ПРИБОРЫ И ТЕХНИКА ЭКСПЕРИМЕНТА, 2027, № 1

ТЕХНИКА ЯДЕРНОГО ЭКСПЕРИМЕНТА

УДК 533.582

КОНТРОЛЛЕРЫ КАРКАСОВ РЕГИСТРИРУЮЩЕЙ ЭЛЕКТРОНИКИ С USB-ИНТЕРФЕЙСОМ ДЛЯ ЭКСПЕРИМЕНТОВ В ФИЗИКЕ ВЫСОКИХ ЭНЕРГИЙ

©2027 г. В. Д. Матвеев, А. П. Филин*, В. И. Якимчук

Институт физики высоких энергий им. А.А. Логунова

Национального исследовательского центра "Курчатовский институт"

Россия, 142281, Протвино Московской обл., пл. Науки, 1

**E-mail: Alexey.Filin@ihep.ru*

Поступила в редакцию 04.06.2026

После доработки 04.09.2026

Принята к публикации 02.11.2026

Описаны системные и автономные контроллеры каркасной регистрирующей электроники, разработанные для набора данных экспериментальными установками в физике высоких энергий. В качестве интерфейса подключения контроллеров использована универсальная последовательная шина USB-2 в высокоскоростном режиме с массовым типом обмена данными. Даны схемы контроллеров, приведен способ реализации команд чтения и записи над регистрами системных контроллеров посредством пакетов USB, описаны протокол обмена с контроллерами и архитектура программного обеспечения, реализующая его.

Ключевые слова: система сбора данных, универсальная последовательная шина, эксперимент в физике высоких энергий.

1. ВВЕДЕНИЕ

Экспериментальные установки в физике высоких энергий, как правило, характеризуются требованиями многоканального и высокоскоростного приема и обработки данных эксперимента и многоканального управления. Количество каналов данных может достигать десятков тысяч, а скорость управления данными и их приемом в системе сбора данных (ССД) прямо определяет эффективность эксперимента. Для решения этих задач широкое применение нашли различные магистрально-модульные каркасные системы, позволяющие распараллеливать функции управления и приема многоканальных данных (CAMAC, СУММА [1], VME, МИСС [2],

ЕвроМИСС [3]), эти системы продолжают использоваться. Примером установки с широкой номенклатурой их использования является установка эксперимента ОКА [4] с 12000 каналов регистрации и приема данных.

Для связи регистрирующей электроники с компьютерами ССД длительное время использовались специализированные шинные интерфейсы Q-bus, ветвь СУММА/CAMAC, VME, специализированная интерфейсная карта PCI7200 (производитель ADLINK) и др.

Промышленное развитие компьютерной техники с быстрой сменой ее поколений и внутренних шин данных привело к проблемам адаптации ССД к новым компьютерам, а появление в компьютерах стандартизированных средств обмена данными с внешними устройствами (USB и Ethernet) поставило задачу их применения в ССД.

Для программного доступа к каркасной регистрирующей электронике физического эксперимента был разработан набор контроллеров с интерфейсом USB-2, полностью заменяющий все устаревшие системные (каркасные) и автономные контроллеры со специализированными интерфейсами.

Интерфейс USB-2 является стандартным для материнских плат компьютеров, он обеспечивает прямое кабельное соединение длиной до 5 м и опосредованное соединение с промежуточными хабами длиной до 30 м, чего достаточно для подключения каркасной электроники к компьютерам системы сбора данных физического эксперимента. Теоретическая скорость передачи данных 480 Мб/с в высокоскоростном (high-speed) режиме достаточна для задач, решаемых используемой электроникой [5]. Реализация интерфейса дешевле таковой для любого аналога в настоящем и обозримом будущем.

Упомянутые выше специализированные интерфейсы обеспечивают связь с шиной данных компьютера в режиме пословных передач, что дает возможность достигать максимально возможной для компьютера скорости передачи как команд управления устройствами, так и данных. Через USB и Ethernet системными средствами реализуется только пакетный обмен данными, а для управления внешними устройствами предусматриваются специальные командные пакеты и процедуры. Поэтому, несмотря на высокую скорость передачи самих пакетов, реализация (достижение) сопоставимо быстрой передачи команд управления каркасным контроллерам представляет специальную задачу внедрения USB-2 в ССД физической установки как с аппаратной, так и с программной точки зрения, и этому вопросу посвящена значительная часть статьи по форматам команд и программной архитектуре.

Для программного управления модулями регистрирующей электроники были разработаны системные контроллеры в стандартах МИСС (ЛЭ98), ЕвроМИСС (ЕМ11), СУММА (ЛЭ99С), CAMAC (ЛЭ99К). Для набора физических данных были разработаны автономные контроллеры в стандартах МИСС (ЛЭ97) и ЕвроМИСС (ЕМ8).

Начало разработок контроллеров было положено успешной модернизацией приема данных в системе сбора данных установки ВЕС через автономные контроллеры ЛЭ68 с дополнительными модулями AC-USB-согласования с шиной USB-2 с группой объединительных USB-хабов [6].

2. ИНТЕРФЕЙС К USB

Единственным активным устройством на шине USB является компьютер, который инициирует все передачи. Все передачи идут между компьютером и USB-устройствами, передачи никогда не идут между устройствами. Время на шине USB делится на фреймы длительностью 1 мс в низко- или полноскоростном (low-/full-speed) режимах (1.5/12 Мб/с) и микрофреймы длительностью 125 мкс в высокоскоростном режиме (480 Мб/с). Каждая передача данных с массовым типом обмена данными (bulk mode) состоит из одной и более транзакций, эта передача гарантирует целостность передаваемых данных. Программно-аппаратные средства компьютера планируют каждую транзакцию так, чтобы в высокоскоростном режиме она атомарно завершалась в рамках одного микрофрейма. В рамках одного микрофрейма может быть выполнено несколько независимых транзакций. USB-1/2 работают в полудуплексном режиме, так как для передачи используется одна витая пара, передающая пакеты только в одну сторону в каждый момент времени. Каждый пакет, передаваемый путем массового обмена данными, кроме последнего пакета, содержит данные размером, равным размеру буфера конечной точки (endpoint) (максимально 512 байт для высокоскоростного режима). Размер данных в последнем пакете передачи по соглашению меньше размера буфера конечной точки, он сигнализирует специализированному драйверу о конце передачи.

Во всех разработанных контроллерах для выдачи данных в шину и управления по USB используется микроконтроллер CY7C68001 [7] в предустановленной конфигурации, реализующей массовый тип обмена данными по шине USB, высокоскоростной режим передачи и сдвоенные по 512 байт буферы обмена для обоих направлений (передачи данных в компьютер через конечную точку №6 и приема команд и данных из компьютера через конечную точку №2). Сдвоенные буферы обмена позволяют совместить во времени запись данных в буфер

микроконтроллера и чтение данных из буфера для достижения максимально возможной скорости обмена.

При включении питания микроконтроллер считывает конфигурационные данные из внешней энергонезависимой памяти EEPROM по шине I2C. На основе полученных данных аппаратно задаются следующие параметры: синхронный режим работы интерфейса на внешней тактовой частоте (40/48 МГц), активный низкий уровень (отрицательная полярность) основных управляющих сигналов, идентификаторы модуля (VID, PID, DID), а также активируется использование встроенного дескриптора конфигурации по умолчанию [7]. На рис. 1 показана унифицированная для всех разработанных модулей блок-схема управления USB.

Рис. 1. Блок-схема управления USB.

В управлении данными памяти FIFO в каналах чтения и записи необходимы для согласования скоростей работы USB-микроконтроллера и остальных частей модулей. Используемые сигналы связи USB-микроконтроллера для управления (USB_control): входные INTN, READY, FLAGBN, FLAGCN; выходные FIFOADR[2..0], CSN, SLOEN, SLWRN, SLRDN, PKTENDN, WAKEUPN; двунаправленный INT_CLK.

Двунаправленная шина данных (USB_D) – 16-разрядная параллельная. Поскольку интерфейс с USB-контроллером в каждый момент времени может находиться в состоянии выполнения только одной операции (запись или чтение на шине USB_D-микроконтроллера), на уровне управления интерфейсом введен аппаратный приоритет для операций чтения (в том числе для приема команд управления от компьютера) с временем ожидания ответа схемы (длительностью 25 мс) на принудительное завершение начатого чтения.

3. КОНТРОЛЛЕРЫ КАНАЛА УПРАВЛЕНИЯ

Описываемый набор контроллеров обеспечивает управление и контроль аппаратуры электронных систем МИСС (ЛЭ-98), ЕвроМИСС (ЕМ-11), СУММА (ЛЭ-99С) и КАМАК (ЛЭ-99К). Новые контроллеры являются модификацией контроллеров, которые осуществляли подключение электронных систем к персональному компьютеру с помощью интерфейса PCI-Qbus [1–3]. Аппаратная модификация заключается в замене узла дешифрации адресов шины Q-bus на новый узел, описанный далее.

Структурная схема этого узла представлена на рис. 2. Из компьютера информация по каналу USB (микроконтроллер CY7C68001 – интерфейсная логика (IL)) поступает во входное FIFO, а ответная информация передается в компьютер из выходного FIFO через тот же канал.

Для выполнения операции записи в какой-либо регистр контроллера необходимо указать адрес этого регистра и код данных для записи, а для операции чтения – только адрес регистра. Поскольку разрядность шин данных микроконтроллера и электронных систем составляет 16 бит, требуется передача двух слов для операции записи. Первое слово указывает адрес регистра и тип операции, второе – информационный код. Для операции чтения достаточно одного слова, но для простоты реализации схемы всегда передается два слова. При выполнении команды чтения значение второго слова игнорируется. Из входного FIFO пара слов переписывается в регистры Рг.1 и Рг.2. Содержимое Рг.1 поступает на дешифратор (Дш) для формирования сигнала записи-чтения в один из регистров P1–Pn (W1–Wn и R1–Rn). После выполнения требуемой операции контроллер формирует ответный пакет и размещает его в выходном FIFO. Необходимые сигналы управления формирует блок управления (Control Unit). Подробнее о пакетах написано ниже.

Рис. 2. Структурная схема узла дешифрации адресов регистров контроллеров канала управления

Будучи ведомым устройством, контроллер после включения питания переходит в состояние ожидания команд от компьютера.

Пакет, посылаемый компьютером в контроллер, состоит из пар 16-разрядных слов. Каждая пара определяет адрес (номер) регистра (слово 1) и код данных для записи или 0 для чтения (слово 2). Структура пары слов дана в табл. 1, 2.

Таблица 1. Командный пакет, кодирование разрядов в слове 1

Номер разряда	15–8	7	6	5	4–0
Значение	0	1–*Запись*, 0–*Чтение*	0	0	Адрес регистра

Таблица 2. Командный пакет, кодирование разрядов в слове 2

Номер разряда	15–0
Значение	Код для записи – для операции *Запись*, 0 для операции *Чтение*

На каждую пару слов командного пакета контроллер после выполнения указанной команды готовит ответную пару 16-разрядных слов. Передача ответного пакета в компьютер происходит после завершения обработки последней пары слов входного пакета. В ответном слове 1 разряды 7–0 декодируются аналогично командному слову 1, разряды 15–8 декодируются значениями:

- 0x0 – успешное выполнение команды,
- 0x80 – превышение времени ожидания ответа при выполнении операции в магистрали системы,
- 0x40 – при задании неверного номера регистра (более 7),
- 0x20 – если принятый контроллером пакет содержит нечетное число 16-разрядных слов,
- 0x10 – при ответе на команду инициализации контроллера

Ответное слово 2 содержит код для записи – для операции *Запись*, прочитанный код – для операции *Чтение.*

Архитектура программного обеспечения управления каркасной электроникой состоит из четырех уровней. К компьютеру системный контроллер подключается как USB-устройство.

Первый нижний уровень (транспортный), реализованный как специализированный драйвер ядра Linux (bcudrv), осуществляет обмен пакетами с USB-устройством. Сразу после отправки командного пакета драйвер выполняет чтение ответного пакета, конвейерный прием с выполнением командных пакетов на стороне контроллера с использованием встроенной двойной буферизации не реализован. Драйвер предоставляет доступ к системному контроллеру посредством специального файла устройства. Файл устройства может быть открыт в каждый момент времени только одним пользовательским процессом. Критические секции в драйвере реализуют смену состояний драйвера и гарантируют атомарность выполнения команд контроллера, закодированных в одном пакете. Замена контроллера СУММА ЛЭ88С, подключенного через адаптер PCI-Qbus с драйвером, не гарантирующим атомарность выполнения команд, на контроллер ЛЭ99С с интерфейсом USB, позволила в одном из экспериментов ИФВЭ избавиться от интерференции выполнения команд модулями, установленными в один каркас и управляемыми разными пользовательскими процессами.

Второй уровень (командный), реализованный в библиотеке-компаньоне драйвера (libbcu), осуществляет конфигурирование и управление состоянием драйвера, обмен пакетами с контроллером посредством специального файла устройства, кодирование команд над регистрами контроллера в отправляемый пакет и декодирование результатов выполнения из принятого пакета. Команды, кодируемые в один пакет, не должны зависеть друг от друга по результату выполнения.

Третий уровень (режимов работы), реализованный в библиотеке libcil, осуществляет алгоритмы режимов работы контроллера (адресный, последовательное чтение информации (ПЧИ), последовательное чтение номеров (ПЧН)) посредством команд над регистрами контроллера независимо от типа программного ввода–вывода. Поддерживается доступ к

регистрам тремя типами ввода–вывода: отображение в оперативную память (memory mapping), через файловый интерфейс к портам ввода-вывода (IO ports) и пакетный интерфейс USB. Прикладной интерфейс (API) данной библиотеки унифицирован для всех используемых типов контроллеров, подключаемых через промежуточные шины и адаптеры (PCI, Q-bus, ветвь СУММА/CAMAC, VME, USB), и не зависит от транспортного и командного уровней. Для контроллеров, имеющих пакетный режим выполнения команд, в библиотеке реализован интерфейс для кодирования набора команд в список транспортных пакетов (кодирование команд неявно добавляет в список новый пакет после заполнения текущего), их выполнения и декодирования ответов. Для данной библиотеки с помощью утилиты ctypesgen сгенерированы Python-обвязки и написаны Python-модули для доступа к каркасной электронике из приложений на языке Python.

Четвертый уровень (прикладной) реализует управление каркасной электроникой в приложении конфигурирования и чтения триггерной электроники, системе сбора данных физического эксперимента и приложениях тестирования каркасной электроники.

Такая архитектура программного обеспечения позволила сделать специализированный драйвер ядра bcudrv не зависящим от типа контроллера с USB-интерфейсом, а значит, более надежным, и предоставить прикладному программисту унифицированный интерфейс доступа к каркасной электронике, не зависящий от типа системного контроллера, интерфейсных адаптеров и шин, что упрощает ее использование.

В ядрах Linux версий 2 и 3 оповещение об окончании передачи от подсистемы USB ядра Linux происходит на границе микрофреймов USB длительностью 125 мкс, что задает минимальное время выполнения команд. При отсутствии полезного сигнала аналого-цифрового преобразователя (АЦП) практически всегда дает в данных ненулевое постоянное смещение (пьедестал). С целью уменьшения количества шума в данных это смещение необходимо хранить в специальной ячейке памяти АЦП (пьедестальной памяти) и вычитать из оцифрованного значения. Для этого пьедестальные памяти перезаписываются нулями в адресном режиме, смещение для каждого канала измеряется и считывается в режиме ПЧИ с помощью набора событий при отсутствии сигналов в детекторе (пьедестальных событий). Усредненное смещение с небольшой добавкой (дельтой) для уменьшения шума записывается в ячейку памяти АЦП в адресном режиме. Система сбора данных сохраняет усредненное смещение, его среднеквадратичное отклонение для каждого канала АЦП и дельту как пьедестальное событие

для контроля работы детектора и АЦП. Чтение набора пьедестальных событий АЦП калориметрии и запись их пьедестальных памятей системными контроллерами происходят в каждом цикле ускорителя. При количестве каналов АЦП в несколько тысяч время выполнения команд системными контроллерами каркасов АЦП критически важно и не может быть увеличено. Чтобы обойти данное ограничение Linux, авторы разработали метод ускоренного завершения приема ответа с помощью цикла ожидания признака конца данных. Такой признак передается в последнем слове ответного пакета, принимаемом в буфер когерентного DMA. При этом посылку пользовательским процессом следующего пакета команд и подготовку к приему ответного необходимо проводить асинхронно к штатным для ядра Linux оповещениям об окончании передач предыдущих пакетов и ответам на них (драйвер получает оповещения в атомарном контексте прямо из обработчика прерывания). Получение структур командного и ответного пакетов и их возврат при обработке оповещений реализованы критическими секциями драйвера в системном вызове и нижней половине обработчика оповещения с использованием списков свободных и занятых пакетов, выделяемых при конфигурировании драйвера. Синхронизация между системным вызовом и нижней половиной обработчика оповещения (ожидание свободных пакетов, ожидание вызова обработчика оповещения после возможного таймаута цикла ожидания ответа) реализована с помощью очереди ожидания Linux [8].

Для сравнения системных контроллеров были проведены измерения времени выполнения команд на испытательном стенде. Программные измерения проводились на суммарной выборке 10^6 команд для каждого типа контроллера и интерфейсных шин с контролем осциллографом. В табл. 4 приведены измеренные времена выполнения команд непосредственно системными контроллерами без учета времени выполнения по каркасной шине. Измеренное среднее время выполнения одиночных команд контроллерами с USB-интерфейсом при использовании данного механизма не зависит от типа контроллера и равно 14 мкс. При выполнении независимых команд пачкой по 100 команд время выполнения одной команды снижается до 1.7 мкс. В табл. 5 приведены измеренные времена выполнения команд модулем Т162 через системные контроллеры СУММА с учетом времени выполнения по каркасной шине. При доступе через контроллер с USB-интерфейсом одиночные команды выполняются в среднем за 14 мкс, в пачке по 100 команд за 3.0 мкс. В ядрах Linux версий 4 реализовано оповещение об окончании передачи вне границ микрофреймов. На тестах времени выполнения команд в ядрах Linux

версий 4 завершение по штатному оповещению проигрывает методу ускоренного завершения, реализованному авторами, в 1.5 раза для одной команды с учетом времени выполнения по каркасной шине, разница уменьшается с ростом количества команд в пакете.

Таблица 4. Время выполнения команд непосредственно системными контроллерами

Тип системного контроллера	Интерфейсные шины	Программный ввод–вывод	Время выполнения одиночной команды, мкс	Время выполнения команды в пакете, мкс
ЛЭ83 (МИСС), ЛЭ88С (СУММА), ЕМ1 (ЕвроМИСС)	PCI-VME-Qbus (Bit3-V07)	Отображение в память (memory mapping)	3.8	–
ЕМ1 (ЕвроМИСС)	PCI-Qbus	Порты ввода-вывода (IO ports)	6.9	–
ЛЭ99С (СУММА), ЛЭ99К (САМАС), ЛЭ98 (МИСС), ЕМ11 (ЕвроМИСС)	USB	USB-пакеты	14	1.7

Таблица 5. Время выполнения команд модулем СУММА Т162

Тип системного контроллера СУММА	Интерфейсные шины	Программный ввод–вывод	Время выполнения одиночной команды, мкс	Время выполнения команды в пакете, мкс
К62	PCI-VME-ветвь СУММА (Bit3-V02)	Отображение в память	4.4	–
ЛЭ88С	PCI-VME-Qbus (Bit3-V07)	Отображение в память	3.8	–
ЛЭ99С	USB	USB-пакеты	14	3.0

4. АВТОНОМНЫЕ КОНТРОЛЛЕРЫ КАНАЛА СБОРА ДАННЫХ

Новые автономные контроллеры ЛЭ-97 (МИСС) и ЕМ-8 (ЕвроМИСС) являются модификацией автономного контроллера ЛЭ-85 [2] с параллельной шиной передачи информации. Контроллеры могут как работать в качестве подчиненного при наличии в каркасе соответствующего системного контроллера, так и единолично управлять блоками каркаса. Как ведомое устройство автономный контроллер выполняет ряд команд чтения–записи на

магистрали системы. Это позволяет проводить тестирование узлов автономного контроллера (регистров и буферной памяти) и запись необходимых параметров для его работы в режиме сбора данных.

Для передачи управления автономному контроллеру системный контроллер устанавливает разрешающий сигнал на соответствующей линии системной магистрали. В этом случае автономный контроллер по сигналам, поданным на его переднюю панель, осуществляет инициализацию регистрирующей электроники каркаса, сбор информации события, занесение ее во внутреннюю буферную память, емкость которой позволяет хранить данные, получаемые за цикл ускорителя.

После окончания цикла ускорителя содержимое буферной памяти передается в компьютер по каналу USB. Передаваемый пакет состоит из информационных событий. Каждое информационное событие представляет собой набор 32-х разрядных слов и состоит из заголовка (5 слов) и информационную часть. В заголовке указываются:

– слово 1– разделитель, разряды которого содержат код 0xFFFFFFFF;

– слово 2 – номер события;

– слово 3 – временная метка события 4

– слово 4 – длительность сигнала Busy;

– слово 5 – размер блока (разряды 10–0), код ошибки (разряды 12–11), номер контроллера (разряды 18–14) и номер детектора (разряды 21–19). разряды 31–22– резервные.

Информационное слово содержит адрес сработавшего канала (разряды 30–16) и код данных (разряды 15–0).

После передачи всей информации набранной за цикл работы ускорителя контроллер посылает идентификатор конца набора (код 0xAAAAAAAA).

Архитектура программного обеспечения чтения физических данных, зарегистрированных каркасной электроникой, состоит из трех уровней.

К шине USB автономный контроллер подключается как USB-устройство, осуществляющее чтение данных компьютером.

Первый нижний уровень (транспортный), реализованный в специализированном драйвере ядра Linux (acudrv), осуществляет синхронное и асинхронное чтение данных в режиме DMA посредством циклического буфера драйвера. Размер элемента циклического буфера должен быть кратен размеру буфера конечной точки. Драйвер предоставляет доступ к USB-

устройству посредством специального файла устройства. Файл устройства может быть открыт в каждый момент времени только одним пользовательским процессом. Критические секции в системном вызове, копирующем данные из циклического буфера драйвера, и в нижней половине асинхронного обработчика оповещения, оканчивающего транзакцию приема и запускающего следующую, реализуют доступ к циклическому буферу и переход состояний драйвера.

Второй уровень (буферный), реализованный в библиотеке-компаньоне драйвера (libacu), осуществляет конфигурирование и очистку драйвера, а также обеспечивает считывание данных, накопленных в буферной памяти автономного контроллера за один цикл ускорителя, посредством специального файла устройства. В рамках одного потока выполнения пользователя данные автономных контроллеров могут быть вычитаны синхронно по очереди или асинхронно все сразу в отдельные буферы.

Третий уровень (прикладной) реализует набор данных системой сбора данных физического эксперимента.

При минимально возможном размере элемента циклического буфера в драйвере 512 байт установившаяся скорость приема данных составляет 3.9 МБ/с, при кратном увеличении размера элемента скорость растет кратно, с 2 Кбайт растет на проценты, с 64 Кбайт скорость не меняется. Максимальная устоявшаяся скорость вычитывания данных одного автономного контроллера по шине USB-2 не зависит от его типа, составляет 18 МБ/с и ограничена используемой памятью автономного контроллера. При одновременном асинхронном вычитывании нескольких автономных контроллеров через один корневой хаб USB устоявшаяся суммарная скорость составляет 46 МБ/с.

5. ЗАКЛЮЧЕНИЕ

Интерфейс USB-2 в высокоскоростном режиме с с массовым типом обмена данными продемонстрировал надежную передачу данных и управляющих пакетов между компьютером и разработанными контроллерами при наборе физических данных и тестах, проводимых экспериментальными установками ВЕС, ОКА, ФОДС в течение нескольких лет в ИФВЭ. Прямое кабельное соединение USB длиной до 5 м и опосредованное соединение с промежуточными хабами длиной до 30 м достаточно для подключения каркасной электроники к компьютерам системы сбора данных физического эксперимента. Наличие USB-портов у любой

современной материнской платы делает ненужным приобретение или разработку более дорогих и редких специализированных интерфейсов и кабелей.

К числу решаемых проблем USB-интерфейса можно отнести чувствительность процедуры инициализации USB-устройства (bus enumeration) к наводкам по земле, что заложено в стандарт USB и не может быть исправлено в его рамках. Практическое ограничение на максимальное количество подключенных к корневому хабу USB-2 устройств, равное 16 у некоторых материнских плат вместо допускаемых стандартом 127, может быть снято настройкой BIOS материнской платы и/или ядра Linux. Теоретическое ограничение на максимальную скорость передачи данных 480 Мб/с для USB-2 в высокоскоростном режиме снято в следующих версиях стандарта: USB-3 допускает передачу на скорости до 20 Гбит/с, USB-4 – на скорости до 120 Гбит/с. Активные удлинители USB позволяют присоединять устройства USB-1 / USB-2 / USB-3 к компьютеру на расстоянии до ста метров и более.

СПИСОК ЛИТЕРАТУРЫ

1. *Алферова О.И., Бушнин Ю.Б., Денисенко А.А., и др.* Система унифицированных модулей многоканального анализа “СУММА”. Серпухов, 1974. https://rusneb.ru/catalog/000200_000018_rc_3713550/
2. *Бушнин Ю.Б., Ваньев В.С., Гончаров П.И. и др.* Быстродействующая система регистрирующей и триггерной электроники для экспериментальных исследований в ИФВЭ. Препринт ИФВЭ 88–47. Серпухов: ИФВЭ, 1988.
3. *Букреева С.И., Емельянов Н.М., Исаев А.Н. и др.* ПТЭ. 2014. № 6. С. 23. https://doi.org/10.7868/S003281621405005X
4. *Донсков С.В., Инякин А.В., Карпеков Ю.Д. и др.* ПТЭ. 2016. № 4. С. 38. https://doi.org/10.7868/S0032816216040042
5. Universal Serial Bus Revision 2.0 specification, http://www.usb.org
6. *Ивашин А.В., Матвеев В.Д., Хохлов Ю.А*. Препринт ИФВЭ 2010-10. Протвино: ИФВЭ, 2010. https://elibrary.ru/item.asp?id=19464423
7. CY7C68001 EZ-USB SX2 High-Speed USB Interface Device, Cypress SemiconductorCorporation, Document #:38-08013 Rev.D.
8. Linux kernel, https://www.kernel.org

ПОДПИСИ К РИСУНКАМ

Рис. 1. Блок-схема управления USB. Операции устройства – набор сигналов для управления операциями чтения и записи; команды устройству – шифрованные данные управления от USB.

Рис. 2. Структурная схема узла дешифрации адресов регистров контроллеров канала управления.

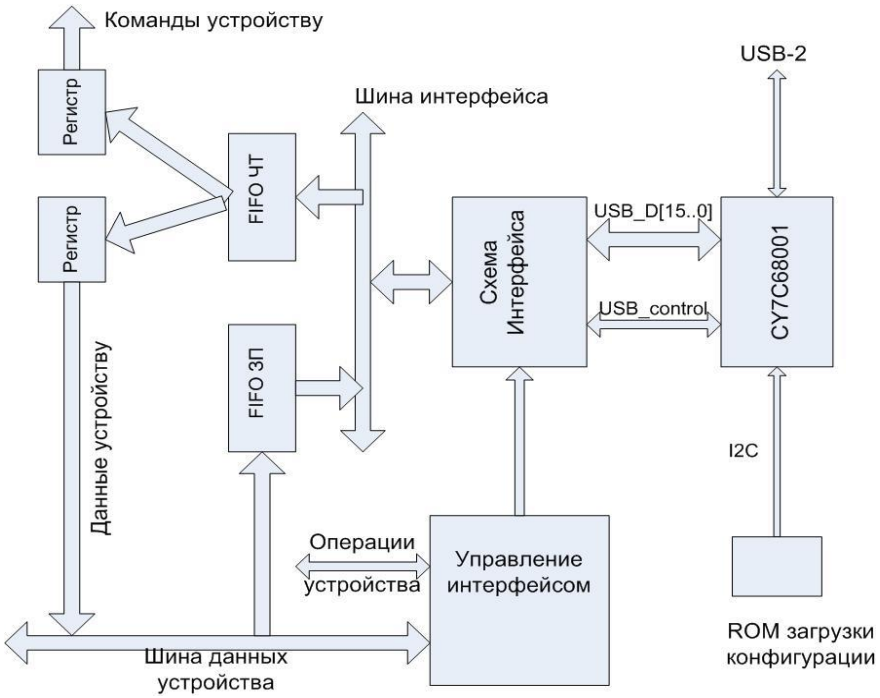

Команды устройству
Регистр
Регистр
Данные устройству
FIFO ЧТ
FIFO ЗП
Шина интерфейса
Схема Интерфейса
USB_D[15..0]
USB_control
USB-2
CY7C68001
I2C
ROM загрузки конфигурации
Операции устройства
Управление интерфейсом
Шина данных устройства

Рис. 1.

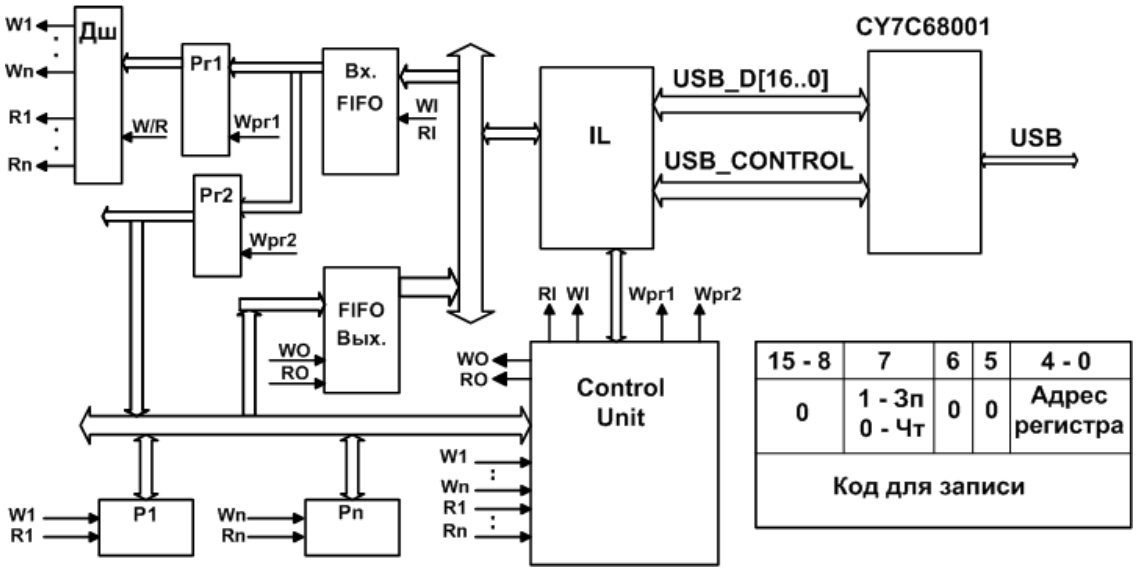

W1
Wn
R1
Rn
Дш
W/R
Pr1
Wpr1
Pr2
Wpr2
Вх.
FIFO
WI
RI
FIFO
Вых.
WO
RO
IL
CY7C68001
USB_D[16..0]
USB_CONTROL
USB
RI
WI
Wpr1
Wpr2
WO
RO
Control
Unit
W1
Wn
R1
Rn
P1
Pn
15 - 8
7
6
5
4 - 0
0
1 - Зп
0 - Чт
0
0
Адрес
регистра
Код для записи

Рис. 2.