соединение по указанному при запуске порту в текстовом виде выдаётся время в UNIX-формате, значение счётчика аномальных превышений средних значений в отдельных выборках за последние 100 СБРОСов, среднее значение пьедестала и его среднеквадратичное отклонение для каждого канала.

7. Прошивка (проект) ПЛИС для применения в ССД установки ВЕС

Большинство основных функций модуля, кратко перечисленных в разделе 2, имеют параметры, программируемые по каналам управления МК и USB. К дополнительным специализированным и тестовым функциям, также имеющим программируемые параметры, относятся:

- программное управление циклами (генерация внутренних сигналов GATE и START);
- генерация тестовых данных на уровне ввода оцифрованных данных с АЦП в конвейеры;
- встроенные средства для настройки и проверки стробирования сигналов с АЦП.

Большинство программируемых параметров имеют предустановленные («дефолтные») значения по включению питания или аппаратной очистке модуля.

7.1. Состояния модуля и связанные с ними команды управления

Есть три состояния модуля: состояние А генерации и приема данных, переходное состояние В готовности к сессии управления и состояние С сессии управления. Для

переключения между состояниями в каждом из них предусмотрены соответствующие команды (см. Приложение С, таб. 3).

Состояние А — основное рабочее состояние модуля, в котором осуществляется приём и обработка внешних и внутренних сигналов СБРОСа, накопление и передача данных в ССД и переход в состояние В.

В состоянии В выполняются ряд одиночных команд управления и переход в состояния А и С.

В состоянии С сессии управления осуществляется основное программное управление модулем. Сессия образуется передачей последовательности байтовых кодов (команд и данных), предназначенных для логических блоков проекта или устройств на шине SPI (до 16 блоков/устройств). Одна сессия управления может включать неограниченное количество команд для одного блока или устройства, но не допускает управления несколькими.

Сессия начинается кодом выборки устройства, далее следуют команды управления устройством. Команды делятся на три группы: адресные (Приложение С, таб. 4), состоящие из двух и более байтов (первый байт — код команды, остальные байты — операнды), кодовые (Приложение С, таб. 5) — однобайтовые (5 значащих битов, до 32-х кодовых команд для каждого блока или устройства) и данные, интерпретируемые SPI устройством (фиксированное количество 3 байта в одной сессии управления).

Для закрытия сессии зарезервирован код 00h, он исключен из допустимых кодов управления для логических блоков. Для SPI устройств во избежание конфликта с кодами, предусмотренными производителями, код модифицируется при формировании сессии, а при выдаче самому устройству восстанавливается. После закрытия сессии управления модуль автоматически переходит в состояние В готовности к открытию новой сессии управления.

7.2. Внешние и внутренние сигналы управления СБРОСом

Сигналы управления в СБРОСе (GATE и START) принимаются извне (обычно — от триггерной электроники ССД) или генерируются в модуле 2-мя способами: разово по команде управления (разовая программная последовательность) или с заданной скважностью (автоциклы). Разовая последовательность задается 2-мя параметрами: количеством триггеров (событий) (1 – 8191) и периодом их следования (2 – 64 мкс). Для автоциклов к ним добавляется параметр паузы между СБРОСами (между сигналами GATE) (1 – 10 с). Параметры разовых последовательностей и автоциклов программируются независимо. Выбор между внешними и внутренними циклами задается переключателем на лицевой стороне модуля и командами управления циклами (см. Приложение С, таб. 5). Разовые последовательности разрешены всегда, их корректная генерация на фоне внешних циклов или автоциклов возлагается на источник управления (МК или USB). Для повышения надежности управления циклами от МК (основной рабочий режим) введены команды блокировки/разблокировки запуска всех циклов (см. Приложение С, таб. 3).

Внутренняя генерация сигналов является полезным инструментом для отладочных работ и стендовых применений модуля.

7.3. Конвейеризация и выборка данных

Прием оцифрованных данных с 8-ми канальных АЦП выполняется с использованием только фреймовой (FCO) выходной частоты тактирования одного из АЦП [11], одной схемы PLL для формирования стробирующих сигналов и стандартной мегафункции восьмиканального LVDS приемника — десериализатора с фактором 6 [13]. Оцифрованные данные 4-х АЦП с приписанными адресами каналов с тактовой частотой выдачи данных с десериализаторов (однозначно синхронизированной с тактированием АЦП) последовательно загружаются через четыре 128 разрядных регистра в 4 основных конвейера данных, реализованные в виде регистров сдвига на блоках памяти ПЛИС. Глубина основных конвейеров для рабочего проекта составляет 64 такта. На выходе основных конвейеров добавлены блоки сдвига битовой последовательности с АЦП для работы в режимах x2 и x4 (см. раздел 4), после которых добавлены дополнительные конвейеры с программируемой индивидуально для каждого канала длиной от 0 до 15 тактов. Дополнительные конвейеры выполнены в виде сегментов регистров сдвига из D-триггеров ПЛИС длиной 1, 2, 4 и 8 тактов. Сегменты либо включаются в конвейер, либо исключаются из него в зависимости от соответствующих битов в коде задержки от 0 до 15. Блоки сдвига битовой последовательности и дополнительных конвейеров вносят задержку 6 тактов при нулевом значении дополнительной задержки. Таким образом общая длина конвейера может изменяться в диапазоне от 70 до 85 тактов индивидуально на каждом канале.

С выходов конвейеров данные параллельно отбираются на прием и обработку по синхронизированному с тактовой частотой конвейеров и задержанному сигналу START_SYN. Количество выборок по одному триггеру (от 1 до 16) является программируемым параметром проекта ПЛИС. Временное положение первой в последовательности выборки задается программированием задержки триггера в диапазоне 7 – 134 тактов. С учетом задержки в АЦП (Pipeline Latency, 8 тактов) это соответствует диапазону от отставания триггера относительно первой выборки на 72 такта до её опережения триггером на 56 тактов (при нулевых значениях индивидуальных поканальных дополнительных задержек). Задержка триггера реализована с помощью счётчика, поэтому она вносит вклад в мёртвое время модуля, равный значению задержки, который будет обозначаться T_1 . После поступления на вход модуля сигнала START, последующие возможные сигналы блокируются на время T_1 . Задержка триггера на 78 тактов соответствует одновременности (с точностью до такта) триггера и входных сигналов (для первой выборки) и величине T_1 в 1.95 мкс.

7.4. Обработка и форматирование данных

При формировании события последовательность выборок со всех программно незамаскированных каналов модуля подвергается цифровой обработке.

Данные с канала могут подвергаться пороговой фильтрации одним из двух способов: по превышению заданного порога в любой из выборок последовательности или по превышению среднего значения по всем выборкам. Пороги могут устанавливаться поканально или одним значением для всех 32-х каналов, во всем диапазоне АЦП (1 – 4096 отсчетов). В событие включаются только каналы «с превышением порога».

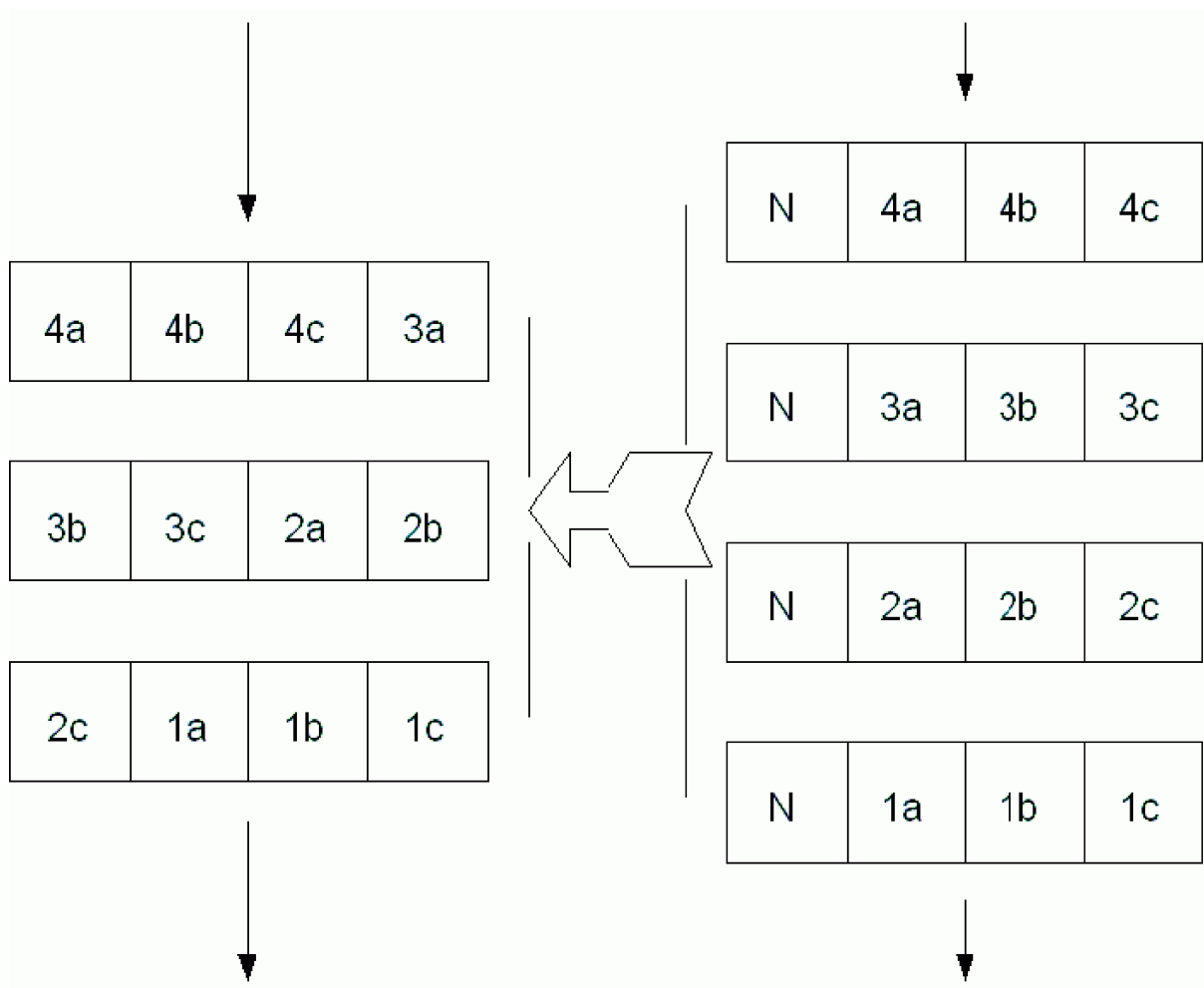


Рис. 13. Форматирование с упаковкой данных.

Далее событие как целое может фильтроваться по запрограммированному максимальному количеству отобранных в событие каналов (1 – 32). При его превышении событие трактуется как «шумовое» и включается в СБРОС в составе одного заголовка (см. ниже) с нулевым количеством каналов и нулевой длиной события, но с сохранением маски сработавших каналов, что отличает такое событие от исходно пустого события.

Отобранные в событие данные форматируются в одной из двух программируемых мод: адресно-масочной без упаковки и масочной с упаковкой данных.

В адресно-масочной моде без упаковки каждое 12-битное значение выборки, со-

проводимое 3-х битным кодом адреса канала, образует слово данных. Максимально возможная длина события без учета заголовка события составляет 512 (32 канала по 16 выборок) слов. Наличие или отсутствие маски сработавших каналов программируется.

В масочной моде с упаковкой в начале массива данных события всегда идет маска сработавших каналов из 2-х слов. Далее в порядке следования каналов (согласно маске) идут упакованные в 16-и-битные слова значения выборок без кодов адресов (см. рис. 13).

Отформатированное событие, в т.ч. пустое, дополняется аппаратно формируемым заголовком, составляющим восемь (или шесть) 16-разрядных слов. Он содержит: временной штамп события в СБРОСе (от начала сигнала GATE до триггера) в диапазоне до 12 с с дискретой времени 6.25 нс; коды грубого и точного измерения фазы; число сработавших каналов в событии; маску отобранных каналов (опционно); объем события (в словах). Временной штамп события является его идентификатором и используется при программной сборке полного события ССД с участием всех ее модулей (подсистем).

События с заголовками накапливаются в буферном FIFO объемом 8192x16 бит (рис. 14), выполняющим функции промежуточного хранения потока данных и его согласования по частоте и формату для последующей записи в SDRAM память СБРОСа (64 МБ).

Накопленные в памяти СБРОСа события при выдаче в МК или USB дополняются заголовком СБРОСа из 8-ми слов, содержащим: длительность СБРОСа (до 12 с в дискретах по 6.25 нс), количество событий в нём и его объём (в 16-битных словах); количество выборок на канал; коды мод фильтрации и форматирования; набор флагов.

Формат данных СБРОСа приведён для справки в Приложении D.

7.5. Тракт формирования и передачи данных в модуле

На рис. 14 представлена блок-схема тракта передачи данных в модуле ADC-32АТС.

В разных частях тракта форматы обрабатываемых данных и тактовые частоты обработки различны. На стадии выборки данных из потока и формирования динамической маски отбора каналов в событие (совместно по запрограммированному (предустановленному) маскированию каналов и их отсечению по порогам) данные в 16-и-байтном формате обрабатываются с частотой тактирования АЦП параллельно по 4-м конвейерам данных и буферизуются в FIFO 4x128x16. На стадии фактического отбора каналов по динамической маске, форматирования и буферизации данных в памяти события (попарно переключаемые FIFO 256x16) и при буферизации потока событий в FIFO 8192x16 СБРОСа — в формате 2-х байтовых слов с частотой 160 МГц. Взаимодействие (чтение/запись) с контроллером памяти СБРОСа (DDR SDRAM 32x16 Мбит) организуется фреймами по 8 байт с частотой 48 МГц, а сами запись и чтение данных в память СБРОСа — в 2-х байтовом формате с эффективной

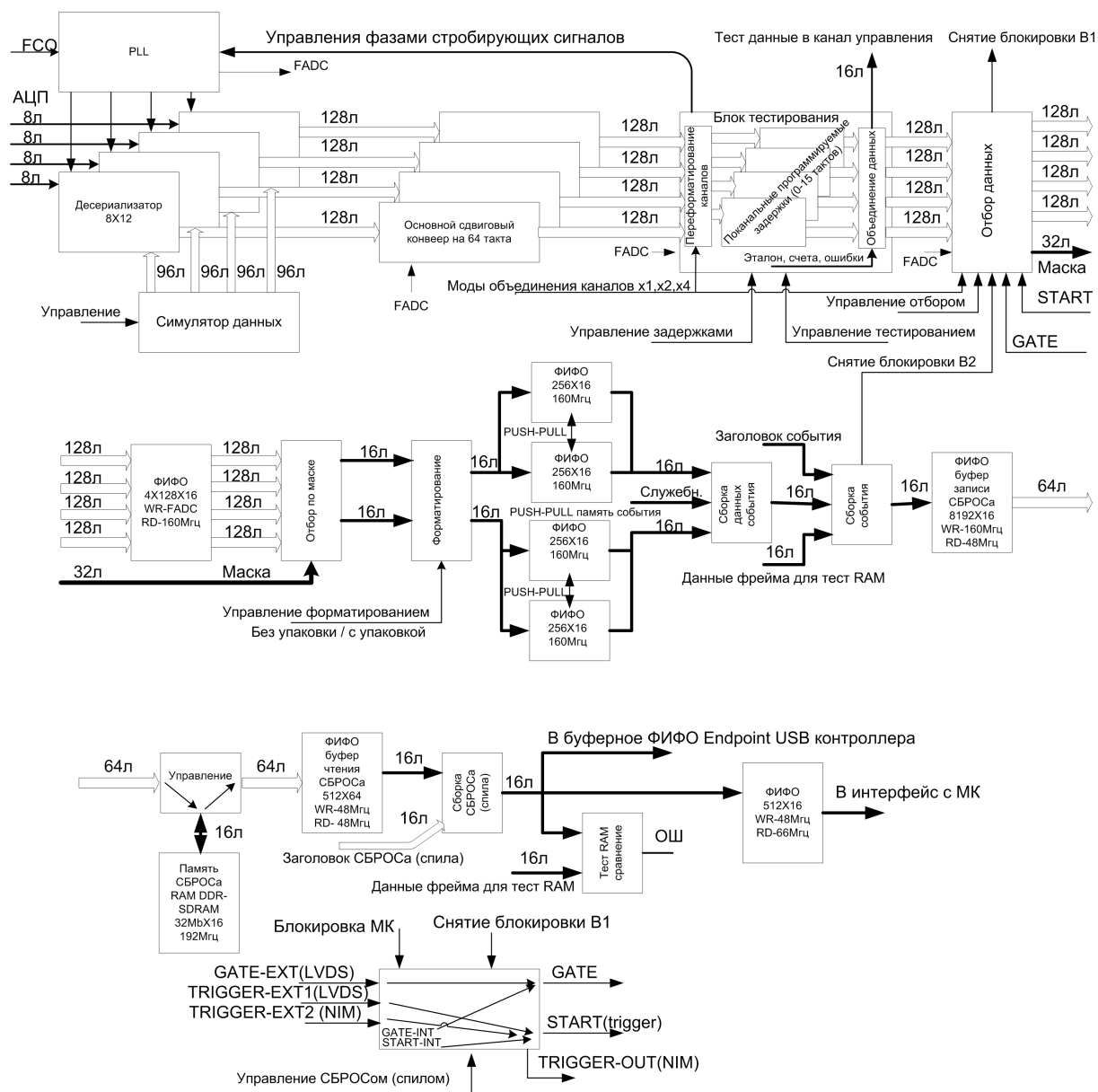


Рис. 14. Блок-схема тракта передачи данных модуля ADC-32ATC.

частотой 192 МГц (96 МГц в режиме DDR). На стадии выдачи данных СБРОСа в интерфейс с USB контроллером реализуется 2-х байтовый формат данных с максимальной для контроллера тактовой частотой 48 МГц (FIFO 512x64), а в интерфейс с МК — 2-х байтовый с тактовой частотой локальной шины МК 66 МГц (FIFO 512x16).

На самом верхнем уровне процесс обработки события представляет собой трёх-стадийный конвейер со следующими стадиями:

1. Задержка триггера.
2. Отбор порции выборок АЦП для записи в событие и запись события в попарно переключаемые FIFO 256x16.
3. Запись события из FIFO 256x16 в буферное FIFO СБРОСа.

Длительность каждой стадии i для конкретного события мы будем обозначать T_i . Величина T_1 есть просто длительность задержки триггера, остальные величины можно вычислить по формулам:

$$T_2 = (M + 3.5) \cdot t_{adc} + (K \cdot M \cdot \max(N_0, N - N_0) + 7) \cdot t_{sys}$$

$$T_3 = (K \cdot M \cdot N + 8) \cdot t_{sys}$$

где N — число отображенных для записи в событие каналов ($0 - 32$), N_0 — число отображенных для записи в событие каналов с номерами ≤ 15 , t_{adc} — период тактовой частоты АЦП, K — коэффициент упаковки (1.0 для неупакованного формата события, 0.75 для упакованного), M — число выборок на канал ($1 - 16$), t_{sys} — период тактовой частоты обработки и буферизации данных (6.25 нс для текущего проекта).

Мы будем обозначать как T_2^{max} и T_3^{max} максимально возможные для заданной конфигурации модуля значения величин T_2 и T_3 (которые достигаются при максимально возможных значениях N и N_0).

Для работы конвейера без потери событий достаточно (но не необходимо!) потребовать следующего ограничения на интервал ΔT между триггерами:

$$T_{dead} = \max(T_1, T_2^{max}, T_3^{max}) \tag{1}$$

$$\Delta T \geq T_{dead}$$

Мы будем называть T_{dead} мёртвым временем модуля. Ограничение (1) должна гарантировать триггерная система установки.

Для корректной работы конвейера с возможной потерей событий при нарушении условия (1) предусмотрены блокировки В1 и В2. При поступлении триггера на вход модуля, взводится блокировка В1, которая блокирует поступление следующего триггера прямо на входе модуля. Сигнал с выхода задержки триггера (которая располагается в блоке отбора данных) сбрасывает блокировку В1 и взводит блокировку В2, которая блокирует следующий сигнал с выхода задержки триггера. Блокировка

В2 сбрасывается блоком сборки события сразу после переключения вдвоенных буферов FIFO 256x16 в момент начала записи события в буферное FIFO СБРОСа (то есть когда событие попадает в последнюю третью стадию описанного выше конвейера).

При работе в режиме 16 выборок на канал, 32 канала, получаются следующие числа: $T_2^{max} = 2.13$ мкс, $T_3^{max} = 3.25$ мкс при использовании неупакованного формата события и $T_2^{max} = 1.73$ мкс, $T_3^{max} = 2.45$ мкс при использовании упакованного формата. Поскольку в большинстве реальных применений входные аналоговые сигналы поступают на вход модуля раньше триггера, то максимальное значение для T_1 составляет 1.95 мкс (см. раздел 7.3 выше). При работе в режиме 8 выборок на канал (или 16 выборок на канал с 16 активными каналами), мёртвое время начинает определяться величиной T_1 . В будущем мы собираемся реализовать конвейерную задержку триггера (вместо текущей задержки на счётчике), которая снизит величину T_1 до 200 – 400 нс. Также необходимо отметить, что существует фундаментальное ограничение на мёртвое время, которое нельзя преодолеть каким-либо усовершенствованием прошивки ПЛИС. Максимальная частота работы применённой DDR SDRAM составляет 166 МГц (пропускная способность $6.64 \cdot 10^8$ байт/с), максимальный размер события — 784 байта для упакованного формата, следовательно мёртвое время не может быть меньше $784/664 = 1.2$ мкс в режиме 16 выборок на канал, 32 канала.

8. Отладочные функции в модуле ADC-32АТС

8.1. Работа с тестовыми данными

Для отладочных и контрольных целей в модуле предусмотрены режимы симуляции данных с АЦП на уровне их ввода в конвейеры данных. Переход на работу с тестовыми данными задается переключателем на лицевой стороне модуля. С помощью съёмной перемычки на плате модуля выбирается тип тестовых данных — фиксированные, привязанные к каналам, или псевдослучайные. Работа с симулированными данными позволяет изучить работу модуля во всех возможных режимах отбора и обработки данных. Кроме этого, в проекте заложен режим аппаратного тестирования части тракта данных и всего объема памяти СБРОСа (тест памяти) с генерацией «фрейма» (блока) тестовых инкрементных кодов, их записью в буферное FIFO СБРОСа и чтением из памяти СБРОСа с аппаратной проверкой на совпадение. Режим теста памяти также выбирается переключателем на лицевой стороне модуля.

8.2. Управление стробированием данных с АЦП

Для настройки и проверки стробирования данных с АЦП (битового и фреймового стробирования на входах и выходах десериализаторов, соответственно) в модуле реализован специальный блок с программированием функций:

- включение режима сравнения данных с заданного или всех каналов АЦП с эталоном;

- включение сбора данных на заданное время с регистрацией количества выборок и количества ошибок (расхождений с эталоном);
- дискретный сдвиг по фазе заданных сигналов стробирования.

Результаты сбора данных могут приниматься в USB или МК в виде тестовых СБРОСов по каналу данных или чтением тестовых данных по каналу управления в МК (при управлении от МК). Для проверки (или настройки) программируются выдача с АЦП одного из возможных фиксированных кодов [11] и соответствующее значение эталона для блока тестирования. Затем выполняется сбор данных в виде сессии управления, содержащей включение сбора, нужное количество «холостых» кодов, задающее время сбора, и его выключение. Для визуального анализа результатов с управлением по USB (без автоматизации процесса) имеется тестовая программа под операционной системой GNU/Linux. Для управления сдвигом фаз стробирующих сигналов одна или две PLL, реализующие формирование этих сигналов (зависит от проекта), используются в режиме разрешения динамического реконфигурирования [7]. Изменение фазы определенной тактовой частоты выполняется адресацией регистра соответствующего выхода PLL и инкрементом/декрементом фазы на фиксированную дискрету времени в $1/8$ периода первичной внутренней частоты PLL (0.26 нс для рабочего проекта). За одну сессию управления можно изменить фазы любого количества сигналов и на любое нужное количество шагов.

С помощью описанной выше процедуры можно при отсутствии заметного разброса по фазам тактовых сигналов на АЦП осуществить оптимизацию стробирования оцифрованных данных с АЦП и внести ее в проектные настройки PLL. В других случаях и при использовании АЦП с максимальной частотой тактирования (65 МГц) может потребоваться формирование тактовых частот для АЦП только с помощью формирователя AD9516-0 (см. раздел 3) и программная настройка индивидуальных задержек для этих частот.

9. Радиационная стойкость модуля ADC-32ATC

Модуль ADC-32ATC никаким испытаниям на радиационную стойкость не подвергался. Однако похожий по конструкции модуль TDC-32MWC⁴ был испытан в горячей зоне эксперимента СПИН на 8 канале У-70 [9]. Общим для ADC-32ATC и TDC-32MWC является применение динамической памяти большого объема и микропроцессора, на котором выполняется операционная система (ОС) Linux. При этом достаточно большой объем исполняемого процессором кода хранится в динамической памяти, и радиационное повреждение (обратимое, так называемое SEU — single event upset) отдельных битов в ней может приводить к зависанию или некорректной работе исполняемой программы.

Испытания в эксперименте СПИН проходили следующим образом:

⁴Цифровая часть этого модуля полностью идентична TDC-48DT, описание которого можно найти в [10].

- Модуль TDC-32MWC располагался на расстоянии от 0.4 до 7.3 м от оси протонного пучка с интенсивностью $\sim 10^{12}$ протонов за цикл ускорителя на расстоянии 7 м вдоль пучка от установленной в пучок мишени толщиной 0.1% длины взаимодействия (было проведено несколько экспозиций).
- Испытываемый модуль подключался по Ethernet к компьютеру (в экспериментальном домике), который периодически испускал ICMP-запросы программой ping. Измерялось время от загрузки модуля до момента прекращения модулем ответов на ICMP-запросы (так называемое время жизни ОС).
- Если модуль после длительного перерыва снова начинал отвечать на ICMP-запросы, то во всех этих случаях было проверено, что это происходило после самопроизвольной перезагрузки микропроцессора.

Результаты испытаний следующие: среднее время жизни ОС варьировалось от 0.5 до 6.5 часов в зависимости от удалённости от пучка; после прекращения работы канала 8 было проверено, что в течение более чем 2-х суток сбой в модуле не наблюдался, при этом модуль продолжал находиться в одной из точек измерения в канале 8.

Был также проведён ещё один эксперимент с применением мощного β -источника (к сожалению, неизвестной интенсивности): источник сначала располагался в течение 1 суток прямо на микросхеме процессора Atmel AT91RM9200 — никаких сбоев (в вышеописанном смысле) зарегистрировано не было, потом источник был размещён прямо на микросхеме SDRAM Samsung K4S561632J-UC75, сбой произошёл через 35 минут, причём после этого модуль не восстановился. Вероятно, это говорит о том, что наиболее слабым местом в смысле радиационной стойкости является динамическая память. Однако в эксперименте с β -источником, скорее всего, произошло превышение по допустимой дозе облучения SDRAM (что вызывает необратимые повреждения в устройстве), а стойкость к SEU существенно зависит от удельных ионизационных потерь, и β -источник в этом отношении является плохим для соответствующих испытаний.

В то же время в эксперименте BEC уже на протяжении нескольких ускорительных сеансов успешно эксплуатируется система из ~ 80 модулей TDC-32MWC и TDC-48DT, которые находятся на расстояниях от 0.5 до 2 м от оси пионного пучка с интенсивностью $\sim 2 \cdot 10^6$ частиц за цикл ускорителя. В пучке установлена бериллиевая мишень $\sim 10\%$ длины взаимодействия. За это время если и были какие-то сбои из-за SEU, то они носили единичный характер, не были идентифицированы и не представляли никаких трудностей при эксплуатации системы. В одном из сеансов один модуль ADC-32ATC располагался на расстоянии ~ 1 м от оси пучка в области мишени, и никаких явных сбоев замечено не было.

Исходя из написанного выше, следует ожидать, что модуль ADC-32ATC должен быть пригоден к массовой эксплуатации прямо на детекторах экспериментальной установки в экспериментах с пучком частиц с зарядом 1 (в единицах элементарного заряда) и интенсивностью $< 10^7$ частиц за цикл ускорителя и заведомо непригоден

при интенсивностях пучка $\sim 10^{12}$ частиц за цикл ускорителя. Необходимо также отметить, что в отличие от модулей TDC-32MWC и TDC-48DT, в модуле ADC-32ATC применяется микроконтроллер MPC8308 с контроллером памяти DDR2 SDRAM с поддержкой коррекции единичных битовых ошибок (ECC). Режим работы с ECC пока не тестировался, но аппаратно в модуле ADC-32ATC реализован и в будущем может использоваться для повышения стойкости модуля к SEU.

Заключение

Разработан автономный полнофункциональный 32-канальный модуль амплитудно-цифрового преобразования ADC-32ATC в «надетекторном» исполнении, предназначенный для оцифровки быстрых импульсных аналоговых сигналов. Модуль является АЦП выборочного типа (sampling ADC), имеет развитую архитектуру, аппаратную реализацию функций на ПЛИС и микроконтроллерное управление. В нём используются микросхемы конвейерных 8-канальных 12-разрядных АЦП AD9222 с быстродействующей схемой выборки-хранения на входе, с высокоскоростным последовательным выходом (от 480 до 780 Мбит/с) и тактовой частотой оцифровки от 40 до 65 МГц. Модуль обладает внутренней буферной памятью 64 МБ, интерфейсом Gigabit Ethernet 1000BASE-T, имеет мёртвое время не более 2.0 – 2.5 мкс при полной загрузке (с перспективой дальнейшего снижения до 1.2 мкс путём усовершенствования прошивки ПЛИС), число выборок на канал — до 16, число событий за цикл ускорителя (с типичной циклограммой ускорителя У-70) — не менее 85000 (ограничивается объёмом буферной памяти и зависит от среднего размера события). Модуль также поддерживает режимы работы x2 и x4 с чередованием АЦП по времени, что позволяет достичь двукратного (от 80 МГц и выше) и четырёхкратного (от 160 МГц и выше) увеличения эффективной частоты оцифровки за счёт пропорционального снижения числа каналов в модуле. Также в модуле реализован высокоточный ВЦП с разрешением не хуже 100 пс в смысле среднеквадратичного отклонения для точной привязки по времени внешнего асинхронного триггера к внутренней тактовой частоте АЦП.

Модуль в комплекте с формирователем входных аналоговых сигналов ориентирован на различные детекторы частиц в составе установок, работающих на ускорителях с циклическим выводом пучка в условиях умеренного радиационного фона. Несколько модулей уже эксплуатируются в составе эксперимента ВЕС на ускорительном комплексе У-70 (ИФВЭ, Протвино).

Благодарности

Авторы выражают благодарность А.А. Семаку за данные по радиационным испытаниям на 8-м канале, составившие основу раздела 9, Н.Е. Филимонову за монтаж модулей, В.Г. Готману и Д.Р. Еремееву за участие в тестировании модулей и М.О. Михасенко за участие в разработке УФ.

Приложение А. Конструкция и компоненты модуля ADC-32АТС

Модуль ADC-32АТС выполнен в виде 6-слойной платы размером 233.3x162 мм, опционально снабжаемым передней панелью. Он требует питание от +8 до +12 В, потребляемая мощность (при отсоединённой плате формирователя аналоговых сигналов) — до 10 Вт. В составе модуля 6 импульсных источников стабилизированного питания с номиналами +5, +3.3, +1.8, +1.2, +1.0 и –5 В и два линейных стабилизатора на +2.5 и +1.8 В. Линейный стабилизатор +1.8 В установлен исключительно для отладочных целей (чистое питание для АЦП) и имеет отдельный клеммник для подачи входного питания. Источники питания ±5 В предназначены для питания подключаемого к модулю формирователя аналоговых сигналов, каждый обеспечивает выходной ток до 3 А.

На передней (лицевой) стороне платы (с доступом через опциональную лицевую панель) располагаются 3 переключателя выбора режимов, ключ аппаратной очистки модуля и набор соединителей: разъём питания, разъём типа IDC-10 для загрузки прошивки ПЛИС в соединённый с ней EEPROM, комбинированный разъём JTAG/RS-232 для МК (тоже типа IDC-10), 8P8C для трёх сигналов LVDS, два РЦ-00 для сигналов NIM, IDC-14 для специализированного гальванически развязанного интерфейса SPI, USB2.0 (тип B) и Ethernet 1000BASE-T (8P8C).

Для приёма внешних сигналов управления (GATE, START, FCLK) через упомянутый выше разъём 8P8C, применяется LVDS-приёмник SN65LVDS348 с расширенным диапазоном синфазного сигнала от –4 до +5 В, что позволяет принимать сигналы также в уровнях ECL/PECL, а не только LVDS.

На плате модуля имеются:

- 6 съёмных перемычек для выбора дополнительных режимов работы ПЛИС; съёмная перемычка «Netboot disable» запрещение/разрешение загрузки прошивки в ПЛИС из МК (см. раздел 6); по включению питания в ПЛИС загружается дефолтная прошивка из EEPROM;
- штыревой соединитель (10 контактов) для вывода контрольных сигналов с ПЛИС;
- 4 LED-индикатора с ПЛИС.

На задней стороне платы размещен соединитель DIN41612-150M. Он выполняет функции съёма выходных аналоговых сигналов с формирователя, подачи на него питания (+5, +3.3, +1.8 и –5 В), соединения для шины SPI и 3-х линий однопроводного интерфейса 1-wire для температурных датчиков.

В модуле используются следующие компоненты:

- АЦП AD9222ABCPZ-40 (/50/65) с частотой оцифровки до 40 МГц (/50/65 МГц соответственно);
- ПЛИС Cyclone III EP3C40F484C6N (Altera/Intel);

- память СБРОСа DDR SDRAM MT46V32M16BN-6 IT:F емкостью 64 МБайт;
- микроконтроллер MPC8308VMAGD (Freescale/NXP);
- загрузочная память NOR Flash M29W640GT70NA (64 Мбит) для микроконтроллера;
- память микроконтроллера DDR2 SDRAM MT47H6M16HR-25E:H ёмкостью 128 МБайт/корпус;
- Ethernet-трансивер KSZ9021RN;
- USB-контроллер CY7C68001-56PVXC;
- разветвитель частот ADCLK846;
- многоканальный программируемый формирователь частот с ФАПЧ AD9516-0;
- импульсный преобразователь TPS54528 (для импульсных ИП с положительной полярностью);
- импульсный преобразователь MAX1846 (для импульсного ИП –5 В);
- LVDS-приёмник SN65LVDS348 с расширенным диапазоном синфазного сигнала.

Приложение В. Интерфейсы ПЛИС с МК и контроллером USB

Интерфейс ПЛИС с МК

ПЛИС подключается к МК с использованием 16-битной синхронной шины eLB микроконтроллера [17], нескольких сигналов GPIO и двух сигналов прерывания. Активным для всех управляющих сигналов является низкий уровень. Полный список сигналов представлен в таблице 1.

Шина eLB работает на частоте 66 МГц и обеспечивает пиковую пропускную способность 126 МБ/с. Её полезной особенностью является поддержка полностью программируемых циклов чтения/записи при помощи нескольких UPM (user programmable machine). Выбор между каналами данных и управления осуществляется при помощи сигналов MC_CS_D и MC_CS_C, каждому каналу соответствует своя UPM. В текущей реализации логической схемы в ПЛИС адресные линии шины eLB не используются. Временные диаграммы циклов чтения/записи показаны на рис. 15 и 16. Входные для МК сигналы должны устанавливаться минимум за 7 нс до момента их захвата (нарастающий фронт MC_LCLK для всех сигналов, кроме MC_D[15:0] в пакетном цикле чтения по каналу данных — в этом случае захват производится по спадающему фронту MC_LCLK) и удерживаться минимум 1 нс после момента

Таблица 1. Сигналы интерфейса ПЛИС – МК

Сигнал	Тип МК	Сигнал МК	Назначение
MC_LCLK	Выход	LCLK0	Тактовый сигнал eLB
MC_CS_C	Выход	$\overline{LCS2}$	Сигнал выбора канала управления на eLB
MC_CS_D	Выход	$\overline{LCS1}$	Сигнал выбора канала данных на eLB
MC_RD	Выход	LGPL0	Сигнал чтения eLB
MC_WR	Выход	LGPL1	Сигнал записи eLB
MC_WAIT	Вход	LUPWAIT	Сигнал ожидания eLB
MC_A[10:0]	Выход	LA[13:24]	Младшие 11 бит шины адреса eLB
MC_D[15:0]	Двунапр.	LD[0:15]	Шина данных eLB
MC_IRQ1	Вход	$\overline{IRQ1}$	Прерывание запроса на чтение данных для циклических запусков
MC_IRQ2	Вход	$\overline{IRQ2}$	Прерывание запроса на чтение данных для командных запусков
MC_GPIO1	Вход	GPIO6	Индикация выбора интерфейса (0 – МК, 1 – USB)
MC_RQ_D	Вход	GPIO9	Запрос на чтение буферной памяти ПЛИС по каналу данных
MC_WAIT_C	Вход	GPIO8	Сигнал ожидания канала управления (активный уровень означает запрет со стороны ПЛИС на посылки по каналу)
MC_GATE	Вход	GPIO10	Копия внешнего сигнала GATE
SOFT_RESET	Выход	GPIO5	Сброс контроллера канала управления в ПЛИС, либо сброс всей схемы в ПЛИС (если сопровождается MC_CS_C)
FPGA_TCK	Выход	GPIO14	Сигнал JTAG TCK ПЛИС
FPGA_TMS	Выход	GPIO13	Сигнал JTAG TMS ПЛИС
FPGA_TDI	Выход	GPIO2	Сигнал JTAG TDI ПЛИС
FPGA_TDO	Вход	GPIO4	Сигнал JTAG TDO ПЛИС

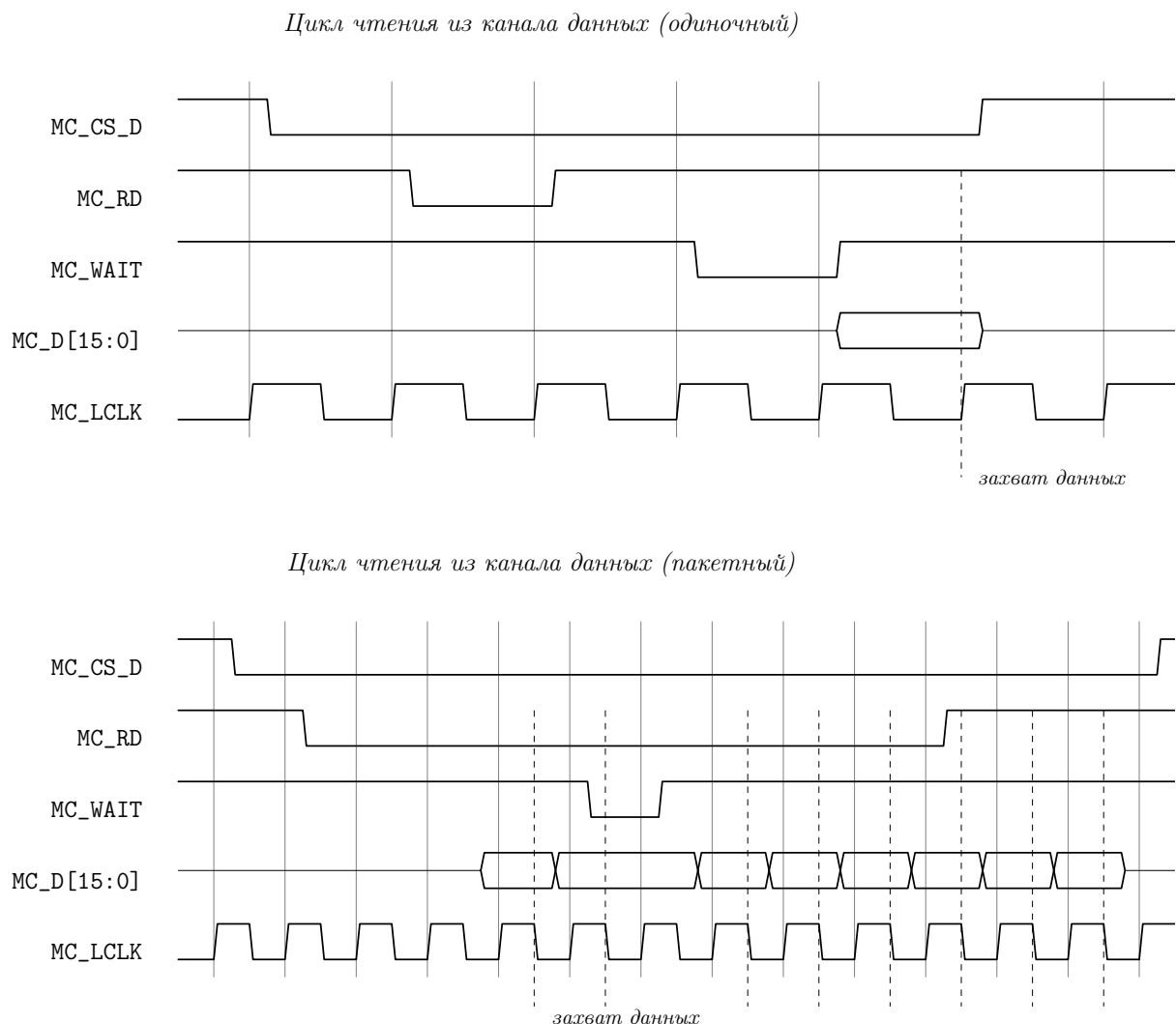


Рис. 15. Временные диаграммы для циклов чтения по каналу данных ПЛИС – МК. Интерфейс полностью синхронный с синхронизацией по нарастающему фронту MC_LCLK (за исключением захвата данных, который в пакетном режиме происходит по спадающему фронту MC_LCLK). Выдача данных начинается спустя фиксированное число тактов после установки RD. В пакетном режиме длительность RD в тактах в точности равна числу слов данных (при отсутствии сигнала MC_WAIT). Если ПЛИС не в состоянии выдать слово данных в требуемом такте, то она имеет право отложить выдачу до следующего такта, выдав вместо данных MC_WAIT. В случае захвата данных по спадающему фронту MC_LCLK сигнал MC_WAIT выдаётся по предшествующему положительному фронту MC_LCLK. При наличии сигнала MC_WAIT текущее состояние управляющих сигналов МК замораживается. Сигнал MC_WAIT может удерживаться несколько тактов подряд, пока ПЛИС не придёт в состояние готовности к выдаче слова.

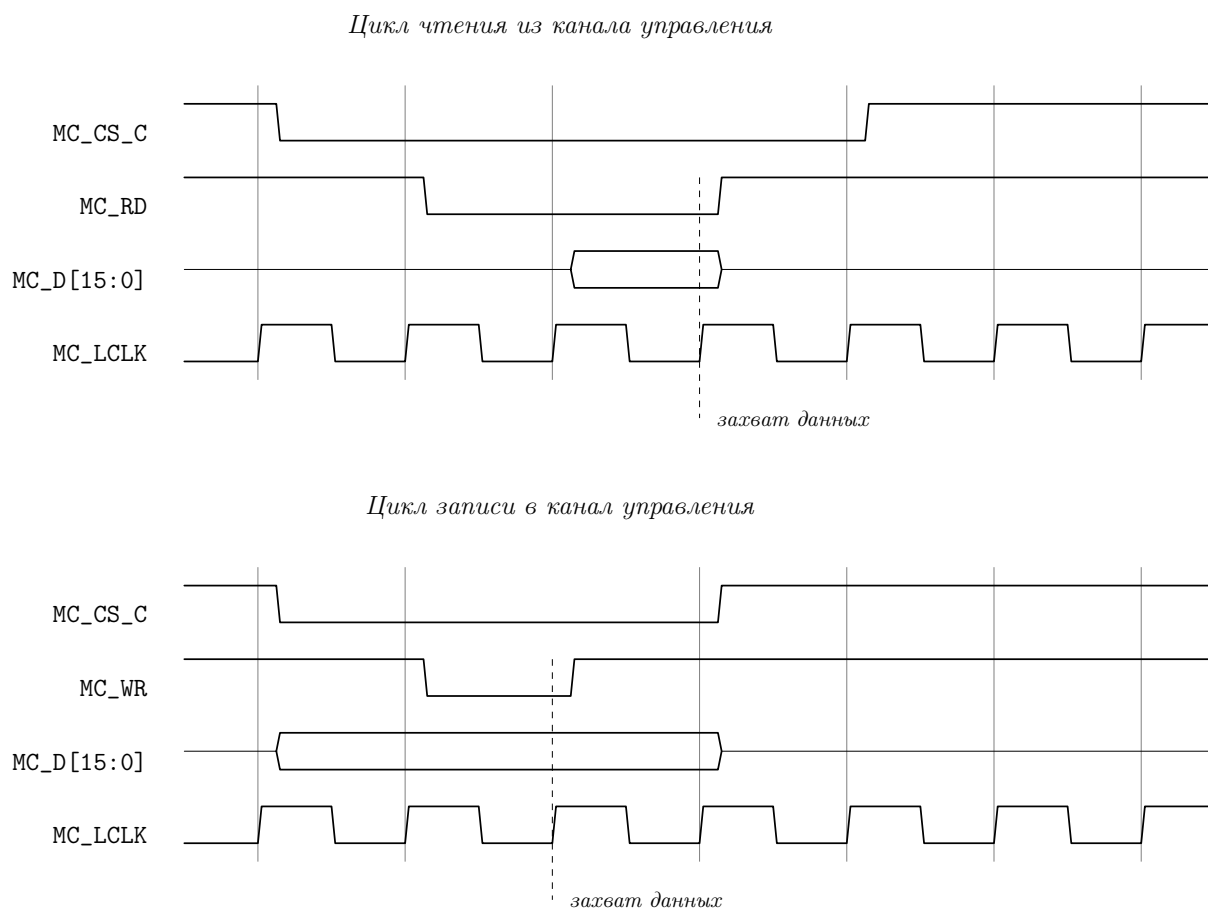


Рис. 16. Временные диаграммы для циклов чтения/записи по каналу управления ПЛИС – МК. Интерфейс полностью синхронный с синхронизацией по нарастающему фронту MC_LCLK.

захвата. Выходные для МК сигналы устанавливаются не более чем через 3 нс после нарастающего фронта MC_LCLK. Все времена приведены для сигналов непосредственно на контактах МК.

Сигналы прерывания представляют собой импульсы отрицательной полярности с длительностью около 100 нс.

Протокол приёма данных из буферной памяти ПЛИС через канал данных выглядит следующим образом:

- сразу после окончания сигнала GATE (или после окончания формирования командного СБРОСа) ПЛИС устанавливает сигнал запроса MC_RQ_D и одновременно генерирует прерывание MC_IRQ1 для циклического запуска (внешнего или внутреннего) или прерывание MC_IRQ2 для командного запуска;
- прерывание обрабатывается драйвером обслуживания канала данных в ядре Linux, который программирует контроллер прямого доступа к памяти (DMA) на выполнение пакетных циклов чтения в канале данных по 16 16-битных слов;
- циклы чтения выполняются до тех пор, пока ПЛИС не сбросит сигнал запроса MC_RQ_D, при этом допускается выполнение холостых циклов чтения после сброса MC_RQ_D, поскольку контроллер DMA всегда программируется на считывание достаточно большой (фиксированной) порции данных и сам по себе состояние MC_RQ_D не проверяет.

Чтение/запись в канал управления возможны только в том случае, если сигнал MC_WAIT_C неактивен, также чтение/запись запрещаются в момент активного состояния GATE (цикл приёма и накопления данных) или MC_RQ_D (выдача данных в канал данных). Поскольку сигнал GATE является внешним асинхронным сигналом, то для надёжной работы, перед выдачей команд в канал управления, циклические запуски должны блокироваться посылкой спецкоманды 0x100 (единственная команда, которую можно послать в момент активного GATE или MC_RQ_D), с последующим ожиданием перехода MC_WAIT_C в неактивное состояние.

Для сброса (приведения в исходное состояние) контроллера канала управления в ПЛИС применяется сигнал SOFT_RESET (с длительностью импульса > 50 нс). Для сброса (приведения в исходное состояние) всей логической схемы в ПЛИС применяется установка активного уровня SOFT_RESET, не менее чем через 100 нс после этого выполнение цикла чтения по каналу управления (для генерирования импульса MC_CS_C), и далее, не менее чем через 100 нс после этого, возврат SOFT_RESET в неактивное состояние. Под исходным состоянием понимается состояние, в котором схема (или участок схемы) оказывается непосредственно после подачи питания и все программируемые параметры принимают дефолтные значения.

Интерфейс ПЛИС с контроллером USB

Интерфейс с контроллером USB соответствует спецификации на микросхему CY7C68001-56PVXC [16], имеет: совмещенную двунаправленную шину данных (FD)

для канала управления (байтовая) и для канала данных (двухбайтовая); три линии адресации (FIFOADR) буферов (Endpoints); два флага (FLAGB, FLAGC) разрешения доступа к буферам для приема и выдачи данных для выбранных Endpoints (6 и 2); сигналы чтения и записи (SLOE, SLWR, SLRD); сигнал конца записи (PKTEND). В управлении контроллером реализуются процедуры анализа окончания режима эnumерации контроллера на USB шине (на готовности к работе) по сигналу INT и конфигурирования (программирования) флагов FLAGB и FLAGC.

С помощью чтения памяти EEPROM, подсоединенной к контроллеру по шине I2C, при включении питания в контроллере USB устанавливаются: синхронная мода, работа с внешней тактовой частотой, низкие активные уровни сигналов интерфейса, VID, PID и DID идентификаторы модуля, использование встроенного дескриптора конфигурации.

Приложение С. Таблицы команд управления модулем ADC-32ATC

В приведённых ниже таблицах команд все числа приводятся в шестнадцатеричном виде, используются следующие обозначения (при управлении от МК все байты посылаются в младшем байте 16-битного слова, старший байт равен 0, за исключением особых случаев при использовании специальных битов 0x100 (блокировка циклов) и 0x200 (посылка адресов и данных SPI)):

8N или 4N — байт, в младшие 4 бита которого записано значение N, в старшие — 8 или 4 соответственно;

FX или EX — байт, в младшие 4 бита которого записано значение X, в старшие — 0xF или 0xE соответственно;

NN — байт со значением $NN + 0x40$ (в случае задержки триггера для NN необходимо использовать диапазон значений от $-0x3F$ до $+0x3F$, при этом реальное значение задержки будет $NN + 0x40$, однако с учётом того, что в сигналах стоит постоянная задержка 64 отсчёта, числа от $-0x3F$ до $+0x3F$ допускают удобную интерпретацию: это задержка относительно ситуации, когда физические сигнал и триггер приходят на модуль одновременно (с точностью до внутренней задержки самих АЦП));

1XX — 16-битное число, в котором младший байт может быть любым, старший равен 1;

HN LN — соответствует последовательности 2-х байт ($0x40 + \text{high}$) и ($0x40 + \text{low}$), где high — старшие 6 бит числа NN, low — младшие 6 бит числа NN (исключение для задания числа стартов: последовательность двух байт ($0x40 + \text{high}$) и low, где high — старшие 6 бит числа NN, low — младшие 7 бит числа NN, нулевые low запрещены);

Hh Ll — соответствует последовательности 2-х байт (0x40 + high) и (0x40 + low), где high – старшие 6 бит числа hl, low – младшие 6 бит числа hl.

Пример последовательности посылок в канал управления для задания числа стартов 1000 (в десятичной системе) командных СБРОСов, изначально модуль находится в режиме А (генерация и приём данных) и после отправки команды снова оказывается в режиме А (начальный переход в режим А (0x04) можно не делать):

0x04 0x03 0x02 0x8B 0xC4 0x47 0x68 0x00 0x04

То же самое, но с блокировкой СБРОСов на время отправки команды (только при управлении от МК):

0x100 0x04 0x03 0x02 0x8B 0xC4 0x47 0x68 0x00 0x04 0x00

После отправки 0x100 при наличии СБРОСа ПЛИС выставляет до его окончания сигнал MC_WAIT_C, после снятия MC_WAIT_C гарантируется, что СБРОСы заблокированы. Поскольку МК не имеет права делать отправки в канал управления при активном MC_WAIT_C, то описанная выше обработка блокировки СБРОСов работает автоматически без каких-либо дополнительных действий со стороны МК.

Запись байта данных D в регистр с 12-битным адресом A для устройства с кодом выборки S по SPI, операторы >>, |, &, ~ понимаются в смысле языка C (при управлении от МК, при управлении по USB необходимо убрать установку бита 0x200, при этом некоторые адреса и данные, которые приводят к отсылке нулевых кодов, станут запрещёнными):

0x04 0x03 0x02 (0x80|S) (((A>>8)&0xF)|0x210)
(((A&0xFF)~0x10)|0x200) ((D~0x10)|0x200) 0x00 0x04

То же самое для чтения по SPI, момент чтения отмечен точкой:

0x04 0x03 0x02 (0x80|S) (((A>>8)&0xF)|0x290)
(((A&0xFF)~0x10)|0x200) 0x10 . 0x00 0x04

Байт перед точкой никакой информации не несёт и может быть любым ненулевым.

Для автоматизации управления модулем по USB в проекте для ПЛИС реализована возможность загрузки пакетов команд управления из внутреннего ROM ПЛИС. Для этого реализованы команды включения и выключения загрузки управления из ROM и программного задания параметров загрузки (начальный адрес и количество кодов) (см. таблицы 5 и 4). Старт исполнения последовательности команд из ROM инициирует команда включения загрузки управления из ROM. Окончание загрузки из ROM и её выключение происходит автоматически по чтению заданного количества кодов, специальному коду конца (FFh) при задании нулевого количества кодов или при исполнении записанной в ROM явной команды выключения загрузки управления из ROM. На время чтения управления из ROM аппаратно блокируется управление и от USB, и от МК.

Таблица 2: Коды выборки SPI-устройств или логических блоков в ПЛИС

Код вы- борки	Устройство
0h	Микросхема AD9222 номер 0
1h	Микросхема AD9222 номер 1
2h	Микросхема AD9222 номер 2
3h	Микросхема AD9222 номер 3
4h	Микросхема AD9516-0 (формирователь тактовых частот АЦП)
8h	Управление обработкой данных
9h	Управление форматированием данных
Ah	Управление тестированием
Bh	Управление циклами

Таблица 3: Коды управления по состояниям модуля.

Код ко- манды	Состояние А (генера- ция и прием данных)	Состояние В (го- товность к сессии управления)	Состояние С (сессия управления)
00h	Запуск теста фрей- ма, запуск разовой последовательно- сти/автоциклов, по- вторное чтение памяти СБРОСа	Очистка памяти СБРО- Са	Конец сессии управле- ния
01h	Блокировка повторных чтений памяти СБРО- Са	Генерация синхроим- пульса фазирования AD9516-0	-
02h	Разрешение повторных чтений памяти СБРО- Са	Переход к состоянию С	-
03h	Переход к состоянию В	Передача управления от USB к МК и переход к состоянию А ⁵	-
04h	-	Переход к состоянию А	-
1XXh	Блокировка всех цик- лов при управлении от МК	-	-

⁵возврат управления к USB происходит по сигналу МК SOFT_RESET

2XXh	-	-	Данные и адрес SPI (включая нулевые значения XX) при управлении от МК
8Nh	-	-	Выбор устройства N(первый код сессии управления)

Таблица 4: Адресные команды сессии управления.

Адресная команда (h)	Управление циклами	Управление тестированием	Управление обработкой данных	Управление форматированием данных
A1 NN	NN — период следования триггеров в разовой последовательности	NN — канал сбора тестовых данных / длина блока чтения из ROM	NN — задержка триггера в тактах	-
A2 NN	NN — период следования триггеров в автоцикле	NN — выбор блока (строба) для изменения фазы	NN — длина последовательности (кол-во) выборок	-
A3 NN	-	-	NN — макс. допустимое кол-во каналов	-
A4 4N	-	-	N — мода отбора по порогу	-
A5 4N	N — пауза в автоциклах	-	N — режим чередования АЦП: 0 – x1, 1 – x2, 2 – x4	-
A6	Признак следования кодовых команд	-	-	-
C3 HN LN	NN — кол-во триггеров в автоцикле	NN — эталон / начальный адрес чтения из ROM	NN — порог для всех каналов	-
C4 HN LN	NN — кол-во триггеров в разовой последовательности	-	-	-
C5 Hh Ll	-	-	hl — маска каналов 11–00	-

C6 Hh Ll	-	-	hl — маска каналов 23–12	-
C7 Hh Ll	-	-	hl — маска каналов 31–24	-
EX Hh Ll ... Hh Ll	-	-	hl — индивидуальные пороги каналов 0Xh и далее	-
FX Hh Ll ... Hh Ll	-	-	hl — индивидуальные пороги каналов 1Xh и далее	-
EX 4N ... 4N	-	N — индивидуальные задержки в тактах оцифровки для каналов 0Xh и далее (для всех режимов, не только тестирования)	-	-
FX 4N ... 4N	-	N — индивидуальные задержки в тактах оцифровки для каналов 1Xh и далее (для всех режимов, не только тестирования)	-	-

Таблица 5: Кодовые команды сессии управления.

Кодовая команда	Управление циклами	Управление тестированием	Управление обработкой данных	Управление форматированием данных
00h	Конец сессии управления	Конец сессии управления	Конец сессии управления	Конец сессии управления
01h	Блокировка внешних циклов и автоциклов	Отключение тестирования	-	Исключение маски из заголовка события
02h	Разрешение внешних циклов и автоциклов	Включение тестирования	-	Включение маски в заголовок события

03h	Старт (включение) автоциклов (выключение внешних циклов)	-	-	-
04h	Включение внешних циклов (выключение автоциклов)	-	-	-
05h	Вкл. тестовых данных	Стоп сбора тестовых данных / при отключенном тестировании — выключение загрузки управления из ROM	-	-
06h	Выкл. тестовых данных	Старт сбора тестовых данных / при отключенном тестировании — включение загрузки управления из ROM	-	Форматирование с упаковкой данных
07h	Фиксированные тестовые данные	Сбор по всем каналам	-	Форматирование без упаковки данных
08h	Псевдослучайные тестовые данные	Сбор по заданному каналу	-	Сброс флага запрета программного объединения каналов
09h	Разрешение внешних триггеров в автоциклах	Выдача тестовых данных в канал управления	-	Установка флага запрета программного объединения каналов
0Ah	Блокировка внешних триггеров в автоциклах	Выдача тестовых данных в канал данных	-	-
0Bh	-	Тест — выборки	-	-

0Ch	-	Тест — количество выборок + ОШ	-	-
0Dh	-	Тест — количество ошибок	-	-
0Eh	-	Инкремент фазы стробирования	-	-
0Fh	-	Декремент фазы стробирования	-	-

Приложение D. Формат данных СБРОСа

Данные СБРОСа, содержащие N событий, имеют следующий формат (смещение в памяти увеличивается сверху вниз):

Заголовок СБРОСа
Заголовок события 0
Полезные данные события 0
Заголовок события 1
Полезные данные события 1
.....
Заголовок события N-1
Полезные данные события N-1

Заголовок СБРОСа бывает двух видов. Для данных, которые считываются непосредственно из ПЛИС (т.е. из файлов устройств `/dev/adc-data-0` и `/dev/adc-data-1`, см. раздел 6.6), ПЛИС генерирует собственный заголовок, который потом заменяется на заголовок другого формата сервером выдачи данных через Ethernet (см. раздел 6.7). Поэтому данные, выдаваемые через Ethernet, имеют отличающийся заголовок. Форматы обоих заголовков, а также заголовка события и полезных данных события, детально описаны ниже. Замена заголовка СБРОСа необходима для добавления в него некоторой дополнительной информации и для обеспечения совместимости с программным обеспечением системы сбора данных (ССД) эксперимента ВЕС.

Для временных измерений (временные штампы событий и длительность сигнала GATE) могут применяться синхронизованные с тактовой частотой АЦП версии сигналов GATE и START.

Все многобайтовые величины (16- и 32-битные слова), фигурирующие ниже, имеют формат little-endian, то есть младший байт располагается в памяти по меньшему адресу.

Формат заголовка СБРОСа (при непосредственном чтении из ПЛИС)

Смещение \ биты	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0xFFFFC (сигнатура СБРОСа)															
1	число выборок				ЗПО	ОД		УД		М	FLE	FLM	FLD	FL1	FL2	FL3
2	1	длительность сигнала GATE, биты [30:16]														
3	длительность сигнала GATE, биты [15:0]															
4	1	размер СБРОСа, биты [30:16]														
5	размер СБРОСа, биты [15:0]															
6	1	количество событий, биты [30:16]														
7	количество событий, биты [15:0]															

Поля заголовка имеют следующий смысл:

число выборок число выборок на канал;

ЗПО бит запрета программного объединения каналов, имеет значение только для режимов с чередованием x2, x4 (см. раздел 4), при единичном значении бита построитель событий в ССД обязан отключить программное объединение каналов, даже если модуль на самом деле работает в режиме x2 или x4, данная возможность предусмотрена исключительно для отладки;

ОД отбор данных, тип отбора сработавших каналов (0 – по превышению какой-либо выборки над порогом, 1 – по превышению над порогом среднего значения по всем выборкам, остальные значения зарезервированы);

УД тип упаковки полезных данных события (3 – неупакованный формат, 2 – упакованный формат, остальные значения зарезервированы);

М бит наличия маски сработавших каналов в заголовках событий (0 – отсутствие, 1 – присутствие);

FLE	флаг пропущенных сигналов START, т.е. событий (1 – есть пропущенные события, 0 – нет пропущенных событий), пропуск события может происходить, если нарушается условие (1) из раздела 7.5 на минимальный интервал между сигналами START;
FLM	флаг переполнения памяти СБРОСа (пока резервный, всегда 0);
FLD	флаг потери (не полного вычитывания) данных предыдущего СБРОСа (пока резервный, всегда 0);
FL1,FL2	режим чередования АЦП по времени, см. раздел 4 (FL1=0,FL2=0 – x1, FL1=0,FL2=1 – x2, FL1=1,FL2=0 – x4, FL1=1,FL2=1 – зарезервировано);
FL3	бит наличия тонкой фазы в заголовке события (1 – значения ϕ_t , ϕ_s присутствуют, 0 – отсутствуют (при этом биты в соответствующих полях заголовка могут иметь произвольные значения));
длительность сигнала GATE	длительность сигнала GATE, измеренная счётчиком с тактовой частотой 160.0 МГц (относительная точность задания частоты $\sim 10^{-4}$);
размер СБРОСа	размер данных СБРОСа без заголовка в 16-битных словах;
количество событий	количество событий в СБРОСе.

Формат заголовка СБРОСа (при использовании сервера выдачи данных)

Смещение \ биты	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0 (зарезервировано)																															
1	0 (зарезервировано)																															
2	0 (зарезервировано)																															
3	0 (зарезервировано)																															
4	размер СБР0Са в байтах																															
5	0 (зарезервировано)																															
6	0 (зарезервировано)																														FLM	
7	длительность сигнала GATE																															
8	UPTIME																															
9	0 (зарезервировано)																число выборов		ЗПО	ОД		УД		М	FLE	FLM	FLD	FL1	FL2	FL3		
10	0 (зарезервировано)																															
11	0 (зарезервировано)																															
12	0 (зарезервировано)																															
13	0 (зарезервировано)																															
14	0 (зарезервировано)																															
15	0 (зарезервировано)																															

Поля, названия которых совпадают с названиями полей из заголовка СБРОСа ПЛИС, имеют точно такое же назначение, некоторые поля продублированы (например FLM) для совместимости с программным обеспечением ССД ВЕС. Поля заголовка имеют следующий смысл:

UPTIME время в секундах, прошедшее с последней перезагрузки микроконтроллера;

размер СБРОСа в байтах размер СБРОСа в байтах без учёта заголовка СБРОСа.

Формат заголовка события

Смещение \ биты	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0xF1FC (сигнатура события)															
1	0xF7FD (сигнатура события)															
2	ФПС	временной штамп события, биты [30:16]														
3	временной штамп события, биты [15:0]															
4	грубая фаза Φ					тонкая фаза ϕ_t					тонкая фаза ϕ_s					-
5	N[4:0]					N[5]	длина события									
6	маска сработавших каналов, биты [15:0]															
7	маска сработавших каналов, биты [31:16]															

Поля заголовка имеют следующий смысл (относительно деталей вычисления фазы триггера см. раздел 5):

ФПС	флаг пропущенного старта (1 – во время обработки предыдущего события был пропущен сигнал START, 0 – пропуска не было);
временной штамп события	время между передним фронтом сигнала GATE и передним фронтом сигнала START для этого события, измеряется тем же самым счётчиком с тактовой частотой 160.0 МГц, что и длительность GATE (см. описание заголовка СБРОСа);
грубая фаза Φ	грубая фаза триггера, полученная с помощью счётчика с тактовой частотой 480 МГц (относительная точность задания частоты $\sim 10^{-4}$);
тонкая фаза ϕ_t	тонкая поправка к фазе триггера для сигнала триггера (START);
тонкая фаза ϕ_s	тонкая поправка к фазе триггера для синхронизированного с тактовой частотой оцифровки сигнала триггера (START_SYN);
-	зарезервированный бит, пока может иметь любое значение;
N	в режиме работы x1 — число сработавших каналов в событии (должно совпадать с числом единичных битов в маске), в режимах работы x2 и x4 значение этого поля не определено;

длина события	размер полезных данных события в 16-битных словах;
маска сработавших каналов	бит i в маске обозначает наличие (1) или отсутствие (0) канала i в полезных данных события.

Формат полезных данных события (неупакованный)

Данные от канала присутствуют только в том случае, если соответствующий этому каналу бит в маске сработавших каналов в заголовке события равен 1. Выборки для каждого канала записываются подряд в виде 16-битных слов (упорядочены по возрастанию: чем больший порядковый номер имеет выборка, тем большим смещением в данных она обладает). Группы выборок от разных каналов записываются в событие подряд и упорядочены по возрастанию номера канала. Номер канала записывается в выборку по модулю 16 в связи с недостатком места, поэтому раскодирование в общем случае необходимо осуществлять с применением маски сработавших каналов из заголовка события. Формат выборки:

Смещение \ биты	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	номер канала				амплитуда выборки											

Формат полезных данных события (упакованный)

Данные от канала присутствуют только в том случае, если соответствующий этому каналу бит в маске сработавших каналов в заголовке события равен 1. Упаковка для младших 16 каналов (0–15) и старших (16–31) выполняется отдельно. Процедура упаковки выглядит следующим образом: 12-битные выборки от всех сработавших каналов из младшей или старшей 16-канальной группы выписываются подряд: сначала все выборки первого сработавшего канала (по возрастанию номера выборки), потом второго и т.д. Каналы упорядочиваются по возрастанию номера канала, то есть первый сработавший канал — это сработавший канал с наименьшим номером в упаковываемой 16-канальной группе. Потом каждые четыре последовательные выборки в получившейся последовательности объединяются в тетрады размером три 16-битных слова. Если общее число выборок в последовательности не кратно 4, то последняя тетрада дополняется до нормального размера (три 16-битных слова) выравнивающими нулевыми битами. Тетрады записываются в событие подряд: сначала от младшей 16-канальной группы, потом от старшей. Раскодирование необходимо осуществлять с применением маски сработавших каналов из заголовка события. Формат тетрады (`amp10,1,2,3` — амплитуды четырёх последовательных выборок, подлежащие объединению в тетраду):

Смещение \ биты	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	amp11[3:0]				amp10[11:0]											
1	amp12[7:0]								amp11[11:4]							
2	amp13[11:0]												amp12[11:8]			

Список литературы

- [1] Ивашин А.В., Матвеев В.Д., Хохлов Ю.А. Модернизированная система сбора данных установки ВЕС. Технические аспекты. Препринт ИФВЭ 2010-10. Протвино, 2010.
- [2] Сенько В.А., Солдатов М.М., Якимчук В.И. 96-канальный модуль МИСС преобразования «заряд-цифра» для калориметров физики высоких энергий. Препринт ИФВЭ 2012-19. Протвино, 2012.
- [3] Дорофеев В. и др. Новый электромагнитный калориметр модернизированной установки ВЕС. ПТЭ 2016, №5, с. 38-46.
- [4] Kester W. HIGH SPEED SAMPLING ADCs. Analog Devices.
<http://www.analog.com/media/en/training-seminars/design-handbooks/Practical-Analog-Design-Techniques/Section4.pdf>
- [5] Dutton H.J.R. Understanding Optical Communications. September 1998. IBM Corporation, International Technical Support Organization.
- [6] COMPASS collaboration. COMPASS TCS documentation. 2001.
<http://wwwcompass.cern.ch/compass/detector/daq/TCS/index.html>
- [7] Altera Corporation. Quartus 2 Handbook Version 9.1. November 2009.
<http://www.altera.com>
- [8] Analog Devices. Interfacing to High Speed ADCs via SPI. AN-877 Application note.
- [9] Семак А.А. Частное сообщение.
- [10] Ивашин А.В. Модуль TDC-48DT.
<http://pcbech.ihep.su/~ivashin/tdc-48dt-doc.pdf>
<http://mail.ihep.ru/~ivashin/tdc-48dt-doc.pdf>
- [11] Analog Devices. AD9222 — Octal, 12-bit, 40/50/65 MSPS Serial LVDS 1.8V A/D Converter. Rev. F. Data sheet.
- [12] Analog Devices. AD9516-0 — 14-Output Clock Generator with Integrated 2.8 GHz VCO. Rev C. Data sheet.

- [13] Altera Corporation. SERDES Transmitter/Receiver (ALTLVDS). Version 4.0. November 2008. <http://www.altera.com>
- [14] Song J., An Q., Lin S. A High-Resolution Time-to-Digital Converter Implemented in Field-Programmable-Gate-Arrays. IEEE Transactions on Nuclear Science, vol.53, no.1, p.236, February 2006. DOI: 10.1109/TNS.2006.869820
- [15] Altera Corporation. Cyclone III Device Handbook. v2.2. October 2008.
- [16] CYPRESS. CY7C68001 EZ-USB SX2tm High Speed USB Interface Device. Revised November 20, 2013.
- [17] Freescale Semiconductor. MPC8308 PowerQUICC II Pro Processor Reference Manual. Rev. 0, 04/2010. <https://www.nxp.com/>
- [18] Freescale Semiconductor. MPC603e RISC Microprocessor User's Manual. Rev. 3., Q2/2002. <https://www.nxp.com/docs/en/reference-manual/MPC603EUM.pdf>
- [19] Freescale Semiconductor. e300 Power Architecture(TM) Core Family Reference Manual. Rev. 4, 12/2007. <https://www.nxp.com/docs/en/reference-manual/e300coreRM.pdf>
- [20] https://www.altera.com/support/support-resources/download/legacy/jam/dnl-byte_code_player.html
- [21] <http://urjtag.org/>
- [22] <https://www.denx.de/wiki/U-Boot>
- [23] <https://www.kernel.org/>
- [24] <https://www.gnu.org/software/libc/>
- [25] <https://busybox.net/>
- [26] <http://www.openssh.com/>
- [27] <https://buildroot.org/>
- [28] <https://gcc.gnu.org/>
- [29] <https://www.gnu.org/software/binutils/>
- [30] <http://crosstool-ng.github.io/>

Рукопись поступила 25 декабря 2017 г.

Препринт отпечатан с оригинала-макета, подготовленного авторами.

Е.В. Волков и др.

Автономный модуль амплитудно-цифрового преобразования ADC-32АТС
с оцифровкой формы импульса.

Оригинал-макет подготовлен с помощью системы **L^AT_EX**.

Подписано к печати 28.12.2017. Формат 60 × 84/16. Цифровая печать.
Печ.л. 3,87. Уч.-изд.л. 5,76. Тираж 80. Заказ 11. Индекс 3649.

НИИ «Курчатовский институт» - ИФВЭ
142281, Московская область, г. Протвино, пл. Науки, 1

www.ihep.ru; библиотека <http://web.ihep.su/library/pubs/all-w.htm>

Индекс 3649

П Р Е П Р И Н Т 2017–8,
НИИЦ «Курчатовский институт» - ИФВЭ, 2017
