

**Инструментальные
системы**

**Протокол обмена данными
PROTEQ (V1.0 и V2.0)**

Внутренний стандарт
DCR1206

февраль, 2014

Москва

Составители: Смехов Д.Г., Петров В.И.

Редакция: 14.02.2014

Индекс: dcr1206

ЗАО «ИнСис»

Юридический адрес: 109004, г. Москва, Тетеринский пер., д.16

Фактический адрес: 113405, г. Москва, Варшавское шоссе, д.125,
строение 1, секция 2, офис 2106

Почтовый адрес: 117587, г. Москва, а/я 162

Телефон: (495) 781-2750, 781-2751, 781-2752, местн. 25-88

Факс: (495) 781-2751

E-mail: info@insys.ru, mail@insys.ru

WWW: <http://www.insys.ru>

Оглавление

1. ВВЕДЕНИЕ	4
1.1 НАЗНАЧЕНИЕ ПРОТОКОЛА.....	4
1.2 ОГРАНИЧЕНИЯ ПРИМЕНЕНИЯ.....	5
2. ФИЗИЧЕСКИЙ УРОВЕНЬ	5
2.1 ФУНКЦИИ ФИЗИЧЕСКОГО УРОВНЯ.....	5
2.1.1 <i>Порядок следования бит</i>	7
2.1.2 <i>Подключение аппаратуры физического уровня</i>	7
2.2 СТАНДАРТНАЯ РЕАЛИЗАЦИЯ ЗАО «ИНСИС».....	9
3. КАНАЛЬНЫЙ УРОВЕНЬ	10
3.1 СЛУЖЕБНЫЙ ПАКЕТ.....	11
3.2 ПАКЕТ С ДАННЫМИ.....	11
3.3 СЛОВО КОМАНДЫ.....	11
3.3.1 <i>SIG</i>	12
3.3.2 <i>RX_ENABLE</i>	12
3.3.3 <i>BUF_NUM</i>	12
3.3.4 <i>PKG_TOGLE</i>	12
3.3.5 <i>ACK_TGLx</i>	12
3.4 ОЧЕРЕДЬ ПАКЕТОВ PROTEQ V1.0.....	13
3.5 ОЧЕРЕДЬ ПАКЕТОВ PROTEQ V2.0.....	13
3.6 ПОДТВЕРЖДЕНИЕ ПАКЕТОВ.....	14
3.7 ИНТЕРФЕЙС АППАРАТУРЫ КАНАЛЬНОГО УРОВНЯ.....	14
3.8 РЕАЛИЗАЦИЯ ЗАО «ИНСИС».....	15
4. СЕТЕВОЙ УРОВЕНЬ	16
4.1 ОБЩИЕ ПРИНЦИПЫ И ПРАВИЛА.....	17
4.2 СОЕДИНЕНИЕ ТОЧКА-ТОЧКА.....	18
4.3 СОЕДИНЕНИЕ ТОЧКА-МНОГОТОЧИЕ.....	18
5. ПРИКЛАДНОЙ УРОВЕНЬ	20

1. ВВЕДЕНИЕ

Протокол потокового обмена данными PROTEQ или просто PROTEQ был предложен в 2012 году ведущим специалистом ЗАО «ИнСис» Дмитрием Смеховым в качестве простого, быстрого и надежного средства для организации обмена между двумя устройствами по высокоскоростным двунаправленным последовательным линиям передач. Применение протокола PROTEQ дает альтернативное решение в задачах потокового обмена для таких распространенных интерфейсов как PCI Express, RapidIO, Aurora, имеет перед ними скоростные преимущества при равных прочих условиях, обеспечивает высокую надежность и оперативность, а самое главное протокольную простоту. Протокол PROTEQ пришел на смену протоколу FOTR, который использовался в изделиях ЗАО «ИнСис» в предшествующий период.

Протокол PROTEQ имеет ряд существенных особенностей реализации:

1. Применение кодирования 64b/67b;
2. Применение скремблирования;
3. Использование нескольких линий (Lane) высокоскоростной передачи;
4. Фиксированный размер полезной нагрузки 1024 байта в пакетах с заголовком всего 8 байт (260 слов по 32xN разряда);
5. Использование очереди из четырех последовательных пакетов приема и передачи;
6. Подтверждение во встречных пакетах.

Стандарт протокола PROTEQ укладывается в стандартную связную модель ISO-OSI и описывается четырьмя ее уровнями:

1. Физическом (Physical);
2. Канальным (Data Link);
3. Сетевым (Network).

В настоящий момент имеются две версии протокола, отличающиеся Канальным уровнем.

Далее в этом документе протокол PROTEQ будет рассмотрен по уровням и приведены примеры его реализации.

1.1 Назначение протокола

Протокол PROTEQ предназначен для надежной двунаправленной потоковой передачи данных с одного устройства на другой по последовательным скоростным интерфейсам. Протокол ориентирован на применение в сетях с топологией точка-точка и точка-многоточие с некоммутируемым трафиком. Надежность передачи поддерживается механизмами группового подтверждения и оперативного повтора.

Протокол полезен, когда надо получить высокую пропускную способность, близкую к максимальной.

1.2 Ограничения применения

Протокол PROTEQ позволяет получить высокую пропускную способность близкую к максимальной, но не более того. Иначе говоря, скорость не может совпадать с пропускной способностью, поскольку при проектировании узлов PROTEQ следует учитывать влияние вероятных ошибок. Там, где плотность ошибок в канале превысит допустимое значение, могут возникнуть потери проскальзывания. Например, в предельном случае при передаче данных со скоростью, равной максимальной, любое искажение пакета будет приводить к невозвратному заполнению буферов системы, что в конечном счете приведет к их переполнению. Несмотря на это, протокол PROTEQ имеет по показателю максимальной скорости очень высокие значения.

Протокол жестко привязан к физическому уровню с кодировкой 64b/67b (или 64b/66b) со структурой пакетов, где полезная нагрузка и заголовки имеют разную физическую природу. Это ограничивает применение протокола без доработки в сетях с физикой другой природы. Напротив, система на основе PROTEQ может легко быть использована в качестве транспорта для других протоколов.

Для встраивания пакетов в полезную нагрузку пакетов других стандартов, структуру пакетов PROTEQ придется доработать, но и в этом случае существуют ограничения, на первый взгляд отсутствующие. Особо следует сказать о нецелесообразности передачи пакетов PROTEQ в сетях Internet, применяя для этого пакеты и протоколы TCP или UDP. Протокол TCP сам является средством надежной доставки, – и передавать в качестве полезной нагрузки пакеты другого надежного протокола бессмысленно. Протокол UDP мог бы быть применен в качестве транспорта PROTEQ, но только при подключении точка-точка, где перестановка пакетов отсутствует. В глобальных сетях пакеты могут приходиться с опозданием и нарушением порядка. Протокол PROTEQ имеет механизм подтверждения, чувствительный к таким перестановкам.

2. ФИЗИЧЕСКИЙ УРОВЕНЬ

Физический уровень PROTEQ базируется на аппаратуре линий высокоскоростной передачи, кодирования и проверки, которые объединяются в аппаратный узел. Узел имеет абсолютно независимые каналы передачи и приема. Данные могут передаваться только между двумя идентичными узлами двух аналогичных устройств. Физический уровень PROTEQ не ориентирован на работу в сетях сложной топологии и требует подключения только точка-точка.

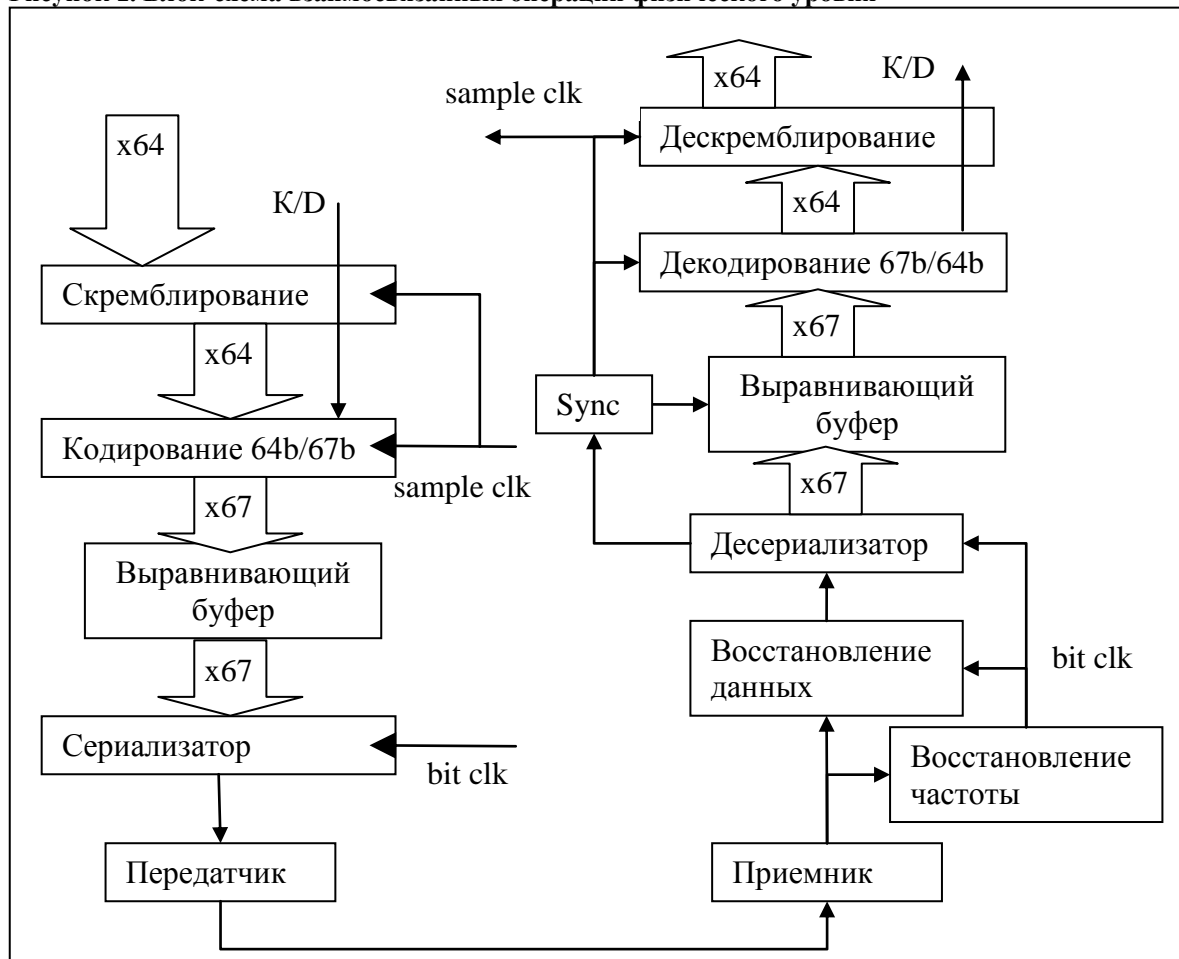
Простейший узел содержит одну пару линий, включающую дифференциальную линию передачи и дифференциальную линию приема. Более сложный узел содержит две таких пары. В общем случае, таких пар в узле может быть больше и их число ничем не ограничено. Все передающие линии должны работать на одной скорости. Аналогично, все принимаемые линии тоже должны работать на одной скорости. В общем случае скорости передающей и принимающей стороны могут отличаться, но, как правило, они равны.

2.1 Функции физического уровня

Каждая пара связных линий подключена к собственной аппаратуре физического уровня. Аппаратура такой пары работает независимо от остальных, но синхронно с ними.

Во время прохождения данных на физическом уровне выполняется последовательные операции, показанные на Рис.1. Приведен вариант, который может отличаться от действительного деталями.

Рисунок 1. Блок-схема взаимосвязанных операций физического уровня



Скремблирование – это операция полиномиальной свертки, обеспечивающая преобразование входного битового потока к форме с наименьшей спектральной плотностью последовательной передаче битов. Дескремблирование – это обратный процесс. Данные на скремблер поступают с уровня Data Link непрерывно в виде потока данных и управляющих символов. Скремблирование полиномиальное мультипликативное длиной 58 бит и выполняется по формуле $X^{58} + X^{39} + 1$. В начале работы скремблеры сбрасывается одинаково на передающем и приемном конце, что в общем случае не является обязательным. Поскольку скремблер самосинхронизирующийся, то его дальнейшая синхронизация не требуется. Восстановление синхронизации гарантировано происходит за 58 переданных битов после возникновения на линии одиночной ошибки.

Кодирование обеспечивает избыточное кодирование путем отображения 64 битных символов и 64 битных управляющих символов на множество чисел разрядностью 67 бит путем добавления к символам трех бит. Декодирование обратный процесс. Кодирование гарантирует на приемном конце достоверное раскодирование с восстановлением по-символьной синхронизации и тактовой частоты.

Сериализация – процесс передачи 67-битных символов последовательно друг за другом. Десериализация – обратный процесс. Биты сериализованных данных через буфера передатчика равномерно по времени передаются до входных буферов приемника, где восстанавливаются. В качестве среды передачи может использоваться дифференциальная медная пара или оптический кабель.

На приемном конце, восстановление битовой частоты производится из входного потока по транзакциям сигнала, которое гарантировано на каждые 67 бит в трех добавленных битах кодирования. Восстановление символьной частоты осуществляется в блоке **Sync**, назначение которого подстройка и удержания синхронизации по тем же трем битам.

Увеличение числа битов потока, требуемого для приема без «проскальзывания», должно быть учтено при передаче. Для этого поступающий на вход уровня поток должен идти не непрерывно, а с регулярной задержкой. Такая задержка должна учитывать увеличения числа бит при кодировании. И передающая, и принимающая аппаратура содержат выравнивающие буфера (Elastic Buffer).

2.1.1 Порядок следования бит

Порядок следования бит должен быть одинаков во всех узлах физического уровня. Если не оговорено особо, то выбирается порядок «младший первый». Обмен данными физического уровня с остальными уровнями стека протокола также подчинены этому правилу. Первыми следуют младшими независимо от того, используются для этого слова, байты, символы или что-либо иное.

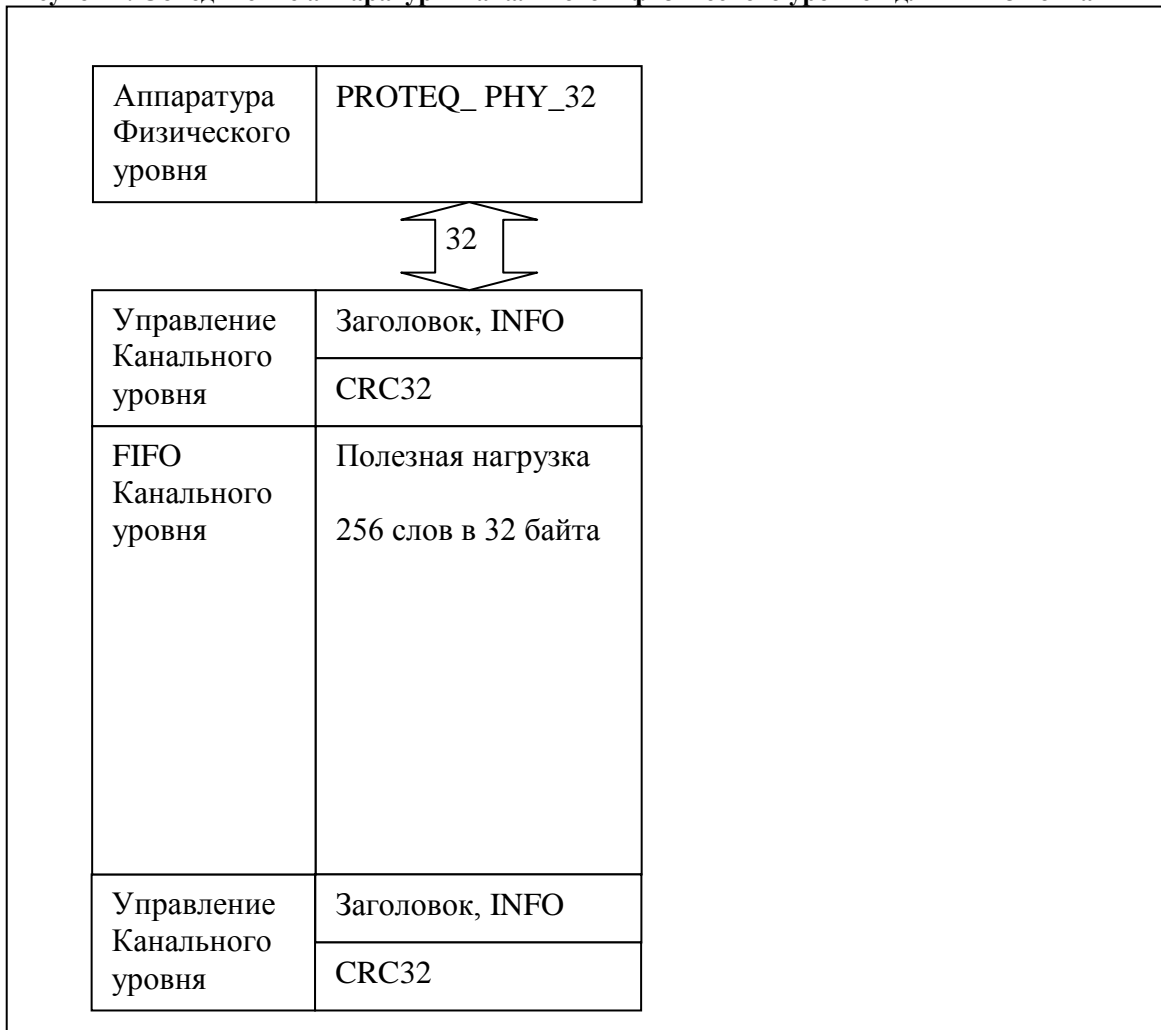
2.1.2 Подключение аппаратуры физического уровня

Пакеты физического уровня протокола PROTEQ задаются структурой следования символов полезной нагрузки и управляющих символов. Данные полезной нагрузки могут поступать на физический уровень и сниматься с него словами разной ширины, но эта ширина должна быть кратна 32-х битному слову, а величина кратности зависит от битовой скорости передатчика MGT.

Так на скорости до 10 Гбит/сек можно использовать слова шириной 32 бита (половина символа кодировки 64b/67b). Для скорости от 10 Гбит/сек и выше целесообразно использовать ширину шины 64 бита (один символ). В дальнейшем, с увеличением скорости, возможен будет переход к шине 128 бит. На первый взгляд создается впечатление, что с наращиванием ширины растет сложность ряда вычислений, таких как скремблирование и расчет контрольной суммы. Со скремблированием это действительно так, поскольку процедура определена для потока символов и требует сканирования по ширине. С контрольной же суммой, которая определена в пакетах канального уровня дело обстоит иначе.

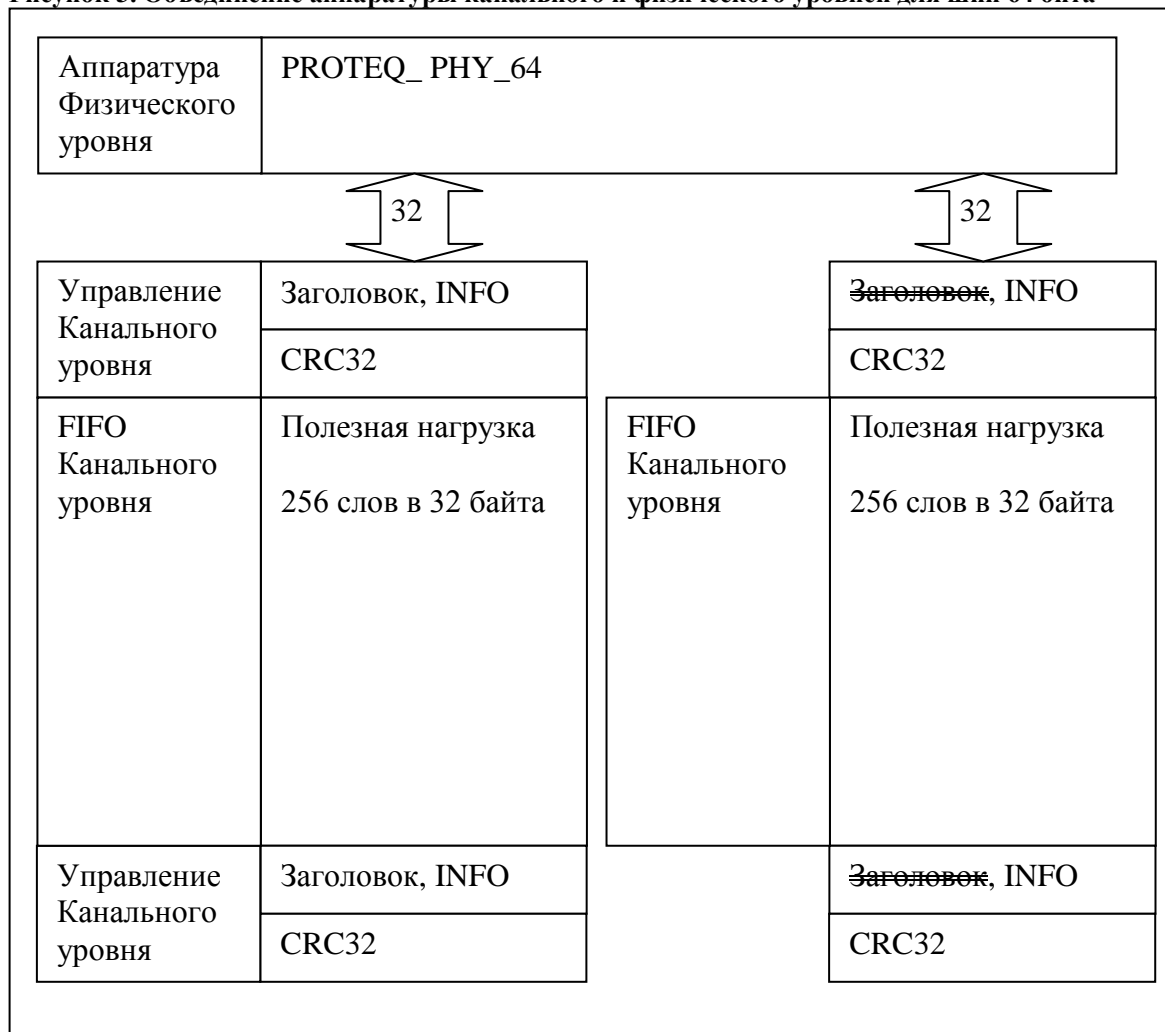
Аппаратура физического уровня подключается к аппаратуре канального уровня всегда по шине шириной 32 бита, как для случая 32-разрядной физики показано на рисунке 2.

Рисунок 2. Объединение аппаратуры канального и физического уровней для шин 32 бита



Если ширина шины аппаратуры физического уровня имеет ширину $N \times 32$, то к ней подключается N параллельных блоков канального уровня. Эти блоки имеют отдельные каналы пакетирования полезной нагрузки и общие цепи синхронизации и управления, как показано на рисунке 3 для случая с шиной 64 бита. Аналогично и для шин большей кратности.

Рисунок 3. Объединение аппаратуры канального и физического уровней для шин 64 бита



Для этих случаев по одному общему физическому уровню пересылается одновременно и параллельно несколько пакетов канального уровня. Каждый такой пакет содержит собственную полезную нагрузку, защищен собственными контрольными суммами, но имеет общий заголовок в младшей половине шины. Пакеты блока считаются принятыми успешно тогда и только тогда, когда все CRC32, участвующие в обмене, вычислены и приняты верно. Надо сказать, что информация прикладного уровня, входящая в заголовок, также пересылается независимо для всех пакетов канального уровня.

Следует отметить, что механизм размножения канального уровня на аппаратуре физического уровня является частью спецификации протокола PROTEQ сетевого уровня.

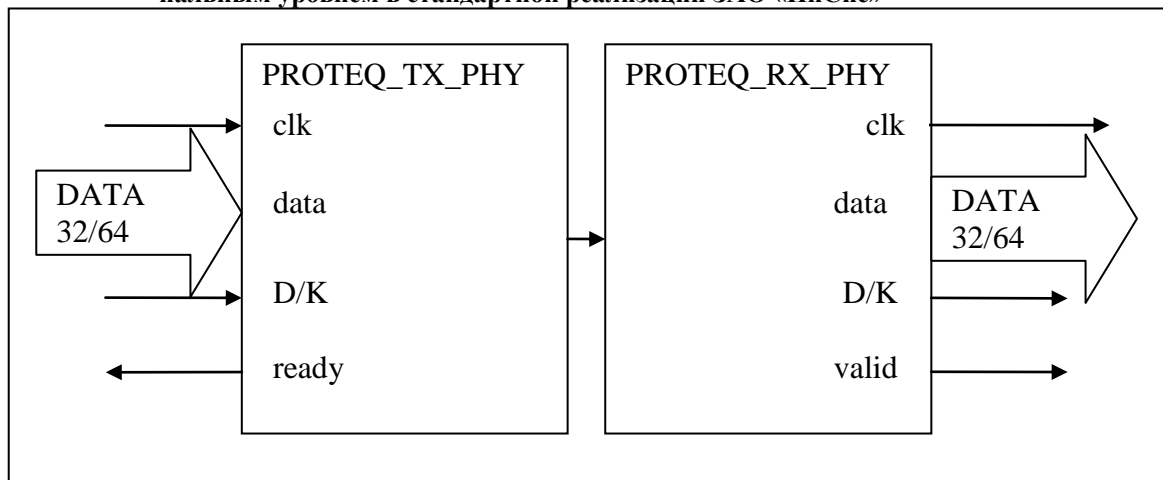
2.2 Стандартная реализация ЗАО «ИнСис»

В ЗАО «ИнСис» применение протокола PROTEQ сопряжено с использованием ПЛИС фирмы Xilinx. Физический уровень такого узла строится на блоках гигабитной передачи MGT (Multi Gigabit Transceiver). Начиная с ПЛИС серии 6, фирма Xilinx встраивает в MGT механизмы кодирования 64b/67b, а также синхронизации и восстановления частоты, используя для этого специальный элемент GEARBOX. Это позволяет с малыми затратами создавать аппаратуру физического уровня для различных протоколов. В случае

же с PROTEQ к блоку MGT достаточно присоединить схемы управления, сериализаторов и десериализаторов, которые размещаются в перепрограммируемой логике ПЛИС.

Стандартная реализация физического уровня предполагает синхронный интерфейс, в котором символы данных передаются на физический уровень или принимаются из него двумя последовательными 32-разрядными словами. Это касается и управляющих символов, которые передаются по тем же линиям, что и данные, но с соответствующим признаком. Интерфейсы приемника и передатчика приведены на рисунке 4.

Рисунок 4. Интерфейс двустороннего подключения PROTEQ между физическим уровнем и канальным уровнем в стандартной реализации ЗАО «ИнСис»



Особенностью реализации можно считать то, что для выравнивания скоростей передачи и приема, а также учета избыточности кодирования используется метод, встроенный в блок GEARBOX. На передающей стороне поток данных должен быть приостановлен синхронно с нулевым значением сигнала ready. На передающей стороне аналогичную роль играет сигнал valid.

Алгоритм выравнивания передачи может быть и внешним, тогда он основан на работе синхронного счетчика, считающего такты от 0 до 66 по циклу. Сигнал ready принимает нулевые значения в циклах 21, 45 и 65.

3. КАНАЛЬНЫЙ УРОВЕНЬ

Канальный уровень (Data Link) обеспечивает формирование пакетов полезной нагрузки с заголовками, защищенные контрольной суммой.

На канальном уровне поддерживаются два типа пакетов:

1. Служебный пакет
2. Пакет с данными

Передача и прием происходят с использованием аппаратуры физического уровня. Канальный уровень подключен к одному блоку PROTEQ_TX_PHY передачи и к одному блоку PROTEQ_RX_PHY приема. Используемый на канальном уровне механизм подтверждения приема осуществляется во встречном потоке.

На начало 2014 года было предложено два варианта протокола PROTEQ, имеющие существенные отличия в работе канального уровня стека. В PROTEQ V1.0 реализована версия для безостановочной передачи потоковых данных. В PROTEQ V2.0 добавлен механизм параллельной передачи двух потоков:

- одного скоростного, допускающего останов на стороне приема;
- одного медленного, не допускающего останов на приемном конце.

Медленный поток отнимает часть ресурсов, которые не влияют на надежность передачи быстрого потока, но в экстремальных ситуациях ограничивают его оперативность по подтверждению с приемной стороны.

Протоколы версий 1.0 и 2.0 неотличимы от по пакетной организации и содержимому пакетов, но часть передаваемых в пакетах управляющих полей интерпретируется по разному.

3.1 Служебный пакет

Служебный пакет – это короткий пакет без данных, который состоит только из одного 64-разрядного управляющего символов и не содержит данных.

№	Слово 32 бит	Имя	Назначение
0	Управляющий	CMD1	Команда начала пакета
1	Управляющий	CRC1	Контрольная сумма для слова 0

3.2 Пакет с данными

Пакет с данными – это пакет с полезной нагрузкой фиксированной длины. Размер полезной нагрузки фиксирован и равен 128 словам с данными по 64 разряда или 256 словам по 32 разряда (1024 байт). Пакет начинается и заканчивается 64-разрядный управляющими символами, которые совпадают по формату со служебными пакетами.

биты	Слово 32 бит	Имя	Назначение
0	Управляющий	CMD2	Команда начала пакета
1	Управляющий	CRC1	CRC32 для слова 0
2	Данные	DATA	Первое слово данных
...	Данные...	DATA	
257	Данные	DATA	256-е слово данных
258	Управляющий	CMD1	Команда конца пакета
259	Управляющий	CRC2	Контрольная сумма для слов от 0 до 258

3.3 Слово команды

Служебный пакет – это короткий пакет без данных, который состоит только из одного 64-разрядного управляющего символов и не содержит данных.

№	Имя	Назначение
31..28	SIG	0xB – Сигнатура (тип управляющего пакета)
27	CMD2/1	1 – пакет с данными (CMD2), 0 – служебный пакет (CMD1) или завершение пакета

26	-	Резерв
25	RX_ENABLE	1 – Готовность приема
24	PKG_TOGLE	Расширение номера буфера (только для CMD2)
23..22	BUF_NUM	Номер буфера (только для CMD2)
21	-	Резерв
20	-	Резерв
19	ACK_TGL3	Флаг подтверждения для пакета с BUF_NUM = 3
18	ACK_TGL2	Флаг подтверждения для пакета с BUF_NUM = 2
17	ACK_TGL1	Флаг подтверждения для пакета с BUF_NUM = 1
16	ACK_TGL0	Флаг подтверждения для пакета с BUF_NUM = 0
15..12		Поле сетевого уровня
11..8		Резерв
7..0		Поле прикладного уровня

3.3.1 SIG

Сигнатура, уникальный номер для распознавания пакета и протокола. Для этой версии PROTEQ используется шестнадцатеричная сигнатура 0B.

3.3.2 RX_ENABLE

Сообщение удаленному концу, что приемная аппаратура готова к приему данных: прошла инициализацию, ресурсы выделены и цепи тактируются.

3.3.3 BUF_NUM

Индекс пакета по порядку от 0 до 3. Вместе с PKG_TOGLE обеспечивает циклическую нумерацию пакетов от 0 до 7.

3.3.4 PKG_TOGLE

Вместе с BUF_NUM обеспечивает циклическую нумерацию пакетов от 0 до 7. PKG_TOGLE – старший разряд номера.

3.3.5 ACK_TGLx

Флаг подтверждения приема пакета:

- значение флага изменилось – пакет подтвержден
- значение флага неизменно – ожидание подтверждения

Имеется четыре флага для каждого буфера с индексами от 0 до 3.

Контрольная сумма

32-разрядная контрольная сумма вычисляется по всем предшествующим символам пакета. CRC1 – по 32-битному слову CMD1. CRC2 – по всем 32-битным словам данным, CMD2, CRC1 и CMD1.

Для подсчета суммы используется 32-разрядный полином

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

Начальное состояние 0.

3.4 Очередь пакетов PROTEQ V1.0

Все пакеты с полезной нагрузкой нумеруются передающей стороной с возрастанием, начиная с 0 и до 7 по циклу. Номер передается в полях PKG_TOGLE и BUF_NUM таких пакетов. Имеется следующее правило:

1. Принимающая сторона распознает порядок пакетов по этому номеру и подтверждает прием сразу же, как только удачно примет этот пакет.
2. Передающая сторона прекращает передачу подтвержденных пакетов.
3. Одновременно к передаче может быть выставлено не более 4-х пакетов, начиная от первого не подтвержденного. Таким образом, формирование новых пакетов может задержаться, если в очереди на отправку находится пакет с меньшим номером.

3.5 Очередь пакетов PROTEQ V2.0

В протоколе PROTEQ версии V2.0 поддерживается не один, а два независимых потока. Из-за этого к правилам PROTEQ V1.0 были добавлены новые:

4. Поступающие на передачу данные первого (скоростного потока) упаковываются в полезную нагрузку не всех пакетов, а только с номерами PKG_TOGLE, BUF_NUM 1, 2, 3, 5, 6 и 7;
5. Поступающие на передачу данные второго (командного потока) упаковываются в полезную нагрузку не всех пакетов, а только с номером PKG_TOGLE, BUF_NUM 0 и 4;
6. Поступление данных обоих потоков происходит независимо друг от друга;
7. Отправка неподтвержденных пакетов происходит подобно PROTEQ V1.0 в общем цикле.
8. Подтвержденный пакет командного потока освобождается для заполнения следующими данными этого потока;
9. Подтвержденный пакет скоростного потока освобождается для заполнения следующими данными этого потока в соответствии с очередностью.
10. На принимающей стороне командный поток выделяется из пакетов с номерами PKG_TOGLE, BUF_NUM 0 и 4 и отправляется на прикладной уровень сразу же после приема независимо от приема пакетов скоростного потока;
11. Подтверждение приема пакетов командного потока происходит во встречных пакетах флагом ACK_TGL0;
12. На принимающей стороне скоростной поток выделяется из пакетов с номерами PKG_TOGLE, BUF_NUM 1, 2, 3, 5, 6 и 7 и отправляется на прикладной уровень после приема в соответствии с очередностью, но независимо от приема пакетов командного потока;
13. Подтверждение приема пакетов командного потока происходит во встречных пакетах флагами ACK_TGL1, ACK_TGL2 и ACK_TGL3.

Пакеты стека PROTEQ V2.0 отличаются только на канальном и прикладном уровне и не отличаются на физическом и сетевом уровнях. По содержимому пакетов невозможно определить, к какой версии протокола они относятся.

3.6 Подтверждение пакетов

Буфера подтверждаются независимо для каждого индекса. Имеется две схемы подтверждения:

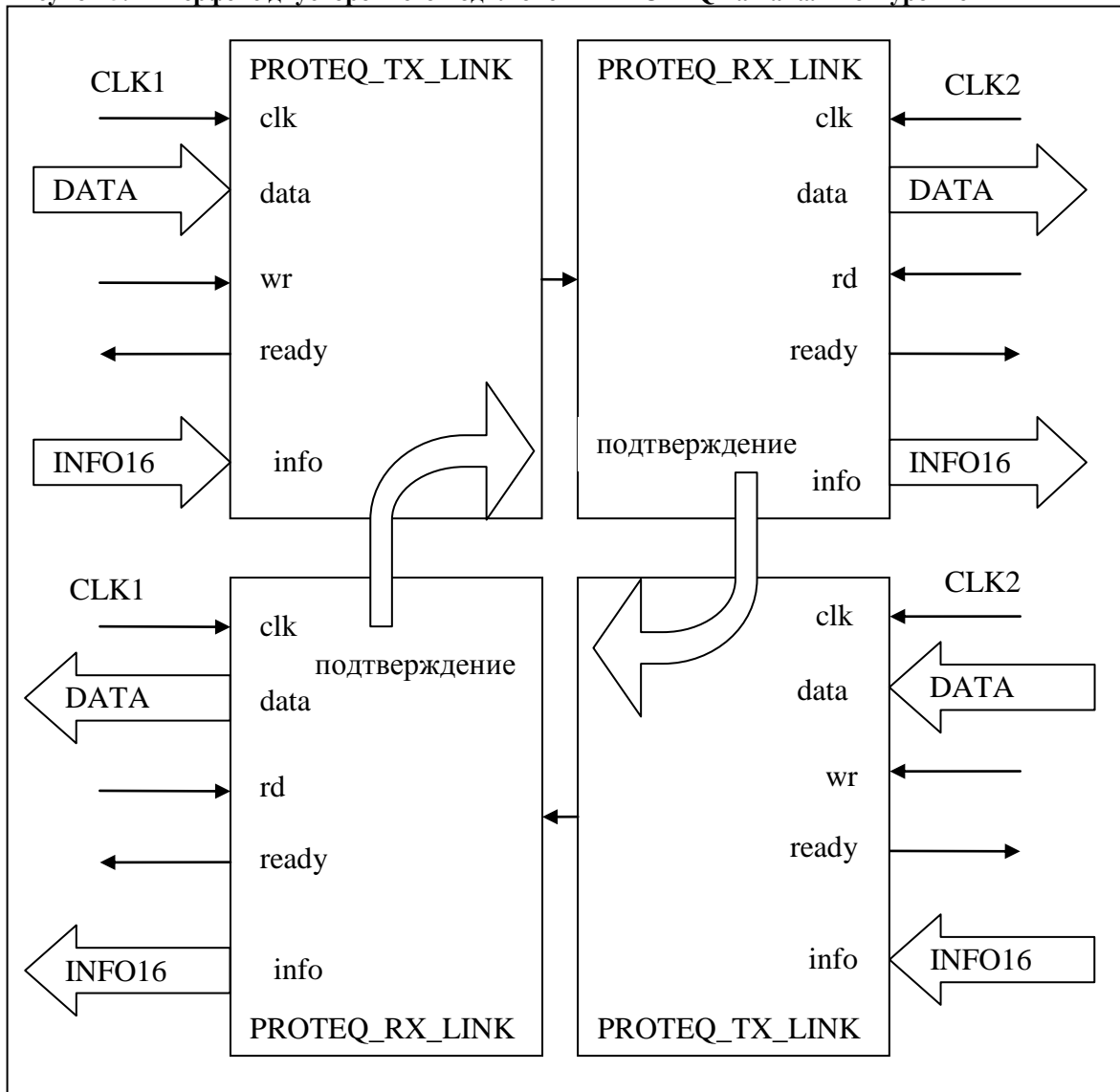
1. С приоритетом передачи
Прием подтверждается сразу же, после того как буфер был принят. В этом случае буфер перестает передаваться и вместо него может начаться передача буфера с противоположным PKG_TOGGLE. Прием этого пакета начнется только после того, как буфер приема будет свободен.
2. С приоритетом приема
Прием подтверждается только после того, как буфер приема будет свободен. В этом случае принятый пакет будет передаваться повторно, и прекратится только после освобождения буфера приема.

Оба метода могут быть применены при реализации протокола PROTEQ.

3.7 Интерфейс аппаратуры канального уровня

Интерфейс канального уровня оперирует с потоковыми данными и дополнительной статической информацией, передаваемой с предыдущего уровня. Блок-схема подключения дана на рисунке 5.

Рисунок 5. Интерфейс двустороннего подключения PROTEQ на канальном уровне



Неформатные потоковые данные DATA передаются на канальный уровень синхронно по стробу WR. Данные можно записывать только при наличии сигнала READY. Данные передающей стороны желательно группировать блоками по 1024 байт.

Переданные потоковые данные снимаются с канального уровня синхронно по стробу RD. О наличии данных говорит сигнал READY. Данные приемной стороны сгруппированы блоками по 1024 байт.

Информация INFO сопровождает данные DATA и запоминается в аппаратуре для каждого формируемого пакета в начале блока данных размером 1024 байта. При приеме эта информация выделяется из пакета. Размер вектора INFO 16 бит.

3.8 Реализация ЗАО «ИнСис»

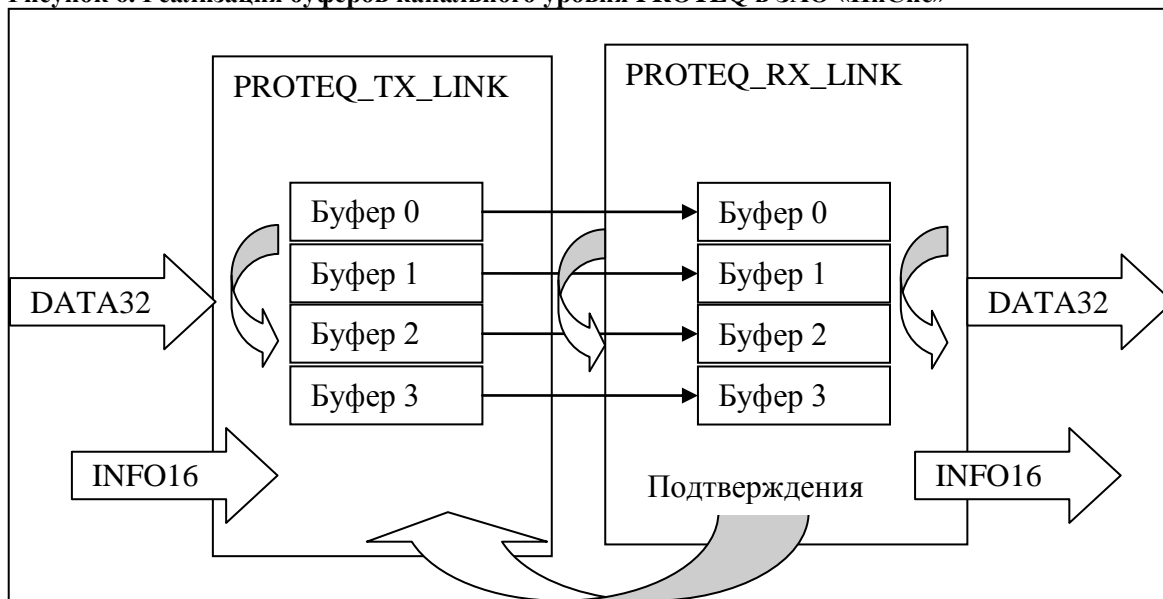
Внутри блока PROTEQ_TX_LINK имеется четыре независимых буфера по 1024 байт каждый, предназначенные для формирования и отправки пакетов с данными. Каж-

дому буферу присвоено номер от 0 до 3 равный одному из возможных значений BUF_NUM номеров пакета. Первоначально все буфера свободны. Свободные буфера заполняются строго по циклу, начиная с номера 0. Далее содержимое полностью заполненных буферов передается в пакетах на физический уровень до тех пор, пока их пакеты не будут подтверждены. Если имеется несколько неподтвержденных буферов, то их отправка чередуется. Буфера, чьи пакеты подтверждены, становятся свободными и их содержимое перестает передаваться.

Аналогично, на приемном конце имеется четыре независимых буфера по 1024 байта каждый, предназначенные для приема данных из пакетов, чьи номера BUF_NUM также совпадают с порядковым номером буфера. Данные принимаются из пакетов только в свободные буфера. Данные же из заполненных буферов вычитываются последовательно, начиная с буфера с номером 0 и далее по порядку. Вычитанный буфер становится свободным. Подтверждение происходит по каждому буферу независимо во встречном потоке путем инвертирования соответствующего BUF_NUM флага подтверждения сразу же после заполнения буфера.

Структуру реализации иллюстрирует рисунок 6.

Рисунок 6. Реализация буферов канального уровня PROTEQ в ЗАО «ИнСис»



4. СЕТЕВОЙ УРОВЕНЬ

Сетевой уровень организации предполагает объединение нескольких канальных уровней для параллельного, одновременного или независимого обмена между аппаратурой в рамках одного узла подключения.

Сетевой уровень позволяет собрать несложную сеть объединения аппаратных узлов, топология которых определяется жестко или управляется с прикладного уровня в каждом узле независимо.

Сетевой уровень отражен в пакетах 4-битным полем, который может быть интерпретирован как номер маршрута установленного соединения. Расположение поля дано в таблице. Номер маршрута это уникальный номер, который определяет, кем будет обработан пакет при получении. В сложных сетях номера маршрутов могут дублироваться, но должны быть уникальны в каждом приемном узле.

№	Имя	Назначение
31..28	SIG	0xВ – Сигнатура (тип управляющего пакета)
27	CMD2/1	
26	-	
25	RX_ENABLE	
24	PKG_TOGLE	
23..22	BUF_NUM	
21	-	
20	-	
19	ACK_TGL3	
18	ACK_TGL2	
17	ACK_TGL1	
16	ACK_TGL0	
15..12	TRACE_NUM	Номер маршрута (сетевой уровень)
11..8		Резерв
7..0		Поле прикладного уровня

Далее на примерах будут рассмотрены некоторые варианты исполнения PROTEQ на сетевом уровне.

4.1 Общие принципы и правила

Протокол PROTEQ предназначен и, как правило, используется для передачи потоковой информации из ПЛИС в другую ПЛИС или из устройства в устройство. Один узел канального уровня, соединенный зеркально с другим узлом канальной передачи позволяет легко осуществить такую передачу для потока слов заданной ширины. Так в случае с реализацией на ПЛИС Xilinx ширина слов может быть кратна 32 бит.

Увеличение ширины слов потока можно выполнить, объединив несколько узлов канального уровня в узел сетевого уровня. Все узлы канального уровня должны при этом работать синхронно и одновременно отправлять и получать данные, при этом внутренняя работа каждого канального узла будет независима от других. Это касается в первую очередь не предсказуемости в процессе возникновения ошибок на линии и взаимного дрожания частот в блоках MGT.

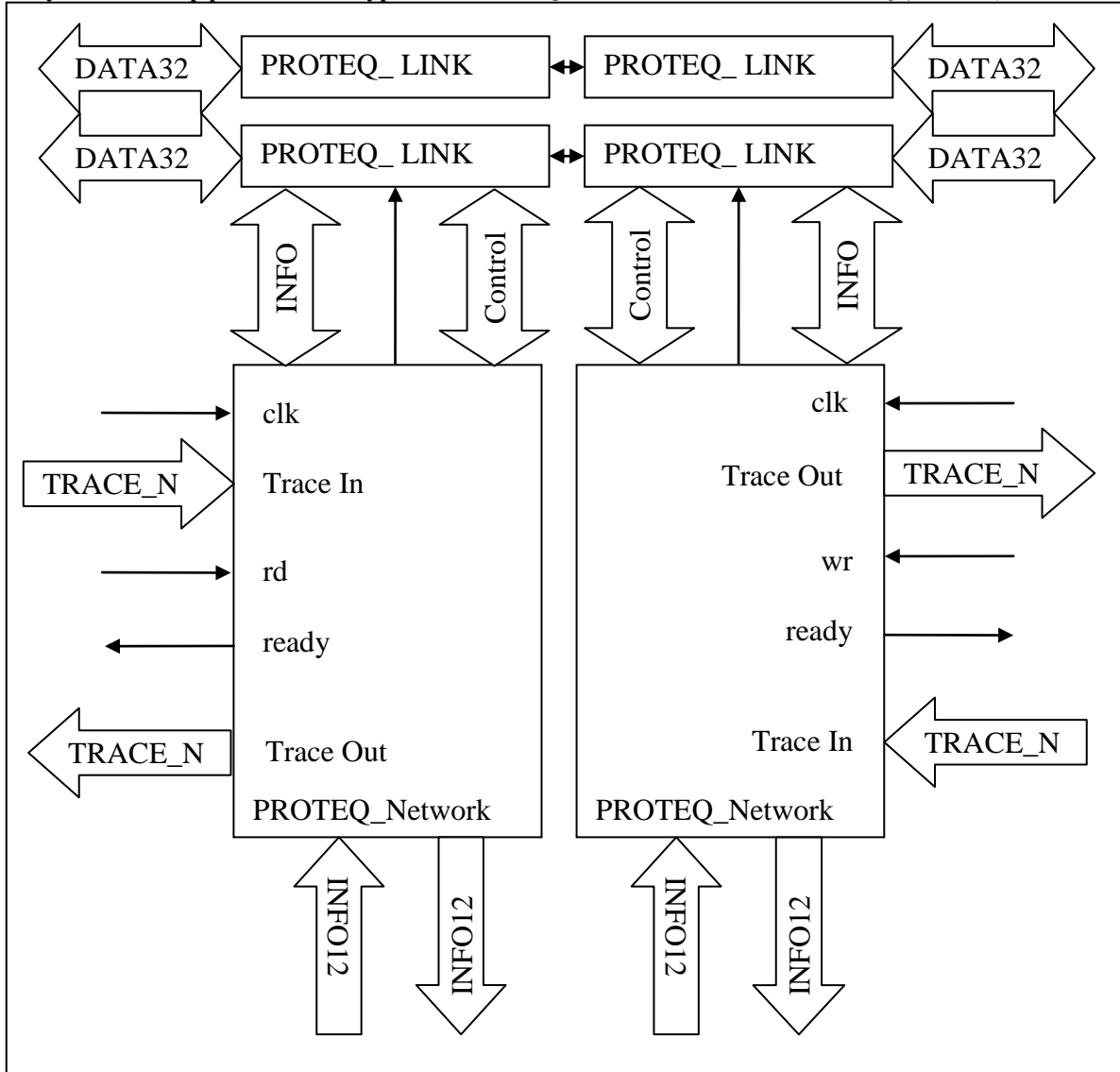
Кроме того возможна такая архитектура, когда данные могут быть переданы сразу нескольким источникам одновременно, по очереди или с выбором направления.

Аппаратура сетевого уровня предназначена для получения необходимой синхронности и одновременности работы всех узлов канального уровня.

4.2 Соединение точка-точка

Это соединение аналогично простейшему подключению PROTEQ, но вместо одного блока MGT используются несколько параллельных (Lane). Это позволяет поднять скорость обмена полезной информацией, не меняя скорость MGT. В протоколе PROTEQ соединение по двум Lane будет выглядеть так, как показано на рисунке 7.

Рисунок 7. Интерфейс сетевого уровня PROTEQ в соединениях точка-точка (Два Lane)

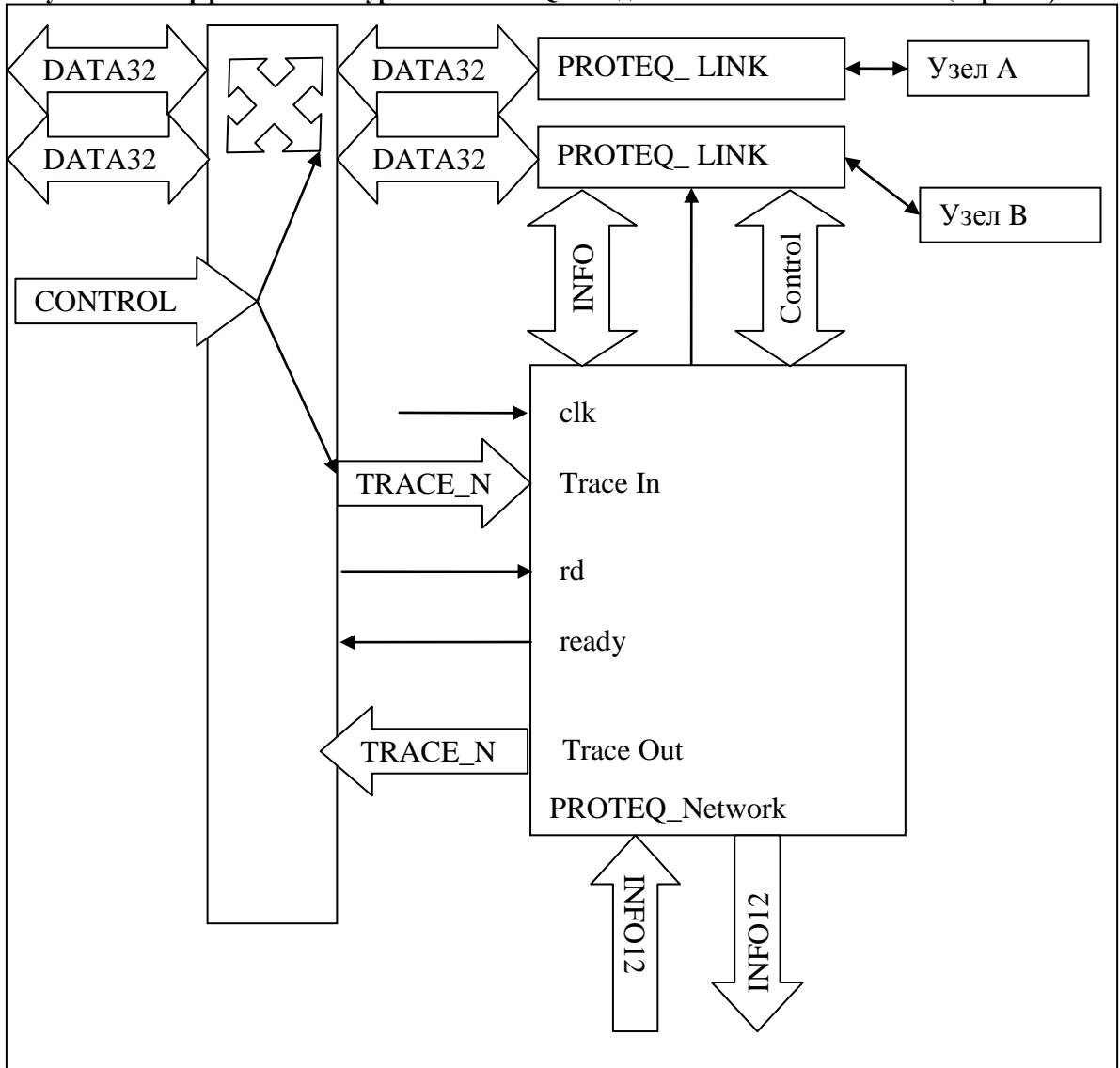


4.3 Соединение точка-многоточие

Это соединение полезно тогда, когда ведется широковещательная передача, или требуется разделить потоки.

Топология сети проиллюстрирована на рисунке 8.

Рисунок 8. Интерфейс сетевого уровня PROTEQ в соединениях точка-многоточие (вариант)



5. ПРИКЛАДНОЙ УРОВЕНЬ

Полезная нагрузка формируется и утилизируется на прикладном уровне. Уровни ниже прикладного и выше сетевого в протоколе PROTEQ не определены. Для дополнительной сигнализации между этими уровнями в пакете PROTEQ зарезервировано 8 битное поле INFO, как показано в таблице.

№	Имя	Назначение
31..28	SIG	0xB – Сигнатура (тип управляющего пакета)
27	CMD2/1	
26	-	
25	RX_ENABLE	
24	PKG_TOGGLE	
23..22	BUF_NUM	
21	-	
20	-	
19	ACK_TGL3	
18	ACK_TGL2	
17	ACK_TGL1	
16	ACK_TGL0	
15..12	TRACE_NUM	Номер маршрута (сетевой уровень)
11..8		Резерв
7..0		INFO (прикладной уровень)

Поле INFO имеет ширину кратную 8 битам, аналогично ширине шины для полезной нагрузки и зависит от конструкции аппаратуры узла на сетевом уровне. Так, если ширина шины полезной нагрузки равна $32 \times N$, то ширина поля INFO будет $8 \times N$ в битовом выражении.