



АРХИТЕКТУРА И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ МОСТОВ SPACEWIRE/SPACEFIBRE – I2C

А. Ю. Виноградов, Е. А. Суворова

Санкт-Петербургский государственный университет аэрокосмического приборостроения

В настоящее время существует большое количество оборудования, имеющего интерфейс I2C, которое необходимо использовать в более современных сетях на базе стандарта SpaceWire, SpaceFibre. В данной статье предложена архитектура мостов, обеспечивающих такую возможность, форматы преобразования данных. Приведена оценка достижимой пропускной способности с использованием этих мостов.

Ключевые слова: SpaceWire, I2C, мосты между разными протоколами передачи данных, оценка пропускной способности.

Для цитирования:

Виноградов, А. Ю. Архитектура и структурная организация мостов SpaceWire/SpaceFibre – I2C / А. Ю. Виноградов, Е. А. Суворова // Системный анализ и логистика. – 2026. – № 1(49). – с. 47-56. DOI: 10.31799/2077-5687-2026-1-47-56.

ARCHITECTURE AND STRUCTURAL ORGANIZATION OF SPACEWIRE/SPACEFIBRE – I2C BRIDGES

A. Y. Vinogradov, E. A. Suvorova

St. Petersburg State University of Aerospace Instrumentation

Currently, there is a large amount of equipment with I2C interface that needs to be used in more modern networks based on the SpaceWire and SpaceFibre standards. This article suggests the architecture of bridges that provide this capability, and data conversion formats. An estimate of the achievable throughput using these bridges is given.

Keywords: SpaceWire, I2C, bridges between different data transmission protocols, bandwidth estimation.

For citation:

Vinogradov, A. Y. Architecture and structural organization of SpaceWire/SpaceFibre – I2C bridges / A. Y. Vinogradov, E. A. Suvorova // System analysis and logistics. – 2026. – № 1(49). – p. 47-56. DOI: 10.31799/2077-5687-2026-1-47-56.

Введение

Шина I2C (IC) разработана ещё в начале 1980-х годов, затем она была стандартизирована в 1992 года, после выходили несколько версий добавляющие в стандарт улучшения и новые скорости [1]. За время было выпущено огромное количество датчиков и устройств, работающих со стандартом I2C, так же стандарт лёг в основу IPMB [2] используемый в IPMI (интеллектуальный интерфейс управления платформой).

SpaceWire [3] и SpaceFibre [4] наоборот являются современными и перспективными протоколами, которые ещё не успели обзавестись большим количеством устройств. Мост SpaceWire/SpaceFibre — I2C позволит подключить всё разнообразие устройств, работающих с I2C, в современные стандарты связи. Наряду с этим он может позволить устройствам I2C расположенным на большом расстоянии друг от друга, взаимодействовать через «прозрачную» для них сеть SpaceWire. В данной статье предлагается архитектура мостов SpaceWire -I2C, выполняется оценка достижимой пропускной способности с использованием этих мостов.

Краткая информация о стандартах

I2C. I2C – (IC – Inter-Integrated Circuit) последовательная шина предназначенная для связи между устройствами внутри электронных приборов и интегральными схемами. Шина имеет ведущие и ведомые устройства, а физически используются 2 проводника, подключенных в схему «монтажное И». Один проводник используется для тактирования (SCL), а второй для полезных данных (SDA). [5]



Ведущие устройства генерируют сигнал SCL, и, в зависимости от направления, SDA. Ведомые устройства принимают SCL и могут его «прижать к земле» если необходимо приостановить приём данных по линии SDA.

Изначально протокол имеет адресное пространство 7 бит, однако существует расширение до 10 бит. Стандартом определено 6 режимов работы [6]:

1. Стандартный режим (Sm) 100 Кбит/с
2. Быстрый режим (Fm) 400 Кбит/с
3. Быстрый режим плюс (Fm+) 1 Мбит/с
4. Высокоскоростной режим (Hs) 1.7 Мбит/с
5. Высокоскоростной режим (Hs) 3.4 Мбит/с
6. Сверхбыстрый режим (Ufm) 5 Мбит/с

Режим (Sm) является режимом по умолчанию, и на нём должны уметь работать все устройства. Количество узлов в сети ограничено адресным пространством и общей ёмкостью шины 400 pF. Используется топология общая шины и в большинстве случаев с одним «мастер» устройством.

На рисунке 1 показан формат пакета при работе в режиме адреса 7 бит. Сначала идёт сигнал Start, затем 7 бит адреса, затем 1 бит режима (чтение, запись), затем ожидается бит (A - Ask) подтверждения от другой стороны, затем идёт байты полезных данных. Каждый байт подтверждается сигналом Ask (A). В конце пакета ставится сигнал Stop (P). Из всех 127 адресов, есть 2 группы которые зарезервированы и не могут использоваться (0000XXX и 1111XXX).



Рис. 1. Формат пакета I2C при работе в режиме адреса 7 бит

А на рисунке 2 показан формат пакета при адресации 10 бит. Различие в том, что часть адреса приходит в первом байте (XX), число 11110 — означает, что адрес 10 бит, и вторая часть адреса во втором байте (YYYYYYYY), остальные поля остаются без изменений.

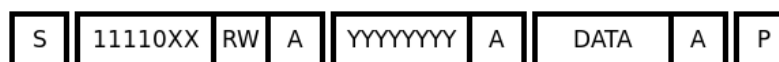


Рис. 2. Формат пакета I2C при работе в режиме адреса 10 бит

В шине I2C может быть несколько мастер-устройств, в этом случае арбитраж происходит следующим образом:

Оба мастера побитово выставляют адрес и данные, и тот чей выставленный адрес не совпадает со считанным, останавливает свою передачу. Это реализуется благодаря использованию на физическом уровне логического «и», когда линия к «1» подтягивается сама через сопротивление, а вот к «0» её тянет драйвер передатчика.

SpaceWire.

SpaceWire – последовательный протокол, использующий 4 дифференциальные пары (LVDS, по 2 пары в каждом направлении [7]) работающий на скоростях от 5 до 400 Мбит/с. Все соединения имеют топологию точка-точка, и от передатчика к приёмнику идёт 2 дифференциальные пары (DS), по одной из них передаются данные, а по второй — строб, позволяющий на приёмной стороне восстановить синхросигнал.

Пакет неограниченного размера состоит из адреса, полезных данных и символа окончания пакета (EOP — нормальное окончание или EEP — ошибочное окончание пакета).

Кроме пакетов могут передаваться Time-code используемые для синхронизации времени.



Протокол имеет систему кредитования, позволяющий контролировать заполнение приёмного буфера. Для этого приёмная сторона отправляет символы кредитования (FCT) для передающей стороны, таким образом сообщая, для какого количества данных имеется свободное место.

SpaceFibre.

SpaceFibre – так же является последовательным протоколом точка-точка и логическим продолжением развития стандарта SpaceWire.

На физическом уровне вместо DS кодирования используется 8b10b кодирование с синхронизацией с помощью символов Comma. Количество дифференциальных пар уменьшилось с 4х до 2х, а скорость выросла до 2,5 Гбит/с.

Протокол SpaceFibre вводит понятие — виртуальный канал [8], являющийся по сути каналом SpaceWire и имеющий внутри себя всё тот же формат пакета — адрес, данные и символ окончания пакета (EOP, EEP).

Архитектура мостов

Реализация моста возможна 3 разными способами, каждый из которых может быть полезен в определённых случаях, например: устройства SpW или I2C не знают о существовании друг друга, в этом случае будут полезны режимы: перенос пакетов SpW через I2C и перенос пакетов I2C через сеть SpW.

В случае, когда требуется именно объединение сети, то можно использовать мост в режиме почти прозрачного моста SpW-I2C.

Фактически может использоваться один и тот же мост SpW-I2C с разными настройками

Режим моста – почти прозрачный из SpW в I2C. В данном режиме заголовок пакета SpW определяет режим работы и адресата на шине i2c, затем происходит перенос основных данных. Для передачи в обратную сторону, адреса I2C устройств транслируются обратно в адресе SpW. Для этого должна быть описана таблица трансляции адресов, которая предварительно заполняется при конфигурировании обеих сетей. Поскольку обе сети имеют возможность управлять потоком, то не понадобится делать больших буферов, и можно использовать червячную маршрутизацию SpW при переходе данных через такой мост.

Дополнительно, мост может управлять одновременно несколькими шинами i2c, поскольку скорость SpW/SpFi во много раз превышает скорость шины i2c, однако в этом режиме может быть проблема в использовании червячной маршрутизации, если устройство на шине i2c приостановит приём данных. Решением данной проблемы может быть промежуточная буферизация данных, чтобы освободить сеть SpW/SpFi и можно было бы обратиться к другому устройству i2c на соседней шине.

Итак, передачу данных в устройство или из устройства i2c пошагово можно представить так:

1. По SpW приходит первый байт, в котором определяется режим работы (restart или start), а также режим адреса (7 или 10 бит) и три старших бита при 10-ти битном режиме.
2. Адрес i2c устройства и режим (чтение или запись)
3. На шине i2c формируется сигнал Start и без изменений передаётся адрес и режим.
4. От подчинённого устройства I2C ожидается сигнал подтверждения.
5. По сети SpW приходит следующий символ данных.
6. Следующий символ данных передаётся без изменений в шину i2c.
7. От подчинённого устройства ожидается подтверждение.
8. Если по сети SpW приходит символ EOP и режим работы был start то на шине i2c формируется символ STOP, означающий окончание передачи.

Основная часть архитектуры этого варианта моста представлена на рисунке 3.

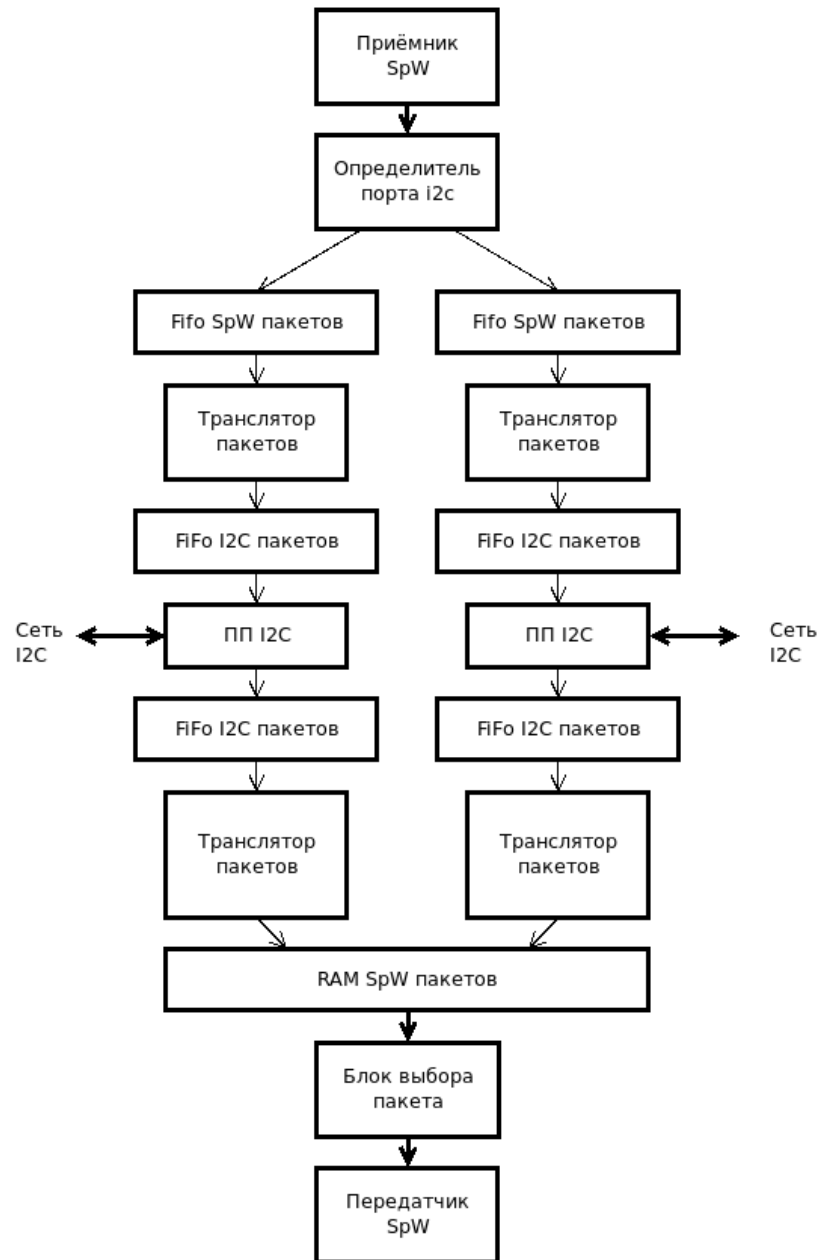


Рис. 3. Основная часть алгоритма работы моста

Архитектура такого моста может масштабироваться для увеличения количества портов i2c. Для каждого порта имеется свой буфер для хранения ещё не распакованных SpW пакетов, транслятор пакетов, буфер хранения I2C пакетов, в котором может производиться переход домена тактирования, затем непосредственно сам порт I2C, принятые пакеты I2C попадают в свой буфер, где происходит обратный переход домена, заканчивается цепочка — транслятором, который переводит принятые пакеты i2c в пакеты SpW и складывает их в общую память.

Вместо приёмника и передатчика SpaceWire можно поставить виртуальный канал SpaceFibre, дополнительно можно разграничить i2c порты по виртуальным каналам.

В таких реализациях как IPMI [9] размер i2c пакета ограничен 256 байтами, поэтому такой мост потребует, как минимум, использование буферов на 256 байт, однако в современных FPGA имеется большой объём блочной памяти [10] поэтому этот размер можно увеличить до объёма одного модуля блочной памяти.

Режим моста — перенос пакетов SpW. Пакет SpW полностью и без изменений



передаётся одному из адресатов сети I2C. Этого адресата необходимо заранее настроить. (Предполагается, что адресат располагает информацией о формате пакета SpW.)

Режим моста – перенос пакетов i2c. Пакет I2C полностью и без изменений передаётся одному из адресатов сети SpW. Для этого перед пакетом добавляется путь адрес сети SarceWire, а после пакета добавляется символ EOP. (Предполагается, что адресат располагает информацией о формате пакетов I2C.)

Описание форматов пакетов

На рисунке 4 показано как соотносятся содержимое пакета SpaceWire (сверху) и пакета i2c (снизу) при 7 битной адресации и без повторного старта. Поле данных пакета SpaceWire может иметь произвольную длину. В простейшем случае Header SpaceWire представляет собой сетевой заголовок пакета SpaceWire. Потенциально могут использоваться различные транспортные форматы передачи пакета, например, RMAP. В этом случае в пакете наряду с сетевым может присутствовать транспортный заголовок. В этом случае перед EOP может присутствовать концевик транспортного формата.

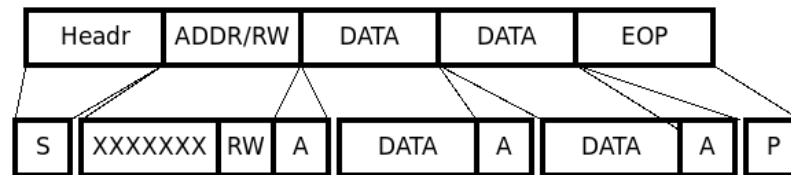


Рис. 4. Схема преобразования SpaceWire → I2C

На рисунке 5 представлена схема преобразования I2C → SpW. Перед полем headr содержится сетевой заголовок пакета SpW, а в самом заголовке Headr содержится информация о повторном старте (pSTART), флаг ACK, флаг NACK, режим 10 бит и 3 старших бита адреса, для 10-битного режима. (В этом случае также могут использоваться различные форматы транспортного уровня.) При 10 битном режиме без повторного старта, при сборке первого адреса шины i2c ставится константа режима 10 бит (11110), затем из HEADR извлекаются 2 бита адреса. Бит RW находится во втором байте пакета SpW, затем, при формировании второго байта адреса i2c, один бит берётся из headr и 7 из 2-го байта пакета SpW. Затем идут данные без изменений и по EOP выставляется STOP на i2c.

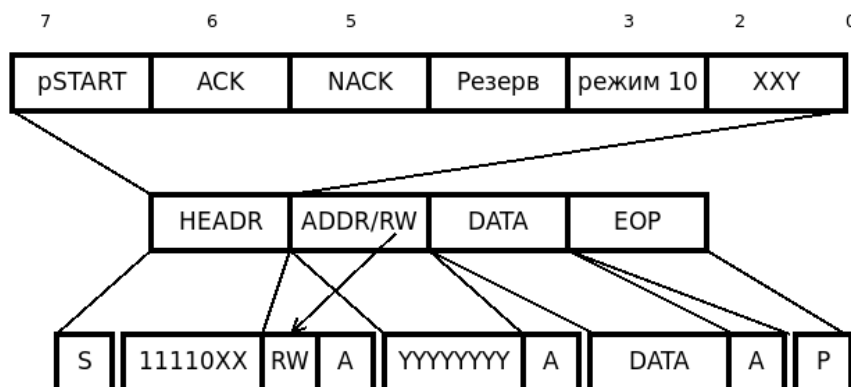


Рис. 5. Схема преобразования I2C → SpW (повторный старт)

Режим, когда на шине i2c появляется второй старт показан на рисунке 6. Как видно, при установленном флаге pSTART в поле HEADR приходящий EOP символ не вызывает отправку сигнала STOP на шине i2c.

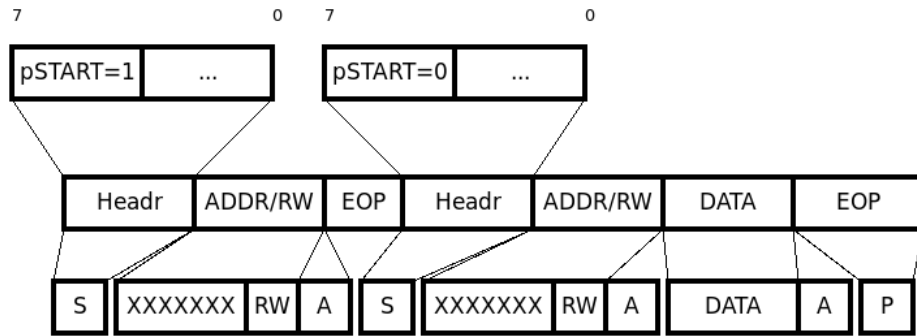


Рис. 6. Схема преобразования I2C-> SpW (несколько Stop/Start)

Взаимодействие двух устройств I2C через сеть SpaceWire

Рассмотрим ситуацию, когда мост i2c-SpW/SpFi объединяет 2 сети i2c. В этом случае мост будет прозрачным для сети i2c. На рисунке 7 показана небольшая сеть, где есть I2C мастер, который пытается запросить 2 байта у I2C Slave (например, EEPROM), к которому он подключен через сеть SpaceWire/SpaceFibre.

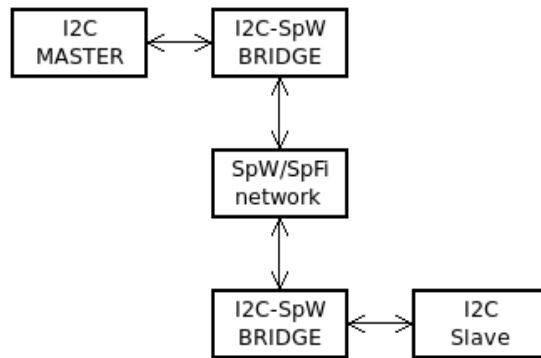


Рис. 7. Схема взаимодействия устройств I2C через сеть SpW

Последовательность передачи пакетов будет примерно такая:

1. I2C Master формирует адрес с флагом записи и одним байтом данных (для EEPROM памяти I2C это адрес откуда будут читаться данные), затем идёт повторный старт.
2. Первый мост I2C-SpW/SpFi принимает данные от I2C Master и видя там повторный старт, приостанавливает приём данных по I2C, удерживая SCL в низком уровне.
3. Первый мост формирует первый пакет с адресом SpW до второго моста I2C-SpW/SpFi. Так же в пакет входит заголовок, с установленным битом pSTART, адресом I2C устройства и 1 байтом данных (в будущем это адрес ячейки памяти EEPROM). Окончание — EOP.
4. Первый мост I2C-SpW/SpFi снимает удержание сигнала SCL.
5. I2C Master снова формирует адрес I2C slave с установленным битом чтения.
6. Первый мост принимает пакет от I2C Master и видя там флаг READ формирует второй SpW пакет с адресом SpW, адресом I2C slave и символом EOP. Снова удерживает шину сигналом SCL.
7. Первый пакет добирается до второго моста I2C-SpW/SpFi. Мост читает заголовок, формирует сигнал START на шине I2C, передаёт адрес устройства I2C, и первый байт данных (адрес ячейки EEPROM) и видя в заголовке флаг pSTART, НЕ выставляет сигнал stop при приёме EOP.
8. Устройство I2C-slave видя свой адрес, выставляет ack, затем принимает один байт данных и снова выставляет ACK.



9. Второй пакет добирается до второго моста I2C-SpW/SpFi. Мост извлекает заголовок, выставляет сигнал START на шину I2C и видя флаг READ передаёт адрес на шину I2C и переходит в режим чтения.
10. Устройство I2C-Slave снова видит свой адрес, но уже с флагом READ, подаёт ACK, а затем байт, прочитанный по адресу переданному в прошлой транзакции.
11. Второй мост принимает байт и формирует ACK пакет для первого моста содержащий адрес slave устройства, первый принятый байт, EOP и удерживает шину I2C сигналом SCL на «паузе».
12. Первый мост принимает пакет, в котором установлен флаг ACK и адрес slave устройства.
13. Первый мост снимает удержание SCL и передаёт мастеру первый байт данных, содержащийся в пакете от второго моста.
14. I2C-Master отвечает первому мосту ACK.
15. Первый мост формирует пакет SpW с SoW адресом второго моста, адресом I2C-Slave устройства, флагом ACK и с EOP.
16. Второй мост принимает ACK пакет с адресом I2C-Slave.
17. Второй мост снимает удержание SCL передаёт I2C-slave ACK и переходит в режим приёма следующего байта.
18. Второй мост принимает байт и формирует ACK пакет для первого моста содержащий адрес slave устройства, второй принятый байт, EOP и удерживает шину I2C сигналом SCL на «паузе».
19. Первый мост принимает пакет, в котором установлен флаг ACK и адрес slave устройства.
20. Первый мост снимает удержание SCL и передаёт мастеру второй байт данных, содержащийся в пакете от второго моста.
21. I2C-Master отвечает первому мосту NACK и формирует сигнал STOP.
22. Первый мост формирует пакет SpW с SpW адресом второго моста, адресом I2C-Slave устройства, флагом NACK и с EOP.
23. Второй мост принимает NACK пакет с адресом I2C-Slave.
24. Второй мост отпускает сигнал SCL, передаёт I2C slave символ NACK и формирует сигнал STOP.

Таким образом, для чтения из удалённого устройства i2c двух байт данных понадобится передать от первого ко второму мосту 2 пакета данных, 1 пакет ACK и 1 пакет NACK (всего 4 пакета). От второго моста к первому будет передано 2 пакета ACK с 2 байтами данных от I2C устройства.

Оценим пропускную способность для этого случая. Будем считать, что I2C работает на стандартной скорости 100 Кбит/с, а в SpaceWire не происходит никаких блокировок и работает на штатной скорости установки соединения — 10 Мбит/с. На рисунке 8 показана диаграмма обмена между устройствами с указанием пакетов, где:

i2c-p0 – пакет с адресом на запись, его длина $1+7+1+8=17$.

i2c-p1 – ответный ack = 1

i2c-p2 — пакет с запросом чтения и повторным start = $1+7 + 1=9$

i2c-p3 — данные от памяти $8 + 1=9$

i2c-p4 — ack = 1

i2c-p5 — данные от памяти $8 + 1=9$

i2c-p6 — nack + stop = 2

spw-p0 — 2 data + headr + eop = $20 + 10 + 4 = 34$

spw-p1 — data + headr + eop = $10 + 10 + 4 = 24$

spw-p2 — data + headr + eop = $10 + 10 + 4 = 24$

spw-p3 — headr + eop = $10 + 4 = 14$



$spw-p4 \text{ — } headr + data + eop = 10 + 10 + 4 = 24$
 $spw-p5 \text{ — } headr + eop = 10 + 4 = 14$
 $i2c-p8 = i2c-p0 = 17$
 $i2c-p9 = i2c-p2 = 9$
 $i2c-p10 = i2c-p3 = 9$
 $i2c-p11 = i2c-p4 = 1$
 $i2c-p12 = i2c-p5 = 4$
 $i2c-p13 = i2c-p6 = 2$

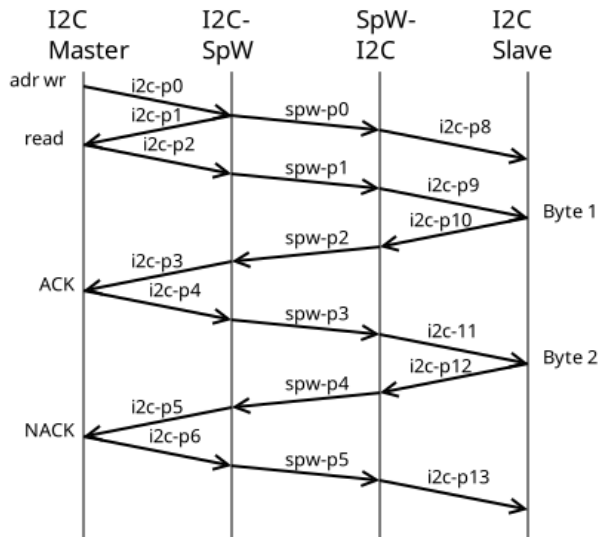


Рис. 8. Иллюстрация последовательности выполняемых действий

Время $T_{i2c-spw}$ передачи между I2C-master и первым мостом вычисляется по следующей формуле:

$$T_{i2c-spw} = \frac{1}{V_{i2c}} * \left(\sum_{i=0}^6 (i2c - pi) \right) \quad (1)$$

Где V_{i2c} – частота передачи по I2C

Если $V_{i2c} = 100000$ (10Кбит/с), то $T_{i2c-spw} = 0,00048$

Время передачи $T_{SpW-SpW}$ между мостами SpW вычисляется по следующей формуле:

$$T_{SpW-SpW} = \frac{1}{V_{SpW}} * \left(\sum_{i=0}^5 (spw - pi) \right) \quad (2)$$

Где V_{spw} – скорость передачи по SpW,

Если $V_{spw} = 10000000$ (10Мбит/с), то $T_{SpW-SpW} = 0,0000134$

И оставшееся время $T_{SpW-i2c}$ вычисляется по формуле:

$$T_{SpW-i2c} = \frac{1}{V_{i2c}} * \left(\sum_{i=8}^{13} (i2c - pi) \right) \quad (3)$$

Если $V_{i2c} = 100000$ (10Кбит/с), то $T_{i2c-spw} = 0,00042$

Всего было передано байт $S_{byte} = S_{i2c_adr} + S_{i2c_data} + S_{i2c_adr} + S_{i2c_data} + S_{i2c_data} = 5$ байт.

Получается для этого случая скорость передачи данных составляет



$$U = S_{byte} / (T_{i2c-SpW} + T_{spw-spw} + T_{spw-i2c}) \quad (4)$$

$U = (5 * 8) / (0,0009134) \sim 43792$ бит в секунду или около 43,8 кбит/с

Если сравнить с со скоростью, если бы абоненты были в одной сети, то скорость будет следующая:

$$U = S_{byte} / T_{i2c-SpW} \quad (5)$$

$U = (5 * 8) / 0,00048 \sim 83333$ бит/с или около 83,3 Кбит/с

Таким образом, 2 моста с 2 сетями i2c вносят почти двухкратное (1,9) снижение пропускной способности.

На графике показано, как изменится скорость передачи по протоколу SpW-I2C в зависимости от скорости канала SpW

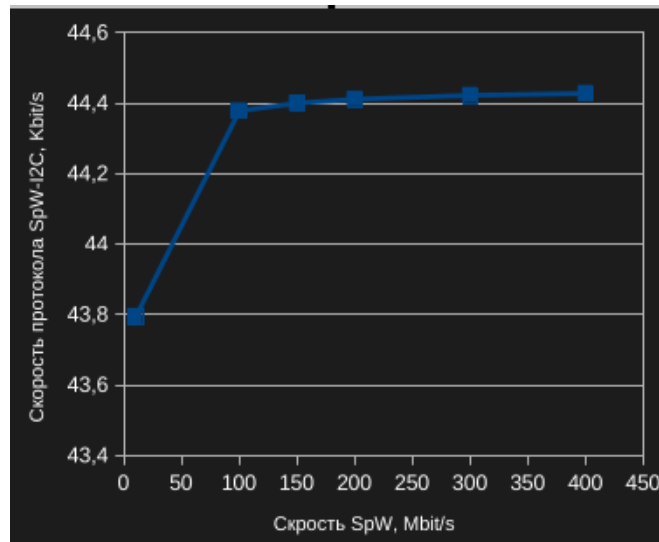


Рис. 9. График зависимости времени обмена данными от скорости передачи по SpaceWire

Заключение

В статье обоснована необходимость разработки мостом SpaceWire – I2C, предложена архитектура таких мостов, форматы преобразования данных при передаче SpW-i2c и I2C - SpW. Выполнены оценки достижимой пропускной способности при использовании мостов. Показано, что сеть SpaceWire может выступать в качестве прозрачной среду передачи данных между устройствами I2C.

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации, соглашение № FSRF-2023-0003, "Фундаментальные основы построения помехозащищенных систем космической и спутниковой связи, относительной навигации, технического зрения и аэрокосмического мониторинга".

СПИСОК ЛИТЕРАТУРЫ

1. The I2C-BUS Specification. V 2.1 // Hamilton: Philips Semiconductors. – 2000. – 46 p.
2. Кураедов В. И. Разработка и исследование I2C / SMBus VIP-блока // Журнал «Евразийский Союз Учёных». – 2020. – Р. 37-40.



3. ГОСТ Р 70020—2022. Космическая техника. Интерфейсы и протоколы высокоскоростного межприборного информационного обмена и комплексирования систем космических аппаратов. SPACEWIRE-RUS // Москва: Российский институт стандартизации. – 2022. – 118 с.
4. SpaceFibre – Very high-speed serial link. ECSS-E-ST-50-11C. – Noordwijk: ESA-ESTEC, 2019. – 233 p.
5. Семенов Б.Ю. Шина I2c в радиотехнических конструкциях. – М.: Солон-Р, 2002. – 190 с.
6. I2C-bus specification and user manual Rev. 6. – Eindhoven: NXP Semiconductors N.V. – 2014. – 64 p.
7. ECSS Standard ECSS-E-ST-50-12C. SpaceWire, Links, Nodes, Routers and Networks. Issue 1. – Noordwijk: European Cooperation for Space Data Standardization. – 2008.– 129 p.
8. Суворова Е. А. Анализ технологии SpaceFibre для высокоскоростных бортовых сетей / Е. А. Суворова, В. Е. Степанов, В. Л. Оленев // Космические аппараты и технологии. – 2023 – Р. 100-106.
9. Intelligent Platform Management Interface Specification Second Generation, V 2.0. Addenda, Errata, and Clarifications document. – Portland: Intel, Hewlett-Packard, NEC, and Dell Computer Co. – 2004. – 34 p.
10. Advanced Micro Devices (AMD). 7 Series FPGAs: Overview (DS180) [Электронный ресурс]. – URL: https://docs.amd.com/v/u/en-US/ds180_7Series_Overview (дата обращения: 29.01.2026).

ИНФОРМАЦИЯ ОБ АВТОРАХ

Виноградов Алексей Юрьевич

Инженер

Санкт-Петербургский государственный университет аэрокосмического приборостроения
Россия, 190000, Санкт-Петербург, ул. Большая Морская, д.67, лит. А
E-mail: alexey.vinogradov@guap.ru

Суворова Елена Александровна

Канд. техн. наук, доцент

Санкт-Петербургский государственный университет аэрокосмического приборостроения
Россия, 190000, Санкт-Петербург, ул. Большая Морская, д.67, лит. А
E-mail: wildcat15@yandex.ru

INFORMATION ABOUT THE AUTHORS

Vinogradov Alexey Yurievich

Engineer

Saint-Petersburg State University of Aerospace Instrumentation
67, Bolshaya Morskaya str., Saint-Petersburg, 190000, Russia
E-mail: alexey.vinogradov@guap.ru

Suvorova Elena Alexandrovna

PhD. tech. Sciences, associate Professor

Saint-Petersburg State University of Aerospace Instrumentation
67, Bolshaya Morskaya str., Saint-Petersburg, 190000, Russia
E-mail: wildcat15@yandex.ru

Дата поступления: 09.12.2025

Дата принятия: 20.01.2026