

RapidIO – КОММУТАЦИОННАЯ СТРУКТУРА ПОСЛЕДОВАТЕЛЬНОГО ТИПА

Н.Слепов
nslepov@online.ru

Прогресс в области компьютерной техники в большой степени был обязан не только развитию микропроцессоров (МП) и оперативной памяти, но и методов передачи потока данных между ними. Сначала для передачи данных применялись шины, которые непрерывно совершенствовались, увеличивая разрядность и тактовую частоту – два показателя, определяющие производительность шины или емкость потока данных. В последние годы определяющими были две шины: VMEbus и PCI [1]. Обе достигли близкой к предельной разрядности (64 разряда) и тактовой частоты 320 МГц (VMEbus) и 533 МГц (PCI). Каждый новый шаг в увеличении тактовой частоты и миниатюризации конструкции шины давался все труднее и с большей затратой средств, а использование удвоенной частоты 1066 МГц оказалось труднорешаемой задачей.

В результате в 2002 году компания Intel вообще отказалась от модернизации параллельной шины и внедрила новую концепцию шины PCI Express – коммутационной структуры последовательного типа (КСПТ) [1]. В предлагаемой статье описывается одна из наиболее проработанных КСПТ, названная технологией быстрого ввода/вывода – RapidIO [2] (нерасшифрованные в тексте сокращения см. во врезке).

ОСНОВНЫЕ ОСОБЕННОСТИ КСПТ

КСПТ представляют собой коммутационную систему с внутренней сетью, соединяющей обслуживаемые системой устройства (узлы сети) так, чтобы передать данные с максимально возможной скоростью (сегодня это 10 Гбит/с) последовательно, используя один или несколько потоков, как это делает кросс-коммутатор в системах связи. При этом циркулирующая по сети информация формируется в конечных точках и передается в виде пакетов данных, сигнальных пакетов (сигналов), сообщений и уведомлений (дейтаграммного типа). В заголовках па-



кетов и сообщений содержится информация для их идентификации, маршрутизации на внутренней сети, обработки в точке приема с обнаружением ошибок и повторной передачей для окончательного безошибочного приема. Передача может проводиться и в скоростном потоковом режиме, но без повторения в случае возникновения ошибок.

Это не просто новый интерфейс, как думают некоторые, а система (или распределенная сеть) обрабатывающих элементов (ОЭ), виртуально соединенных между собой, например с помощью двух маршрутизирующих коммутаторов. В этом смысле она больше напоминает виртуальную локальную сеть на базе коммутаторов Ethernet. Как и сеть, она имеет многоуровневую модель взаимодействия. В данной реализации представлено три уровня: физический, транспортный и логический (с соответствующими протоколами).

КСПТ пришли на смену параллельной шине, которая, достигнув 64-разрядной ширины (требующей много места на чипе) и гигагерцевого рубежа тактовой частоты, оказалась неспособной обеспечить дальнейшую поддержку растущих возможностей микропроцессоров (МП) не только в плане роста тактовой частоты, но и в плане многопоточности и многоядерности.

Существуют пять признанных КСПТ: на базе коммутаторов Ethernet, на базе структур типа InfiniBand, StarFabric (StarFabric Trade Association – STA), PCIExpress (Intel) и структуры RapidIO (RapidIO Trade Association – RTA).

Для обеспечения высокой производительности коммутаторы этих структур должны иметь простую архитектуру и тип связи (например, "точка-точка") между устройствами, хотя система коммутации может иметь более сложную архитектуру, включающую свои коммутаторы, маршрутизаторы и концентраторы, и может поддерживать большинство других известных топологий.



Преимущества КСПТ

По сравнению с шинной архитектурой КСПТ позволили:

- значительно сократить число сигнальных линий (примерно в пять раз);
- уменьшить размеры разъемов и кабелей, что дало возможность уплотнить монтаж, упростить компоновку, снизить стоимость и упростить обслуживание;
- облегчить масштабирование за счет объединения нескольких синхронных потоков (полос передачи) путем мультиплексирования на физическом уровне;
- увеличить эффективность и легкость маршрутизации;
- использовать "горячую вставку" и "горячую замену" карт/плат благодаря использованию для передачи физического уровня;
- упростить топологию (используя в основном топологию "точка-точка"), что позволило легко изолировать отказы и использовать резервирование для повышения надежности системы.

Топологии коммутационных структур

Коммутационные структуры используют следующие топологии коммутируемых межсоединений:

- звезда, двойная звезда, сдвоенная двойная звезда (топологии, называемые централизованными);
- кросс-бар, двойной кросс-бар;
- полносвязная ячеистая сеть, неполносвязная сеть (топологии, называемые распределенными).

Наиболее часто используются следующие топологии (рис. 1): звезда, двойная звезда и полносвязная ячеистая сеть.

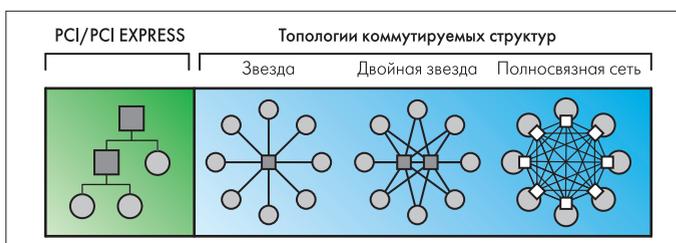


Рис. 1. Основные топологии коммутационных структур

С точки зрения сетевой топологии это может быть *линейная последовательная цепь* (часто называемая потоком, или полосой) или *звено связи* из нескольких параллельно работающих цепей (нескольких полос), мультиплексируемых по типу *байт-интерливинга* [3], для формирования более широкой полосы пропускания или большей скорости передачи, возрастающей пропорционально числу полос – коэффициенту мультиплексирования. Число используемых полос может быть 1, 2, 4, 8, 16 и 32, а число звеньев – 1, 2, 4, 8.

ОБЩАЯ ХАРАКТЕРИСТИКА СТАНДАРТА

Коммутационная структура RapidIO разрабатывалась, начиная с 1997 года, компаниями Motorola и Mercury как перспективная система связи для встраиваемых компьютерных систем (ВКТ).

Ее основная цель – создание простого и эффективного протокола, неотягощенного программным обеспечением (ПО) и сфокусированного на внутренней связи (коммутации соединений между чипами и картами встраиваемых систем COTS реального времени [4]). Уже в 1999 году была закончена первая версия стандарта, а в 2001 году были получены первые ИС (массивы FPGA, процессоры, мосты и коммутаторы).

Предполагалось, что система будет расширена до уровня, удовлетворяющего жестким требованиям компаний связи. Окончательно система RapidIO была сформирована в августе 2004 года. Она поддерживается Ассоциацией RTA. Наиболее полно RapidIO, кроме стандартов RTA [2], нашла также отражение в стандартах PICMG и VITA [1], поддерживающих современные технологии ВКТ и, в частности, технологию ATCA (стандарт PICMG 3.5).

Успешный старт

Стандарт RapidIO [2], по оценке специалистов [4], сегодня проник на все наиболее важные рынки ВКТ: военный, систем связи, хранения данных и обработки изображений. Он реализован в рамках стандартов PICMG и VITA (см. таблицу [4]). Такой популярности RapidIO достиг благодаря тому, что смог органично вписаться в цикл разработки систем ВКТ новой генерации, использующих приложения, предлагаемые на этих рынках. Он был поддержан такими важными игроками на рынке ИС, программного обеспечения и средств измерений, как компания Freescale Semiconductor (с ее новым семейством связанных процессоров PowerQUICC, поддерживающих RapidIO, например PQIII 8548) и компания TI (анонсировавшая новый цифровой сигнальный процессор (ЦСП), также поддерживающий RapidIO). Протокол RapidIO используется в программах таких знаковых компаний, как Northrop Grumman и Lockheed Martin. И, наконец, он поддержан компаниями Agilent и Tektronix – законодателями мод в измерительной технике [4].

Какие же требования, предъявленные при разработке этого стандарта, позволили ему столь успешно стартовать?

Основные требования к системе RapidIO:

- высокая производительность, обслуживание множества ОЭ/МП;
- наличие общей памяти, прямой доступ к памяти (канал ПДП) и возможность распределенной обработки;
- прямое одноранговое взаимодействие между элементами системы;
- поддержка различных операционных систем (ОС);
- поддержка различных топологий (с использованием механизмов анализа состояния маршрутов и организацией резервных путей);

Использование системы RapidIO в других технологиях и стандартах

Технологии*	VME VX	VPX	ATCA	CompactPCI	XMC	AMC
Версии с RapidIO	VITA 41.2	VITA 46.3	PICMG 3.5	PICMG 2.18	VITA 42.2	AMC.4

* Описание технологий см. в работе [1].

- использование пакетной передачи для повышения гибкости системы передачи;
- поддержка систем любого размера, от малого до большого, теми же или совместимыми форматами пакетов;
- независимость передачи от ширины полосы и задержки физической структуры;
- независимость спецификации пакета от ширины физического интерфейса с другими устройствами;
- возможность организации мультикастинга (передачи типа "точка-многоточка");
- высокая надежность (автоматический повтор передачи и синхронизация устройств, обработка ошибок на системном уровне);
- управление качеством обслуживания (QoS).

АРХИТЕКТУРА СИСТЕМЫ RapidIO

Модель с обрабатывающими элементами

На рис. 2 представлена блок-схема RapidIO, использующая модель системы с обрабатывающими элементами (ОЭ) [2a]. Центральным связующим блоком системы является структура взаимосвязи системы RapidIO, которая коммутирует набор парных связок – обрабатывающий элемент-локальная память (ОЭ-ЛП), на одной стороне структуры, с мостом (на другой стороне), обеспечивающим выход на сервисные подсистемы ввода/вывода, представленные интерфейсами высокоскоростной шины PCI, портами гигабитного Ethernet, высокопроизводительным последовательным портом FireWire компьютеров Apple, подсистемами управления прерываниями и другими подсистемами и интерфейсами.

Связка ОЭ-ЛП в этой модели представлена следующими шестью типами:

- процессор-агент-память, с выходом на систему RapidIO через агента;
- процессор-память, с выходом на систему RapidIO через процессор;
- только память с блоком управления, с выходом на систему RapidIO через блок управления памятью;
- процессор-агент, с выходом на систему RapidIO через агента;
- элемент ввода-вывода, с выходом на систему RapidIO через мост;

- коммутирующий процессорный элемент, базовый элемент системы RapidIO.

Многоуровневая архитектура

Архитектура RapidIO интересна тем, что она модульная, легко масштабируется и имеет общий транспортный уровень, позволяя осуществить связь типа: микрочип-микрочип или плата-плата через соединительную панель. Эта архитектура иерархическая и многоуровневая. Стандарт [2] описывает три уровня: логический, транспортный и физический – но это не уровни OSI, хотя они и выполняют похожие функции. Описан также дополнительный – организационный или служебный уровень. Эти уровни играют в системе следующие роли:

- *логический* – описывает систему ввода-вывода, передачу пакетов/сообщений, глобальную память с разделением ресурсов, организацию и управление потоками данных;
- *транспортный* – описывает общую организацию транспорта пакетов;
- *физический* – ориентирован на использование объединительной панели и описывает параллельную 8-/16-разрядную передачу пакетов данных, последовательную передачу 1x/4x (однополосную/четыреполосную) с использованием XAUI-совместимого электрического интерфейса, а также другие скоростные методы передачи;
- *служебный* – решает задачи системной загрузки, конфигурации системы, взаимодействия между ОЭ/устройствами и RapidIO, осуществляет контроль ошибок, организует режим мультикастинга и решает другие системные вопросы.

Рассмотрим особенности и спецификацию каждого уровня более подробно.

СПЕЦИФИКАЦИЯ ФИЗИЧЕСКОГО УРОВНЯ

Спецификация физического уровня приведена в двух стандартах [2] и определяет сигнал, протокол передачи, форматы пакетов, управление потоком и обработку ошибок. Параллельная передача описана в стандарте [2г], последовательная – в [2е]. Форматы данных, поддерживаемых системой: байт (8 бит), полуслово (16 бит, или 2 байта), слово (32 бита, или 4 байта) и двойное слово (64 бита, или 8 байтов).

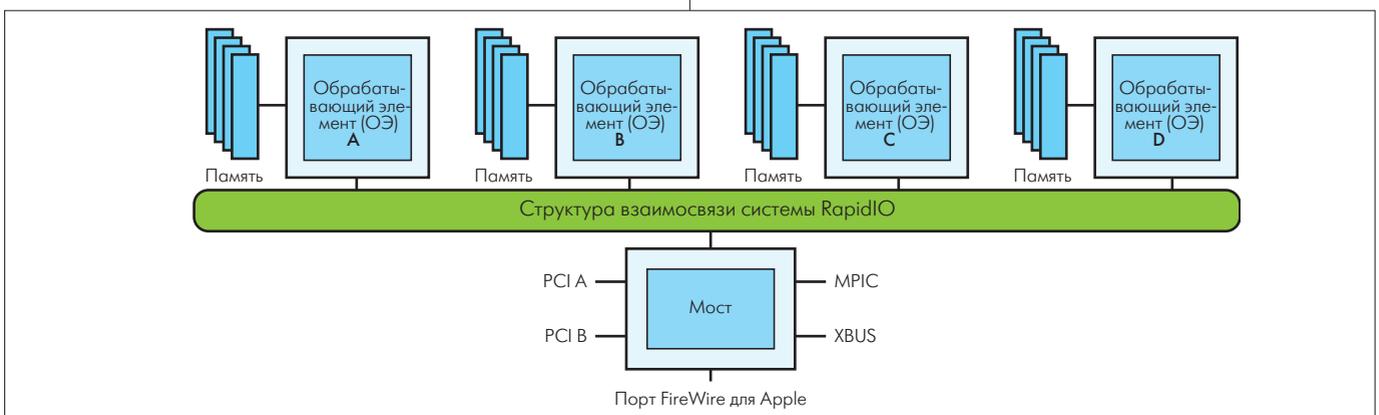


Рис.2. Вариант схемы вычислительной системы с RapidIO



Физический уровень. Параллельная передача данных

Стандарт [2г] определяет параллельную 8-/16-разрядную одностороннюю передачу типа "точка-точка" (обозначение 8/16 LP-LVDS) с помощью интерфейса, имеющего выделенные 8- или 16-разрядные порты с тактовыми сигналами, синхронизирующими пакет данных по переднему и заднему фронтам или кадровыми эталонными управляющими сигналами. Дифференциальная сигнализация (LVDS) применяется для упрощения интерфейса, устойчивости сигнала и обеспечения хорошей масштабируемости по частоте при использовании различных плат и разъемов.

RapidIO обеспечивает механизм управления потоком данных. Протоколы физического уровня и форматы пакетов практически независимы от используемой топологии, типа соединений, ширины полосы и запаздывания. Они одинаково обрабатывают упорядоченные и неупорядоченные пакеты и устойчивы к высокочастотным переходным помехам, генерируемым интерфейсными операциями.

Протокол физического уровня

Протокол физического уровня – это протокол обмена пакетами и управляющие символы, с помощью которых элемент назначения или элемент промежуточной обработки (такой, как коммутатор) подтверждает прием от источника пакета запроса или отклика. Если пакет не может быть принят по ка-

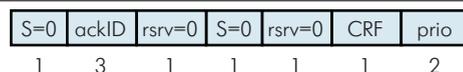


Рис. 3. Формат полей заголовка физического уровня для пакетов

кой-то причине, управляющий символ указывает, что исходный пакет должен быть передан вновь. Такой алгоритм обеспечивает управление потоком и упорядочение транзакций между ОЭ.

Выравнивание, форматы заголовков пакетов и символов управления

Все пакеты выравниваются по 16-разрядным границам (полуслову), а пакеты с символами управления, пересылаемые через 8-/16-разрядные порты, выравниваются по 32-разрядным границам (слову), что снижает внутреннюю частоту обработки. Если выравнивания в пакете нет, добавляются пустые биты. Такое выравнивание позволяет устройствам работать с внутренней шириной, составляющей 32 разряда (ширина слова), что позволяет снизить их рабочую частоту.

Все символы управления имеют номинальную ширину 16 разрядов (ширина полуслова), но они дополняются 16-разрядной побитно-инвертированной копией для 32-разрядного выравнивания, что увеличивает возможности обнаружения ошибок. Такие 32-разрядные символы называются *выровненными символами управления*.

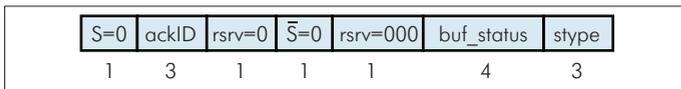


Рис.4. Формат полей заголовка физического уровня для символов управления

Физический уровень (будучи нижним из трех) добавляет к пакету свой заголовок, который оказывается его внешним заголовком. Таких заголовков два – один для пакетов, второй для символов управления. Форматы полей заголовка физического уровня для пакетов приведены на рис. 3, а для символов управления – на рис. 4 (длина поля в битах указана снизу). Эти поля используются для управления потоком, контроля ошибок и выполнения системных функций и означают следующее:

- S – S-бит: "0" – для пакетов запроса/отклика, "1" – для символов управления;
- \bar{S} – инвертированный S-бит, равный "1" (избыточный бит нечетности);
- ackID – идентификатор подтверждения приема/квотирования пакета, посланный отправителю;
- CRF – опциональный бит, позволяющий отличить критичные (важные) потоки среди потоков, имеющих одинаковый приоритет (если CRF не поддерживается, то бит считается резервным);
- prio – биты указателя приоритета: 00 – минимальный, 01 – средний, 10 – высокий, 11 – самый высокий приоритет;
- buf_status – 4 бита, определяющие число доступных пакетных буферов в принимающих устройствах;
- stype – тип символа управления;
- rsrv – резервное поле.

Эти заголовки добавляются к форматированной последовательности транспортного (tt) и логического (ftype) заголовков пакета, представляя в целом кадр пакета, завершаемого полем контроля ошибок – CRC-16, как показано ниже на рис. 6 для пакета длиной не более 80 байтов, где CRC занимает 81 и 82 байты. На рисунке показаны границы 32-/16-разрядного выравнивания. Первые 16-бит поля кадра пакета называются *первым символом* (рис. 5). Эти 16 бит защищены следующим образом: 1-е и 4-е поля (S и его инверсия) защищены контролем четности; 2-е и 3-е поля защищены протоколом, а остальные поля защищены CRC.

Максимальная длина пакета при параллельной передаче составляет 276 байтов, а переносимая полезная нагрузка 256 байтов. Поэтому, если длина пакета (без поля CRC) превышает 80 байтов, то используется второе поле CRC-16 в конце пакета, дополняемое при необходимости пустым полем длиной до 16 бит. При этом общий кадр пакета состоит из полей:

- Header – поле заголовков физического, транспортного и логического уровней (2 байта);
- Source ID – идентификатор источника транспортного уровня (2 байта);
- Destination ID – идентификатор назначения транспортного уровня (2 байта);
- Trans/wrsize – поля логического уровня (1 байт), определяющие класс "транзакция" ("trans") среди классов форматов, и размер записи ("wrsize");
- srcTID – поля логического уровня (1 байт), определяющие ID транзакции пакета;
- Address – различные адреса, формируемые на логическом уровне (8 байтов);
- Payload – полезная нагрузка (256 байтов);
- CRC – поля контроля ошибок (8 байтов).

Передача пакетов и символов управления осуществляется через 8-/16-разрядные порты (отличающиеся только числом контактов физического интерфейса). Управление синхронизацией (выравниванием начала) кадра осуществляется его сопоставлением с сигналом типа NRZ, полярность которого переключается в момент начала первого символа пакета или первого символа управления. ОЭ на приеме (для выборки данных и сигналов синхронизации) снабжается эталонным тактовым сигналом данных, полярность которого переключается при каждом переходе через контакты интерфейса. Этот тип эталонного сигнала называют *тактовым сигналом с удвоенной скоростью передачи данных* (DDR clock). В результате тактовый сигнал синхронизации переключается через каждые четыре байта, фиксируя границы 32-разрядного выравнивания, а эталонный сигнал данных – через каждые два байта для 16-разрядного порта и через каждый байт для 8-разрядного порта.

Параллельный физический интерфейс RapidIO имеет специальную систему распределения сигнала тактовой синхронизации, позволяющую исключить перекося и джиттер тактовых импульсов благодаря использованию гибких временных буферов.

Драйверы переменного тока позволяют использовать большой набор тактовых частот для системы DDR: 250, 375, 500, 750 и 1000 МГц, что делает возможным соответствующий набор скоростей передачи: 500, 750, 1000, 1500 и 2000 Мбит/с.

Физический уровень. Последовательная передача данных
Стандарт [2e] определяет требования к устройствам, использующим электрическую среду для последовательной связи в системе RapidIO. Он определяет последовательный дуплексный

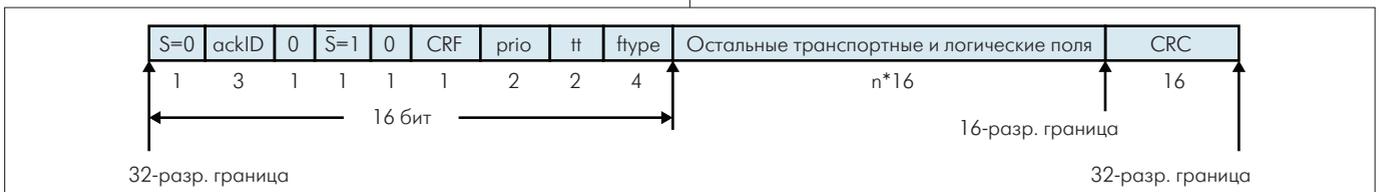


Рис. 5. Формат параллельного потока пакетов с полем CRC



интерфейс (однополосное звено 1x) физического уровня между устройствами, использующими однонаправленные дифференциальные сигналы в двух направлениях. Стандарт допускает также объединение четырех таких последовательных звеньев (4x) для увеличения производительности передачи, требуемое современными приложениями. Определяется также протокол, управляющий этим звеном и транспортировкой пакетов между последовательными устройствами RapidIO.

Спецификация 1x/4x LP-Serial физического уровня имеет следующие особенности:

- использует синхронизируемую таймером схему кодирования данных типа 8В/10В;
- поддерживает в каждом направлении одну последовательную дифференциальную пару, ссылаясь на нее, как на одну полосу (lane), или четыре такие (объединенные) пары, ссылаясь на четыре полосы;
- допускает коммутацию пакетов между последовательными портами 1x/4x LP-Serial и системой RapidIO;
- реализует аналогичные параллельной передаче физического уровня протоколы повтора передачи и восстановление сигнала, переданного с ошибкой;
- поддерживает (в расчете на одну полосу) скорости передачи: 1,25; 2,5 and 3,125 Гбит/с (скорости данных: 1,0; 2,0 и 2,5 Гбит/с, соответственно);
- обеспечивает агрегатную скорость передачи данных до 10 Гбит/с при 4-полосной передаче (sRIO link).

Спецификация [2e] описывает сначала отдельные элементы протокола связи: пакеты, символы управления и схему кодирования. Затем она описывает формат пакетов, протокол связи, управление, регистры, сигналы и электрические спецификации.

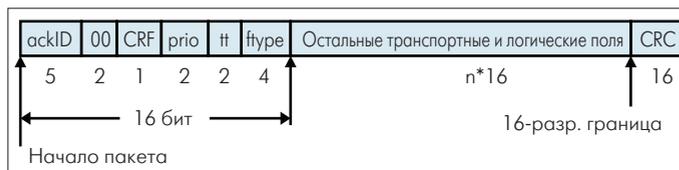


Рис. 6. Формат последовательного потока пакетов с полем CRC

Формат пакета

Формат пакетов при последовательной передаче (см. рис. 6) состоит из 10-битного заголовка физического уровня в начале пакета с полями "ackID" (5 бит), "rsrv" (2 бита "00"), "CRF" (1 бит) и "prio" (2 бита) (расшифровка аналогична предыдущему); заголовков транспортного ("tt" – 2 бита) и логического ("ftype" – 4 бита) уровней; остальных транспортных и логических полей и поля "CRC-16" (16 бит) в конце пакета.

Как и пакеты при параллельной передаче, пакеты при последовательной передаче должны быть целократными полю в 32 бита, что упрощает логику портов, ширина которых также 32 бита. Однако пакеты, определенные в этой спецификации, как и в спецификациях транспортного и логического уровней, целократны 16 битам. Поэтому, если общая длина пакета со-

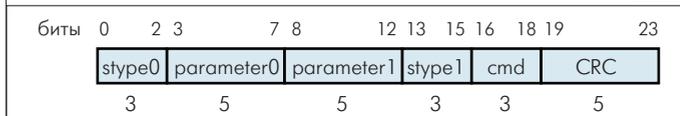


Рис. 7. Формат символа, управляющего функциями обработки при передаче пакета

стоит из нечетного числа 16-битных полей, то в конце его помещается 16-битная пустая вставка (0x0000), чтобы сделать длину пакета кратной 32 битам.

Поле CRC в конце пакета используется для обнаружения ошибок при передаче пакета, причем первые 6 бит ("ackID" и "rsrv") остаются без защиты CRC и защищены только протоколом связи.

Максимальная длина пакета при последовательной передаче составляет 276 байтов, а переносимая полезная нагрузка – 256 байтов. Поэтому, если длина пакета (без поля CRC) превышает 80 байтов, то используется второе поле CRC-16 в конце пакета, дополняемое при необходимости пустым полем длиной до 16 бит. При этом общий кадр пакета состоит из тех же полей, что и при параллельной передаче.

Формат управляющего символа

Управляющие символы – это элементы сообщений, используемых портами, связанными последовательным звеном, для управления всеми этапами его работы. С их помощью управляют звеном, сообщениями об ошибках, устанавливают границы пакетов и устраняют ошибки.

На рис. 7 показан формат символа, управляющего обработкой пакета передачи в зависимости от состояния (т.е. принятого с ошибкой или без нее). Этот символ имеет длину три байта, а его формат состоит из 6 полей:

- stype0 – код управляющих символов (8 типов), отражающий состояние пакета и позволяющий использовать "parameter0" и "parameter1" (3 бита);
- parameter0/parameter1 – параметры, определяются по специальной таблице в зависимости от кода "stype0" (5 бит);
- stype1 – код управляющих символов (8 типов), отражающий состояние пакета и позволяющий использовать поле "cmd" (3 бита);
- cmd – команда, определяется по специальной таблице в зависимости от кода "stype1" (3 бита);
- CRC – CRC-5 – пятибитный код обнаружения ошибок при передаче символов (5 бит).

В соответствии с рис. 7 и используя коды полей "stype0" и "stype1", можно сформировать 12 различных символов, управляющих обработкой пакетов в различных ситуациях, например при подтверждении и неподтверждении приема, при повторной передаче, формировании статуса, фиксации отклика звена, при установлении начала и конца пакета и т.д.

Подуровни PCS и PMA физического уровня

Физический уровень использует код 8В/10В и два подуровня – PCS и PMA аналогично гигабитному и 10-гигабитному

Ethernet. Функция подуровня PCS отвечает за генерацию пустых последовательностей, формирование (мультиплексирование), выравнивание и расформирование (демультиплексирование) полос (операции 1x и 4x), кодирование и декодирование, используя код 8В/10В. Полученные в результате 10-битные кодовые символы передаются в виде 10-битной параллельной кодовой группы на подуровень PMA. При формировании потока (звена) 4x на подуровне PCS происходит мультиплексирование четырех полос с символьным интерливингом [3] кодовых символов. Функция подуровня PMA преобразует параллельную 10-битную кодовую группу обратно в непрерывный последовательный поток кодовых символов, принадлежащих соответствующей полосе (т.е. осуществляет операцию демультиплексирования).

Код 8В/10В. Кодирование 8В/10В сводится к установлению (с помощью таблицы) соответствия между 8-битными (8 информационных бит) и 10-битными кодовыми символами. При этом 8 бит разбиваются на две группы: 5 LSB ("abcde") и 3 MSB ("fgh"). Каждая из них кодируется по таблице (кодом, сбалансированным по "0" и "1") таким же количеством кодовых бит (5 и 3), к которым добавляется по одному (вычисляемому) биту (*i* и *j*) *текущего диспаритета* (RD). Получается 10-битный кодовый символ ("abcdei"+"fghj"). При декодировании, напротив, из 1024 возможных блоков выделяются 256 8-битных блоков данных и 12 специальных управляющих символов. Код позволяет выделять сигнал синхронизации на приеме и выявлять одиночные и некоторые множественные ошибки. Кодовая последовательность для одной полосы передачи зависит от того, что передается: управляющие символы или пакеты данных. Если это блоки данных, то поток состоит из блоков, составленных из трех 10-битных кодов информационных данных. Если передается управляющий символ, содержащий ограничитель пакета, то добавляется спецсимвол PD, а если ограничителя пакета нет, то добавляется спецсимвол SC. Использование спецсимволов обеспечивает посылку приемнику "раннего предупреждения" о передаче управляющего символа.

Последовательный LP-протокол

LP-протокол реализован в виде *протокола обмена пакетами* между устройствами, соединенными последовательным LP-звеном. В нем каждый пакет, передаваемый одним устройством, квитируется с помощью управляющих символов, передаваемых с помощью другого устройства. Если пакет не может быть принят, то управляющий символ указывает в процессе квитирования не только причину, но и то, что этот пакет, а также другие пакеты, составляющие с ним единое целое, должны быть переданы повторно. Этот процесс создает механизм контроля пакета и управления ошибками между соединенными ОЭ.

На рис. 8 показан пример прохождения пары пакетов запроса-отклика через коммутационную структуру взаимосвязи с подтверждениями между парами приемник-передатчик вдоль

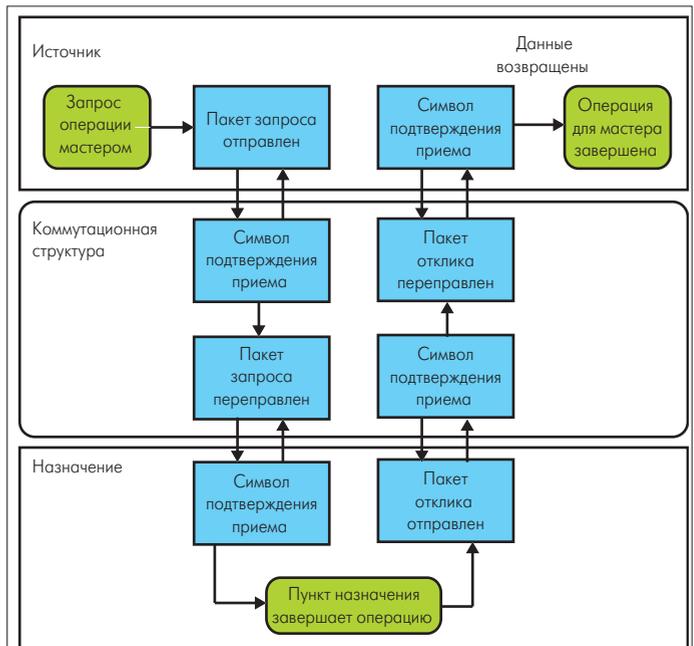


Рис.8. Пример прохождения пары пакет запроса – пакет отклика через коммутационную структуру взаимосвязи

тракта передачи. Это позволяет обслуживать управление потоком и обработку ошибок между каждой электрически соединенной парой устройств, а не между источником и пунктом назначения данной транзакции. Устройство в конечной точке должно передать символ подтверждения принятия пакета запроса до передачи пакета отклика, соответствующего этому запросу.

Процесс инициализации.

Передача управляющих символов

После инициализации последовательного LP-порта через него нужно прогнать без ошибок 15 пустых последовательностей, состоящих из 1024 кодовых групп, пока от соединенного с ним порта не будет получен управляющий символ со статусом "ошибок нет". Символов с таким статусом должно быть получено 7, прежде чем начнется нормальное функционирование. После этого, через каждые переданные 1024 кодовые группы должен передаваться управляющий символ "buf_status", контролирующий число доступных пакетных буферов в принимающих устройствах.

Передача пакетов. Пакеты для передачи ограничиваются управляющими символами. Так как пакеты имеют переменную длину, требуются два ограничителя: "Начало пакета" и "Конец пакета". Кроме указанных используются еще три ограничителя: "Подавить" (Stomp), "Начать после повтора" и "Запросить любое звено". Все эти пять символов управления можно использовать для терминирования пакета.

Приоритеты пакетов. Каждому пакету присваивают один из четырех (0,1,2,3) приоритетов (0 – минимальный, 3 – максимальный), помещаемый в поле "prio". Он используется для упорядочения транзакций и предотвращения тупиковых ситуаций. Если поддерживается поле "CRF", то можно дополнительно



но выделить критичные (важные) потоки среди потоков, имеющих одинаковый приоритет.

Ликвидация пакетов. Если порт требует повторной передачи пакета или получил управляющие символы, требующие терминирования пакета, то текущий пакет ликвидируется, чтобы исключить незавершенность операции передачи. При этом не генерируется какой-то ошибки. Если же пакет ликвидируется в результате получения передающей стороной управляющего символа "Пакет не принят", то генерируется "стандартная" ошибка.

Тупиковые ситуации. Ситуация, в которой два ОЭ (использующие одни и те же ресурсы) препятствуют друг другу в доступе к этим ресурсам, приводя тем самым к остановке операций системы, называется *тупиковой* (deadlock). Для преодоления таких ситуаций пакет отклика всегда имеет более высокий начальный приоритет, чем приоритет связанного с ним пакета запроса, а конечный ОЭ может еще и повышать приоритет пакета отклика (более подробно о правилах преодоления тупиковых ситуаций см. в стандарте [2e]).

Поврежденные пакеты. Входные порты считают поврежденными все пакеты с нестандартным значением "ackID" или с символами, имеющими некорректные значения CRC, а также при выходе пакета за установленные границы (например, при превышении допустимой длины данных).

СПЕЦИФИКАЦИЯ ТРАНСПОРТНОГО УРОВНЯ

Спецификация транспортного уровня [2в] определяет основной транспортный механизм передачи: транспортный заголовок, добавляемый к пакету, сформированному на логическом уровне RapidIO, и то, как он интерпретируется коммутационной структурой. Этот механизм формируется независимо от того, что реализовано на физическом уровне. Его особенности определяются требованиями к физической реализации, такими как уровни сигнализации ввода-вывода, топология соединения, протокол физического уровня и обнаружение ошибок, приведенными в спецификации физического уровня.

Транспортная спецификация не зависит от спецификации физического уровня и имеет следующие особенности.

- Система RapidIO имеет единую транспортную спецификацию, что гарантирует совместимость различных реализаций.
- Транспортная спецификация является гибкой настолько, чтобы адаптироваться к будущим приложениям.
- Предполагается, что пакеты направляются от одного источника к единственному пункту назначения, однако это требование не обязательно.
- Операции бродкастинга и мультикастинга можно осуществить путем интерпретации транспортной информации коммутационной структуры.

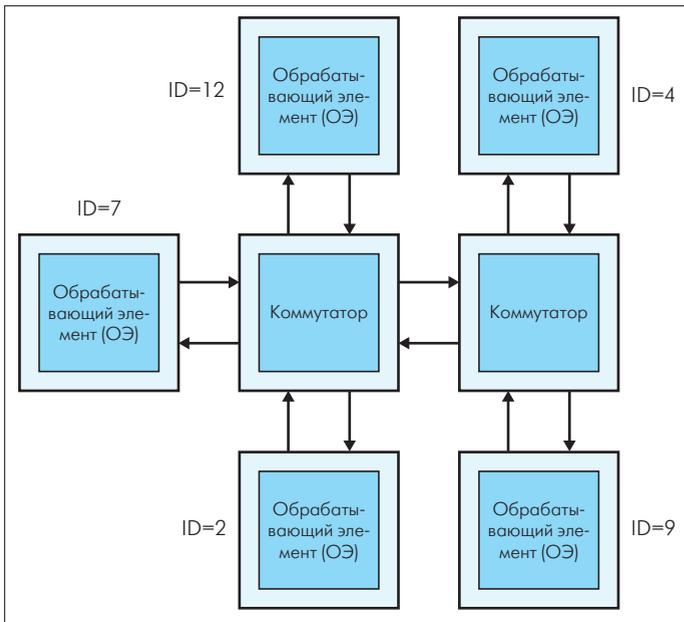


Рис. 9. Схема топологии системы с двумя коммутаторами

- Определения транспортных механизмов не зависят от ширины физических интерфейсов между устройствами коммутационной структуры.
- В системе RapidIO не существует требований "географической" привязки адреса, идентификаторы устройств не зависят от их места на адресной карте, но могут быть назначены при использовании других средств.

Системные топологии

Ниже рассмотрены две системные топологии: одна из них построена на основе двух коммутаторов (рис. 9), соединяющих, например, пять ОЭ, вторая – имеет вид кольца, соединяющего те же пять ОЭ (рис. 10).

При использовании коммутаторов пакет (имеющий уникальный ID), посланный от одного ОЭ к другому, маршрутизируется с помощью коммутаторов, которые используют информацию транспортных полей.

Для кольца схема упрощается, так как уникальные ID связываемых элементов делают ее похожей на схему "точка-точка".

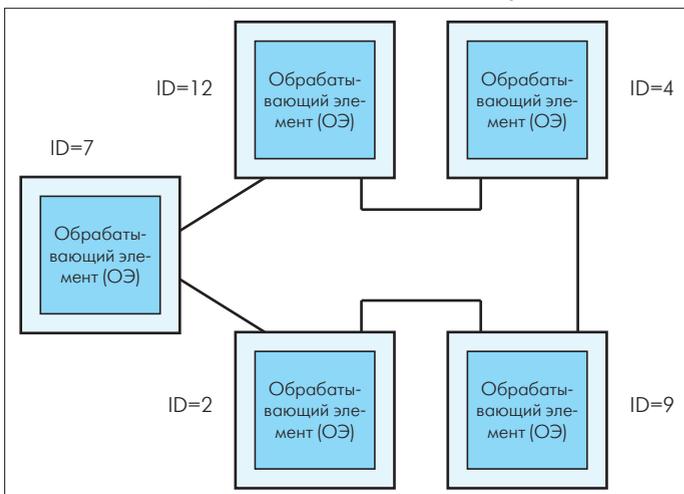


Рис. 10. Схема системы с кольцевой топологией

Пакет в кольце должен иметь ID источника и назначения. При его движении по кольцу ОЭ считывают его адрес назначения и забирают себе, если он совпадает с их адресом, или пропускают дальше, если не совпадает.

Маршрутизация пакетов

Для маршрутизации пакетов используются ID. Каждое адресуемое устройство может иметь один или больше ID. При генерации пакета в его заголовок помещают как ID места назначения, так и источника, чтобы знать, куда посылать отклик на запрос. При формировании отклика адрес назначения и адрес источника меняются местами. Если применяется схема с коммутаторами, то для маршрутизации используется таблица маршрутизации (как в локальных сетях): либо статическая (один путь следования для пакета), либо динамическая (путь следования выбирается в зависимости от состояния сети передачи). Вариант передачи зависит от реализации коммутационной системы.

Любой ID устройства может быть использован как одиночный ID (при передаче типа "точка-точка", т.е. при уникастинге) или как групповой ID (при передаче типа "точка-многоточка", т.е. при мультикастинге). Пакеты уникастинга направляются на выход одиночного порта, а пакеты мультикастинга на выход порта, который является корнем дерева, соединенным с группой узлов-листьев.

Транспортные поля

Транспортный уровень описывает и вставляет в заголовок три поля: поле "tt" (2 бита), поля ID назначения и ID источника. Последние два поля могут быть большими (16 бит) или малыми (8 бит). Малые поля допускают адресацию максимум 256 устройств, большие – 65536 устройств. Схема расположения транспортных полей в заголовке пакета логического уровня приведена на рис.11. Крайним левым будет транспортное поле tt, затем поле заголовка логического уровня ftуре, а затем два других транспортных поля: ID назначения и ID источника и остальная часть пакета логического уровня. Учитывая, что коммутаторы (как и ОЭ) не имеют ID, но их идентификация важна, используется дополнительное четвертое поле "hop_count" (8 бит), указывающее число скачков (от коммутатора к коммутатору) в процессе маршрутизации, и алгоритм его обработки, позволяющие четко идентифицировать данный коммутатор.

Регистры транспортного уровня

Транспортный уровень использует два класса регистров, имеющих у каждого ОЭ: CAR и CSR. Регистры CAR хранят данные о возможностях ОЭ в обслуживании операций ввода/вывода. Регистры CSR хранят данные о командах и статусе. Содержимое этих регистров может быть считано другими внешними ОЭ. Длина регистров – 32 разряда.



Рис. 11. Схема транспортных полей в формате пакета

СПЕЦИФИКАЦИИ ЛОГИЧЕСКОГО УРОВНЯ

Спецификации логического уровня не предполагают наличия каких-то транспортных или физических интерфейсов и оперируют форматами потоков бит, добавляемых в логический заголовок пакета. Эти спецификации описаны в шести стандартах [2а, 2б, 2д, 2и, 2к, 2л].

Спецификация ввода-вывода логического уровня [2а] описывает логическую взаимосвязь протоколов и форматов пакетов и определяет набор операций, необходимых для обработки транзакций. Спецификация [2б] описывает программную модель передачи сообщений, которая делает возможным обработку распределенного ввода-вывода. Спецификация [2д] описывает распределенную общую память GSM с отображением на ней устройств ввода-вывода. Однако поддержка кэш-памяти и модели когерентности памяти отданы на откуп конечным устройствам, например МП. Спецификация [2и] описывает управление потоком параллельной и последовательной передачи, а также перегрузки и механизмы управления ими. Спецификация [2к] была разработана для повышения эффективности, гибкости и увеличения независимости протоколов с целью минимизировать ресурсы, необходимые для передачи данных, и сохранить совместимость и полноту взаимодействия с остальными спецификациями RapidIO. Наконец, спецификация [2л] описывает расширение стандарта [2], позволяющее обрабатывать потоки мультикастинга, соответствующие топологии "точка-многоточка".

Логический уровень. Спецификация ввода-вывода Транзакции

Протоколы ввода-вывода [2а] используют пары транзакций типа "запрос-ответ". Один ОЭ посылает запрос другому ОЭ, который возвращает отклик запрашивающему ОЭ независимо от того, запрос принят без ошибки или с ошибкой. Транзакция посылается в виде пакета. Например, ОЭ, требующий данные от другого ОЭ, посылает транзакцию NREAD в пакете запроса этому ОЭ, который читает свою локальную память по запрошенному адресу и возвращает данные запрашивающему ОЭ в виде транзакции DONE в пакете отклика. Заметим, что не все запросы требуют откликов, некоторые из них предполагают, что отклик не требуется.

Два возможных типа транзакции отклика могут быть получены запрашивающим ОЭ:

- отклик DONE, указывающий, что желаемая транзакция закончена. Если транзакция запроса была типа NREAD (чтение), то с откликом возвращаются также и данные;
- отклик ERROR, означающий, что цель транзакции не может быть достигнута из-за ошибки.

Пакеты могут содержать дополнительную информацию, которая интерпретируется структурой взаимосвязи, маршрутизирующей пакеты по сети структуры от источника до назначения (например, устройства с таким-то номером). Эти требования описаны в соответствующей спецификации транспортного уровня. Могут генерироваться и другие пакеты, например при обработке функций протокола физического уровня по управлению потоком данных или при обработке ошибок.

Для многих типов транзакций транзакция-запрос, посланная в систему, маркируется уникальным ID для каждой пары взаимодействующих ОЭ. По этому ID легко определить, кому послать отклик. Оконечное устройство не может использовать такое ID при отправке отклика, пока от предполагаемого назначения не будет получен запрос. ID транзакций могут указывать, что послана информационная последовательность, если требуется упорядоченный прием транзакций для ОЭ назначения (учитывая, что структура взаимосвязи может изменить последовательность пакетов). В этом случае приемное устройство должно принять, но не завершать прием последовательности, если произошла ошибка в номере следования пакета, пока потерянная из последовательности транзакция не будет получена, а информационная последовательность в целом не будет завершена.

Операции ввода-вывода

Ниже приведен список операций ввода-вывода, определенных в RapidIO и их использование.

- NREAD, RESPONSE – считывание данных из памяти по адресу, указанному в запросе ОЭ, с откликом, т.е. доставкой данных требуемого формата;
- NWRITE – запись данных по адресу, указанному в запросе ОЭ, без отклика; форматы данных: двойное слово, слово, полуслово и байт с выравниванием по границам двойного слова;
- NWRITE_R, RESPONSE – запись данных, устойчивая к сбоям, с откликом из точки назначения;
- SWRITE – запись данных в потоке без отклика, данные имеют формат только двойного слова, SWRITE – версия NWRITE с меньшим заголовком;
- ATOMIC, RESPONSE – атомарная (неделимая) операция: считывание-модификация-запись, данные могут иметь форматы слова, полуслова и байта;
- MAINTENANCE – инициализация, конфигурация, исследование и обслуживание системы; в этой операции ОЭ использует регистры CAR, CSR или другие местные регистры для записи и считывания данных; поддерживаемые форматы 8 и

4 байта, кроме откликов операций записи и считывания, могут использовать MAINTENANCE-отклик.

Кроме указанных операций возможны операции загрузки памяти (Load), операции хранения данных (Store) и операции прямого доступа в память (DMA).

Все операции ввода-вывода данных проводятся с указанием требуемого формата данных, однако, если он меньше 8 байт, то данные выравниваются по формату двойного слова. На рис. 12 показано двойное слово с MSB в "0" байте, а также выравнивание слова (байты 4–7) и полуслова (байты 2–3).



Рис. 12. Схема выравнивания по двойному слову

Упорядочение операций

Хотя большинство операций ввода/вывода не налагают особых требований на порядок их завершения, существует ряд задач, требующих особого порядка выполнения операций. Например, ОЭ может потребовать осуществить операцию записи в ряд регистров другого ОЭ. Последовательность, в которой эти записи проводятся, может быть критичной для этого другого элемента.

Поэтому нужны определенные правила, выполнение которых могло бы гарантировать упорядоченное завершение этих операций, что важно при выполнении операций между определенными парами источник-назначение.

Формат пакетов. Существуют четыре типа форматов ввода-вывода: форматы запроса, отклика, резервный и формат, определенный при реализации системы. Резервный формат до использования должен быть определен в другой логической спецификации.

Пакет запроса формально может быть 12 (0–11) типов, но типы 1, 3–4, 7 и 9–11 – резервные, тип 0 – зависит от реализации; тип 2 в двух модификациях – ATOMIC и NREAD, тип 5 – ATOMIC, NWRITE, NWRITE_R, тип 6 – SWRITE, тип 8 – MAINTENANCE. Пакет имеет различные виды заголовков в зависимости от его типа и может иметь до 10 полей логического уровня.

Размер адреса может быть разным. Базовый адрес составляет 32, 48 и 64 бита. С помощью поля "xamsbs" в пакете запроса он может быть расширен на 2 бита (MSB), что дает адреса длиной 34, 50 и 66 байтов.

Пакет отклика может быть четырех типов (12–15), из них два резервных (12 и 14), тип 15 зависит от реализации. Единственным рабочим является тип 13 – RESPONSE. Пакет имеет заголовок с четырьмя полями логического уровня общей длины 20 бит.

Регистры логического уровня

Логический уровень, как и транспортный, использует два класса регистров, имеющихся у каждого ОЭ: CAR и CSR, у ко-

СПИСОК СОКРАЩЕНИЙ

- API** – Application Program/Programming Interface – интерфейс прикладного программирования. Набор системных модулей, используемых прикладными программами для выполнения сервисных функций, реализуемых действующей ОС.
- ATCA** – (AdvancedTCA) – Advanced Telecommunication Computing Architecture – усовершенствованная архитектура телекоммуникационных вычислений. Стандарт на телекоммуникационное оборудование следующего поколения, разработанного Консорциумом PICMG. Стандарт покрывает разработку конструкции шасси, обеспечение питания и охлаждения, взаимодействие блоков и организацию интерфейсов.
- CRC-n (CRCn)** – n-bit Cyclic Redundancy Check (pattern) – n-битная проверка на наличие ошибок, использующая n избыточных бит полиномиального циклического кода, добавляемых в виде последовательности для проверки кадра (FCS) к проверяемой на приеме последовательности. n – Целое число (порядок полинома), напр., 16. FCS вычисляется при передаче кадра.
- COTS** – Commercial Off-The-Shelf – готовый коммерческий продукт (ПО, решение).
- CSIX** – Common Switch Interchange – общий обмен между коммутаторами. Протокол, определенный в стандарте PICMG 2.20 для усиления вклада поставщиков коммутационных структур.

- DSLAM** – Digital Subscriber Line Access Multiplexer – мультиплексор доступа цифровых абонентских линий.
- LVDS** – Low Voltage Differential Signaling – дифференциальная сигнализация с пониженным напряжением. Техника передачи сигналов по дифференциальной паре с пониженным напряжением.
- MTU** – Maximum Transfer/Transmission Unit – максимальный размер передаваемого блока.
- NPSI** – Network Packet Switching Interface – сетевой интерфейс пакетной коммутации.
- OSI** – Open Systems Interconnection – (эталонная модель) взаимодействия открытых систем (ВОС, МВОС или ЭМВОС). Семьюуровневая эталонная модель, разработанная ISO (ISO-7498) и принятая ИТУ-T (X.200) для определения иерархии интерфейсов в системах связи.
- PCS** – Physical Coding Sublayer – подуровень кодирования физического уровня. Верхний подуровень физического уровня PHY. Расположен над подуровнями WIS и/или PMA, но ниже интерфейса XGMII или XMIИ, в модели OSI технологии Ethernet для скоростей 100 Мбит/с.
- PMA** – Physical Medium Attachment – подуровень подсоединения к физической среде (передачи) – средний подуровень физического уровня PHY, расположен над подуровнем PMD, но ниже подуровня WIS и/или PCS, в модели OSI технологии Ethernet для скоростей 100 Мбит/с.
- QOS (QoS)** – Quality Of Service - качество обслуживания.



торых – то же функциональное назначение, возможности считывания и длина. Они доступны при использовании операций обслуживания.

Логический уровень.

Программная модель передачи сообщений

Данная модель имеет следующие особенности [26].

- RapidIO поддерживает как программную модель передачи сообщений (с программной когерентностью), так и традиционную компьютерную программную модель глобальной общей памяти.
- Разные размеры системы (от самой малой до очень большой) поддерживаются теми же самыми или совместимыми форматами пакетов.
- RapidIO поддерживает возможность инкапсуляции данных различного размера в формат пакета.
- В RapidIO предполагается, что если данная область ЛП доступна локальному устройству, то удаленное устройство может получить доступ к этой же области памяти только через интерфейс передачи сообщения, контролируемый локальным устройством.
- Протоколы и форматы пакетов не зависят от физической топологии связи, а протоколы работают на топологиях "точка-точка", "кольцо", "шина", "коммутируемая многомерная сеть", "последовательное дуплексное соединение" и т.д.

- Протоколы обрабатывают прием и передачу неупорядоченных данных.

Формат и передача сообщений

Модель передачи сообщений широко используется в компьютерах с распределенной памятью. В этой модели ОЭ имеют доступ только к ЛП. ОЭ взаимодействуют с помощью специального аппаратного обеспечения (АО), работающего контролем приложений. Если ОЭ-А (см. рис.2) хочет установить связь с ОЭ-В, то прикладное ПО обращается (с помощью ОС-вызова или библиотеки API) к АО передачи сообщений (АО почтового ящика – АО-ПЯ) и с его помощью формирует и посылает сообщение для ОЭ-В, АО-ПЯ которого принимает сообщение и сохраняет его в своей ЛП, информируя об этом ОЭ-В. После чего оно может считать полученное сообщение.

Сообщение представляет собой *дейтаграмму* длиной до 4096 байтов. Часто оно велико для передачи в одном пакете, и его разбивают на ряд *пакетов-сегментов*, которые при приеме автоматически упорядочиваются. Одновременно можно передавать несколько сообщений. В рамках одного сообщения можно генерировать до 16 пакетов, содержащих до 256 байтов данных полезной нагрузки. При этом используются ряд форматов данных разной длины (см. выше). Все или 15 из 16 пакетов при этом должны быть одинаковой длины, кроме последнего пакета, который может быть меньше. Пакеты форматируются с тре-

мя полями: размер полезной нагрузки пакетов (кроме последнего), размер поля полезной нагрузки последнего пакета и порядок следования.

Письма и почтовые ящики. Для сообщений с несколькими пакетами поля "письмо" и "почтовый ящик" (ПЯ) позволяют одновременно обрабатывать до четырех сообщений (или писем) для четырех разных почтовых ящиков (т.е. до 16 писем одновременно). Поле ПЯ можно использовать для указания приоритета сообщения (0 – старший, затем 1, 2 и 3) или как ID в случае использования нескольких адресов для ПЯ. Для сообщений из одного пакета поля "письмо" и ПЯ позволяют одновременно посылать/обрабатывать четыре письма для 64 разных ПЯ (с приоритетами 0 – 63). Один и тот же ОЭ может получать сообщения в двух ПЯ по два письма в каждом.

Короткие извещения. RapidIO поддерживает также очень короткие (8-16-битные) сообщения, или *извещения* (doorbells). Они позволяют переслать небольшую порцию информации для ОЭ (МП), содержащуюся в заголовке пакета, не имеющего полезной нагрузки. Формирование, посылка и прием извещений и сообщений аналогичен вышеизложенному (см. также Дополнение к стандартам [2м]).

Доставка транзакций и сообщений. Существуют две основные схемы доставки транзакций: неупорядоченная и упорядоченная. Протокол RapidIO предполагает, что большинство транзакций доставляются по неупорядоченной схеме, однако он же дает возможность организации полностью упорядоченной доставки. Операции с сообщениями могут состоять из нескольких транзакций, которые могут достичь ПЯ назначения в произвольном порядке. Однако эти транзакции содержат явные информационные теги, позволяющие реконструировать данное сообщение с помощью ОЭ назначения.

Операции передачи сообщений

Протокол передачи сообщений RapidIO поддерживает ряд операций и пары транзакций запроса-отклика, описанных выше. Например, ОЭ, которому нужно послать сообщение другому ОЭ, посылает пакет запроса сообщения типа MESSAGE этому ОЭ, который обработает его и вернет посылающему пакет отклика DONE. Три возможных транзакции отклика может получить запрашивающий ОЭ:

- отклик типа DONE. Указывает, что желаемая транзакция завершена;
- отклик типа RETRY. Генерируется транзакцией сообщения, которая пыталась достичь ПЯ, занятого обслуживанием другого сообщения, или извещением, которое столкнулось с занятым АО извещения. Транзакция запроса, получившая в ответ отклик RETRY, должна быть передана повторно для завершения операции;
- отклик типа ERROR означает, что случилась непоправимая ошибка, не давшая завершить транзакцию. Эти ситуации описываются в соответствующей спецификации транспорт-

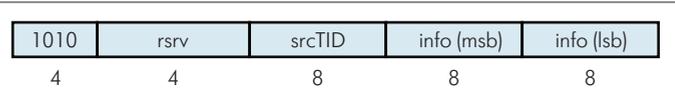


Рис. 13. Схема логических полей в формате уведомления

ного уровня [2з]. Некоторые ошибки могут обрабатываться и в рамках протокола физического уровня, управляющего потоком пакетов [2е]. Каждая транзакция маркируется своим ID, уникальным для пары запрос/ответ. ID позволяет отклику легко найти нужный запрос для возврата отклика. Конечная точка не может повторно использовать этот ID для того же самого назначения, пока не получит отклик на исходную транзакцию.

Два типа транзакций передачи сообщения определены в этой части логической спецификации:

- *извещение* – состоит из транзакций DOORBELL и RESPONSE (как правило, типа DONE) и используется для посылки очень короткого сообщения в информационном поле (определяемом ПО) транзакции DOORBELL, не имеющей полезной нагрузки. ОЭ, получивший транзакцию DOORBELL, помещает ее в очередь, реализованную программно или аппаратно, для последующего считывания;
- *сообщение данных* – состоит из транзакций MESSAGE и RESPONSE (как правило, типа DONE) и используется для посылки сообщения данных другому ОЭ. Оно может состоять из 16 транзакций типа MESSAGE, полезная нагрузка которых всегда кратна двойному слову.

На приемной стороне АО анализирует ряд полей этого сообщения данных, чтобы записать отдельные пакеты данных сообщения MESSAGE в ЛП, а именно:

- msglen – указатель длины сообщения, определяющий число транзакций, объединенных в этом сообщении;
- msgseg – указатель сегмента, определяющий, какой пакет данных принимаемого сообщения содержится в данной транзакции;
- mbox – указатель ПЯ, определяющего, какому ПЯ адресовано данное сообщение;
- letter – указатель номера письма, позволяющий получить квитанцию на получение ряда писем одним и тем же ПЯ от одного и того же источника;
- ssize – указатель стандартного формата длины данных, определяющего (возможно за исключением последней транзакции) стандартный формат (размер) данных в сообщении.

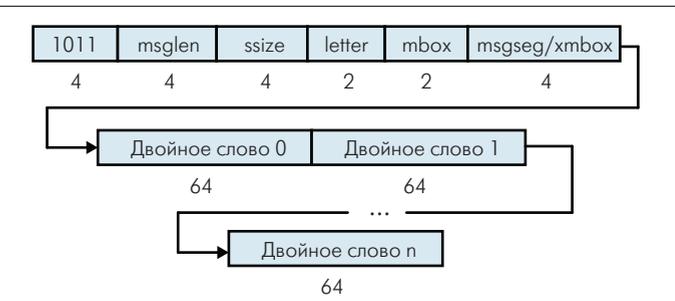


Рис. 14. Схема логических полей в формате сообщения данных



На рис.13 и 14 приведены схемы расположения логических полей в форматах "уведомление" и "сообщение данных". Кроме этих полей, в формате "уведомление" используются два поля "info(msb)" и "info(lsb)" длиной по 1 байту, определяющие возможность передачи сообщений длиной 1 или 2 байта. В формате "сообщение данных" вместо поля "msgseg" может использоваться указатель номера ПЯ "xmbx" (от 0 до 63), дополнительно определяющего четыре старших бита для номера ПЯ.

Продолжение в следующем номере

ЛИТЕРАТУРА

1. **Рыбаков А., Слепов Н.** Компьютерные встраиваемые технологии – тенденции развития. – ЭЛЕКТРОНИКА: НТБ, 2006, №3, с.24–32; №4, с.44–49.
2. **RapidIO™ Interconnect Specification.** RapidIO Trade Association. 2005.
 - а – Part 1: Input/Output Logical Specification. Rev. 1.3, 6.05.
 - б – Part 2: Message Passing Logical Specification. Rev. 1.3, 6.05.
 - в – Part 3: Common Transport Specification. Rev. 1.3, 6.05.
 - г – Part 4: Physical Layer 8/16 LP-LVDS Specification. Rev. 1.3, 6.05.
 - д – Part 5: Globally Shared Memory Logical Specification. Rev. 1.3, 6.05.
 - е – Part 6: 1x/4x LP-Serial Physical Layer Specification. Rev. 1.3, 6.05.
 - ж – Part 7: System and Device Inter-operability Specification. Rev. 1.3, 6.05.
 - з – Part 8: Error Management Extensions Specification. Rev. 1.3, 6.05.
 - и – Part 9: Flow Control Logical Layer Extensions Specification. Rev. 1.3, 6.05.
 - к – Part 10: Data Streaming Logical Specification. Rev. 1.3.a., 6.05.
 - л – Part 11: Multicast Extensions Specification. Rev. 1.3.1., 6.05.
 - м – Annex 1: Software/System Bring Up Specification. Rev. 1.3, 6.05.
3. **Слепов Н.Н.** Современные технологии цифровых оптоволоконных сетей связи. – 2-е изд. исправл. – М.: Радио и связь, 2003. – 468 с.
4. **Strod E.C.** RapidIO fabric unifies VITA and PICMG platform architecture. – VMEbus Systems, Feb. 2006.
5. **RapidIO™ Technology Overview and Applications.** Rev. 03, May 2005. RapidIO Trade Association 2005.