

# ПОСЛЕДОВАТЕЛЬНЫЙ ИНТЕРФЕЙС RapidIO И ЕГО ПРИМЕНЕНИЕ В ПАКЕТНОЙ КОММУТАЦИИ

ВЛАДИМИР ЕГОРОВ, к.т.н., и.о. в.н.с., Институт проблем информатики РАН

*В статье рассмотрены особенности применения последовательного интерфейса RapidIO (SRIO) в пакетной коммутации. Описаны механизм передачи сообщений, построение транспортной среды и характеристики некоторых коммутаторов интерфейса SRIO от ведущих мировых производителей. Обсуждаются вопросы реализации интерфейса SRIO в интегрированных коммуникационных микроконтроллерах.*

## ВВЕДЕНИЕ

Интерфейс «быстрого ввода-вывода» — RapidIO — это современный интерфейс мультипроцессорных систем с высокой скоростью передачи данных и удобным протоколом связи между равноправными абонентами с минимальными программными издержками. Он используется для коммуникаций как между чипами в пределах одной платы, так и между платами в пределах устройства.

Спецификация интерфейса RapidIO охватывает три уровня:

- логический, определяющий форматы и структуру пакетов, обеспечивающих передачу данных и сообщений между абонентами интерфейса;
- транспортный, несущий ответственность за идентификацию абонентов и доставку пакетов от абонента к абоненту через транспортные среды;
- физический, отвечающий за электрические параметры сигналов, установление соединения между абонентами и контроль передач на звене интерфейса.

Логический уровень включает три сервиса:

- адресный доступ, аналогичный другим компьютерным интерфейсам, в частности PCI и PCI Express и предоставляющий абоненту доступ к памяти;
- обмен сообщениями, позволяющий абстрагироваться от физических адресов памяти и передавать данные между абонентами на сетевых принципах;
- потоковая (streaming) передача блоков данных размером до 64 Кбайт с возможностью инкапсуляции пакетов других протоколов.

Физический уровень интерфейса RapidIO, в свою очередь, имеет два варианта реализации:

- параллельный, основанный на классической 8- или 16-разрядной шине, работающей на тактовых частотах 250 МГц...1 ГГц;
- последовательный дуплексный, реализуемый парафазными линиями с

низкими уровнями сигналов и агрегатными скоростями передачи данных до 10 Гбит/с.

Параллельный интерфейс RapidIO не нашел широкого применения, зато все большую популярность приобретает его последовательный вариант (serial RapidIO — SRIO). Интерфейс SRIO нашел широкое применение в процессорах обработки сигналов (ПОС) во многом благодаря тому, что одним из инициаторов его создания и внедрения была компания Texas Instruments — признанный лидер на рынке ПОС. Другой ведущий поставщик ПОС, компания Freescale Semiconductor, стал активно применять интерфейс SRIO в своих интегрированных коммуникационных микроконтроллерах (ИКМ). Таким образом, определились две основные области применения интерфейса SRIO: цифровая обработка сигналов и телекоммуникация с пакетной коммутацией.

В цифровой обработке сигналов интерфейс SRIO используется для объединения множества ПОС и подключения их к хост-процессорам. Областью массового применения ПОС стала бурно развивающаяся беспроводная связь. Базовые станции мобильных телесистем преобразуют аналоговые потоки из эфира в цифровые пакетные потоки сжатой речи, и наоборот. Для такой работы используются большие агрегации ПОС, между которыми необходимо организовать распределение и обеспечить своевременную доставку обрабатываемых данных в режиме реального времени. Интерфейс SRIO как нельзя лучше подходит для решения такого рода задач. Например, одно звено SRIO способно пропустить восемь цифровых потоков телевидения высокого разрешения HD1080i или, соответственно, четыре потока HD1080p.

Аналогичная картина наблюдается в области телекоммуникаций и пакетной коммутации. Современные ИКМ поддерживают, как правило, несколько телекоммуникационных интерфейсов, но

число их в одном приборе ограничено количеством выводов корпуса, производительностью встроенных процессоров, максимальной потребляемой мощностью и др. Увеличения количества портов связи можно добиться путем объединения нескольких ИКМ в некую «сверхлокальную сеть», обеспечив при этом эффективный обмен данными между ними. Поскольку современные коммуникационные интерфейсы достигли гигабитных скоростей передачи, объединяющие ИКМ тракты должны иметь аналогичную пропускную способность. Кроме того, при этом должна обеспечиваться возможность эффективной коммутации высокоскоростных трафиков. С этой задачей SRIO справляется лучше, чем например интерфейс PCI Express или даже Gigabit Ethernet.

## МЕХАНИЗМ ПЕРЕДАЧИ СООБЩЕНИЙ ЧЕРЕЗ ИНТЕРФЕЙС SRIO

Как уже отмечалось выше, интерфейс SRIO — высокоскоростной последовательный дуплексный интерфейс для равноправного (peer to peer) взаимодействия множества абонентов. Это взаимодействие может выражаться как в прямом адресном доступе к оперативной памяти другого абонента для чтения или записи, включая атомические (atomic) операции и блочную (streaming) запись, так и в обмене сообщениями, не предполагающем «вторжение» в чужую память. Именно второй вариант взаимодействия, более естественный для систем телекоммуникации и пакетной коммутации, оказывается наиболее удобным для объединения множества ИКМ. Сервис обмена сообщениями на логическом уровне интерфейса SRIO дает возможность обмениваться сообщениями размером до 2 Кбайт с гарантией их доставки и контролем целостности.

Обобщенная модель взаимодействия двух абонентов по интерфейсу SRIO, показанная на рисунке 1, предполагает в общем случае два типа участников.

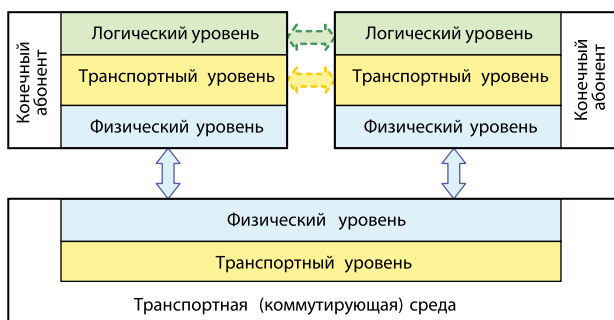


Рис. 1. Модель взаимодействия абонентов по интерфейсу SRIO

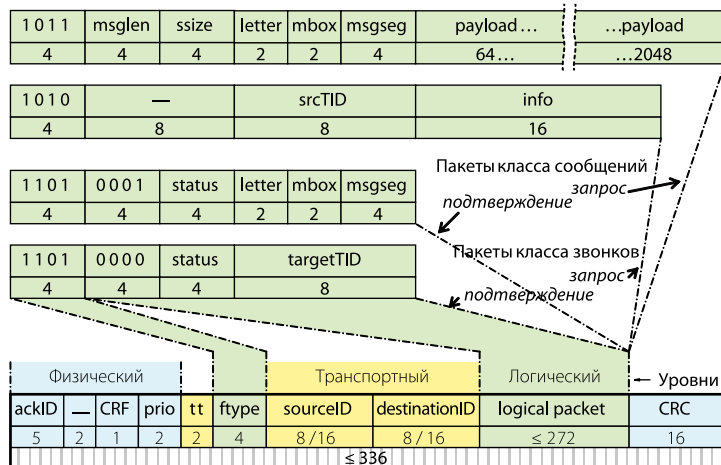


Рис. 2. Структура пакетов интерфейса SRIO

1. Взаимодействующие конечные абоненты (endpoints), каждый из которых может быть отправителем и получателем сообщений.

2. Транспортная среда, соединяющая все конечные абоненты и выполняющая доставку пакетов, в частности, с использованием коммутационных элементов.

Конечные абоненты включают все три уровня взаимодействия: логический, транспортный и физический, а транспортная среда — только два последних.

Интерфейс SRIO допускает пакеты двух классов:

- пакеты класса сообщений (message class), несущие в поле **payload** длиной 8...256 байт целое сообщение или его сегмент;
- пакеты класса звонков (doorbell class), включающие поле **info** фиксированного размера (2 байта) с произвольной информацией.

Прием каждого пакета конечный абонент-получатель подтверждает ответным пакетом. Пакеты-подтверждения однозначно соотносятся с подтверждаемыми пакетами-запросами: для подтверждения пакетов класса сообщений используется их естественная явная нумерация, а для пакетов класса звонков в их заголовок вводится специальный идентификатор транзакции. Структура пакетов показана на рисунке 2.

На логическом уровне конечного абонента готовое к отправке сообщение, идентифицируемое уникальной парой номера почтового ящика и кода письма, разбивается на сегменты (segmentation) одинаковой длины, которая может быть равна 8, 16, 32, 64, 128 или 256 байт. Последний сегмент дополняется до выбранного размера нулями (padding). Каждый сегмент нумеруется по порядку и помещается в поле **payload** отдельного пакета, к которому добавляется заголовок. Заголовок логического уровня пакета класса сообщений содержит следующие поля (см. рис. 2):

- **ftype** — тип пакета (сообщение или звонок, запрос или подтверждение);
- **msglen** — количество пакетов (сегментов) в сообщении;
- **ssize** — длина поля **payload** в байтах (8...256);
- **letter** — код письма;
- **mbox** — номер почтового ящика;
- **msgseg** — порядковый номер пакета (сегмента) в сообщении.

Заголовок логического уровня пакета класса звонков имеет всего одно поле идентификатора транзакции **srcTID**, куда помещается очередной порядковый номер звонка, увеличиваемый с каждым следующим звонком на единицу по модулю 256.

Одинаковая длина полей **payload** во всех пакетах, относящихся к одному сообщению, и сквозная нумерация пакетов в пределах сообщения позволяют логическому уровню получателя корректно восстанавливать сообщения (reassembly) из пакетов, поступающих вразбивку, а это, в свою очередь, дает возможность транспортной среде не заботиться о сохранении естественного порядка доставки пакетов. В частности, физический уровень любого элемента транспортной среды может задействовать механизм повторной передачи пакета вследствие той или иной ошибки, и возникшее из-за повторного отставания этого пакета компенсируется логическим уровнем конечного абонента-получателя.

Механизм транспорта пакетов на интерфейсе SRIO предполагает, что каждый конечный абонент имеет свой уникальный абонентский идентификатор. Длина абонентских идентификаторов может быть 8 или 16 разрядов (задается полем **tt** в заголовке пакетов), но она должна быть одинаковой у всех конечных абонентов, объединяемых одной транспортной средой. Транспортный уровень добавляет к пакету-запросу идентификаторы абонентов: отправителя (в поле **sourceID**) и получателя (в поле **destinationID**). При отправке пакета-подтверждения содержимое этих полей меняется местами.

Физический уровень дополняет пакет своим заголовком и контрольной суммой. Заголовок физического уровня включает три поля (см. рис. 2):

- **ackID** — идентификатор подтверждения;
- **prio** — приоритет;
- **CRF** — необязательный маркер критического пакета.

Поле приоритета и маркер критического пакета, дискриминирующий пакеты по срочности внутри каждого уровня, используются транспортной средой для диспетчеризации пропускания пакетов на тех звеньях, где возникают разного рода коллизии.

В интерфейсе SRIO помимо механизма подтверждения между конечными абонентами на логическом уровне с помощью специальных пакетов, предусмотрен механизм подтверждения получения каждого пакета на физическом уровне, работающий на любом отдельном звене интерфейса как между абонентами, так и между компонентами транспортной среды. Для этого используются служебные управляющие символы (control symbols) — специальные различаемые физическим уровнем 24-разрядные структуры фиксированного формата, который показан на рисунке 3. Кроме подтверждения пакетов, управляющие символы используются для разграничения пакетов, установления и поддержа-

stype0	parameter0	parameter1	stype1	cmd	CRC
3	5	5	3	3	5
24					

Рис. 3. Формат управляющего символа

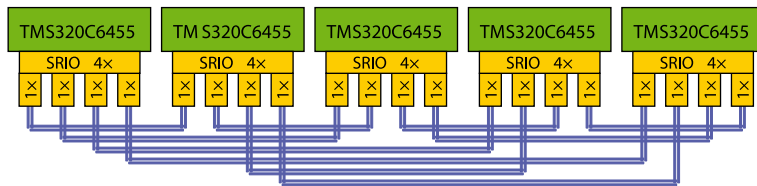


Рис. 4. Непосредственное объединение пяти ПОС TMS320C6455

ния соединения на звене, а также выполнения других служебных функций.

Поле **stype0** управляющего символа содержит либо информацию о состоянии звена интерфейса, либо код подтверждения пакета: принят, не принят, требуется повтор. Поля **parameter0** и **parameter1** дополняют эту информацию кодами состояния звена или идентификатором подтверждения из поля **ackID** полученного пакета. Поле **stype1** содержит либо индикаторы начала, конца или отмены пакета, либо запрос состояния звена. В последнем случае семантика запроса уточняется полем **cmd**. Весь управляющий символ защищается 5-разрядным контрольным кодом в поле **CRC**.

Управляющие символы непрерывно циркулируют на каждом звене интерфейса SRIO. Они обязательно выдаются перед каждым пакетом, а также после завершения его передачи и включают в себя индикатор начала и конца пакета. (Если пакеты следуют вплотную один за другим, между ними достаточно одного индикатора начала пакета.) Управляющие символы поддержания соединения на звене могут выдаваться периодически, причем не только между пакетами, но и внутри них.

#### ТРАНСПОРТНАЯ СРЕДА ИНТЕРФЕЙСА SRIO

Физическая реализация интерфейса SRIO специфицирована для дуплексного звена, имеющего в каждом направлении одну (используется обозначение 1x) или четыре (4x) дифференциальные пары (lanes) с низкоуровневыми высокочастотными сигналами и кодированием 8b/10b. Поскольку интерфейс SRIO рассчитан на сопряжение как чипов на плате, так и плат в устройстве, для него предусмотрены три скорости передачи данных, причем из-за избыточности кодирования различаются две группы скоростей: «сырая» непосредственно на линии и «рафинированная» пользовательская с соотношением между скоростями 10:8 согласно принятому кодированию. Из таблицы 1 следует, что для интерфейса SRIO диапазон пользовательских скоростей передачи данных составляет от 1 Гбит/с (низкая скорость в варианте 1x) до 10 Гбит/с (высокая скорость в варианте 4x).

Одно дуплексное звено интерфейса SRIO может соединить не более двух конечных абонентов. Если конечный абонент имеет несколько портов SRIO, то их объединение не представляет проблем. На рисунке 4 показан простой пример объединения по принципу «каждый с каждым» пяти ПОС TMS320C6455 компании

Texas Instruments, имеющих вариантный интерфейс SRIO: один порт типа 4x или четыре порта типа 1x.

Однако в общем случае для объединения множества конечных абонентов, особенно имеющих один-единственный порт SRIO, приходится организовывать те или иные транспортные среды, основным элементом которых является коммутатор. Например, 8-портовый коммутатор SRIO (элемент S на рисунке 5) может объединить восемь конечных абонентов (A 1...A 8 на рисунке 5). При этом типы звеньев могут быть любыми согласно возможностям конкретных конечных абонентов и системным требованиям (в частности, на рисунке 5 конечные абоненты A 1 и A 5 подключены через звенья 4x, а все остальные конечные абоненты — через звенья 1x).

Пример более сложной топологии транспортной среды показан на рисунке 6, где четыре 8-портовых коммутатора S 1...S 4 образуют транспортную среду для объединения 20 конечных абонентов A 1...A 20. В этом примере коммутаторы SRIO соединены с конечными абонентами звеньями 1x в предположении, что эти звенья могут быть относительно длинными, а между собой, «каждый с каждым», — звеньями 4x для обеспечения более высокой пропускной способности при условии, что все коммутаторы находятся относительно близко друг от друга. На разнообразных коммутаторах SRIO можно строить транспортные среды еще более сложных многоуровневых иерархических топологий с практически неограниченным количеством конечных абонентов. Далее будут вкратце рассмотрены коммутаторы SRIO, выпускаемые двумя ведущими в данной области компаниями: Tundra Semiconductor и IDT.

Компания Tundra занимается выпуском микросхем поддержки разнообразных

Таблица 1. Скорости передачи на звене 1x SRIO

Скорость	«Сырая», Гбод	«Рафинированная», Гбит/с
Низкая	1,250	1,0
Средняя	2,500	2,0
Высокая	3,125	2,5

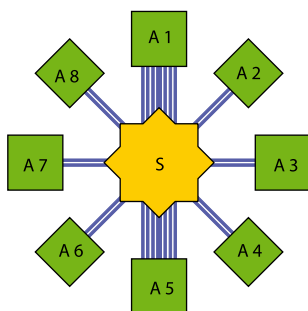


Рис. 5. Звездообразная топология транспортной среды SRIO

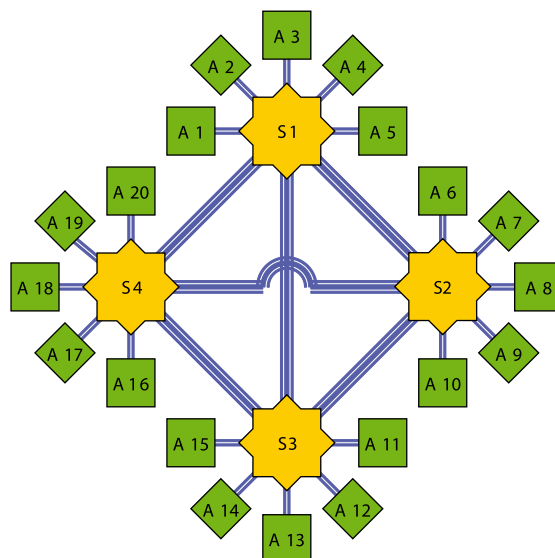


Рис. 6. Пример топологии транспортной среды с 4-мя коммутаторами SRIO



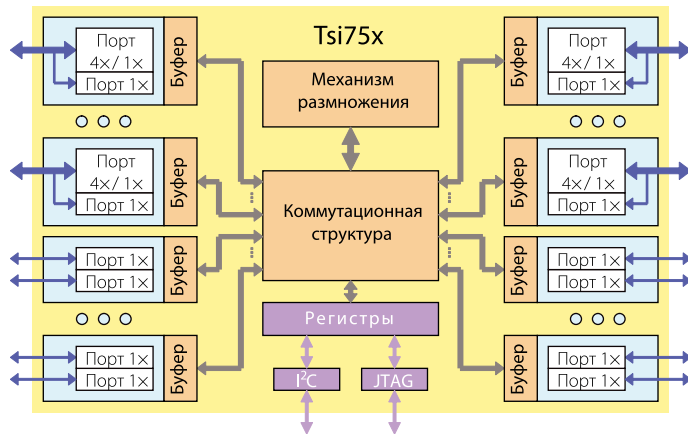


Рис. 7. Концепция коммутаторов SRIO компании Tundra

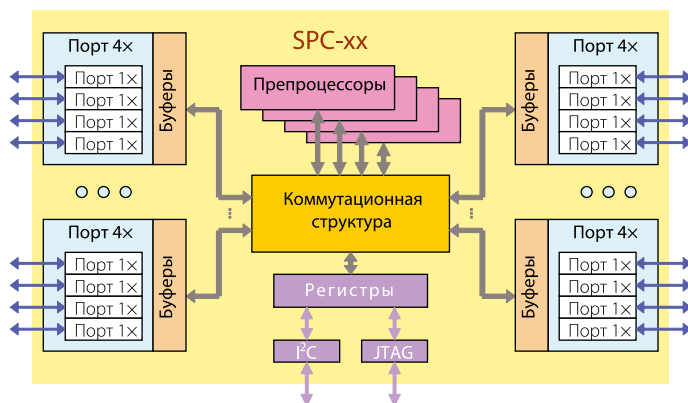


Рис. 8. Концепция коммутаторов SRIO компании IDT

интерфейсов, в т.ч. VME, PCI Express и SRIO. В частности, в серии коммутаторов SRIO Tsi75x прибор Tsi752 положил начало уже четвертому поколению коммутаторов. Впрочем, каждые поколения отражают в основном технологический прогресс; концепция коммутаторов SRIO этой компании, проиллюстрированная на рисунке 7, остается в целом неизменной.

В этой концепции двоякие внешние порты коммутатора SRIO бывают двух видов:

- комбинированные, которые можно сконфигурировать либо как один порт типа 4x, либо как два порта типа 1x;
- упрощенные, жестко сконфигурированные как два порта типа 1x.

Маршрутизация входящих пакетов выполняется входными портами коммутатора на основании распределенной по портам статической таблицы маршрутизации (lookup table). Эта таблица формируется парами программируемых параметров: идентификатором конечного абонента и поставленным ему в соответствие номером порта коммутатора. Эти программируемые параметры записываются в адресуемые регистры коммутатора, доступные как конечным абонентам через любой рабочий порт SRIO, так и специальному административному микропроцессору через служебные порты I<sup>2</sup>C и JTAG.

Непосредственно коммутацию пакетов выполняет коммутационная структура (switch fabric), которая представляет собой классическую ортогональную матрицу, обеспечивающую неблокирующую коммутацию на скорости поступления данных. Буферизация входящих пакетов выполняется в децентрализованных буферах с произвольным доступом, чтобы обеспечить дифференциацию пакетов по приоритетам (согласно полю `prio` заголовков) и исключить блокировку в голове очереди (head-of-line blocking). Кроме того, входные буферы хранят многоадресные (multicast) пакеты в процессе их размножения специальным механизмом. Этот механизм, подключенный к коммутационной структуре, реплицирует многоадресные пакеты для одновременной их доставки всем адресатам группы.

Коммутатор способен коммутировать пакеты в двух режимах:

- с полной буферизацией (store-and-forward mode), когда решение о коммутации принимается только после завершения приема и контроля всего пакета;
- «на лету» (cut-through mode), непосредственно сразу после приема поля destinationID транспортного заголовка пакета.

Очевидно, что во втором случае коммутация осуществляется быстрее, но

остается вероятность того, что будет скомутирован пакет, принятый с ошибкой. При полной буферизации ошибочные пакеты вообще не коммутируются, а входной порт запрашивает соответствующим управляющим символом повторную передачу такого пакета.

Буферизация в выходном порте преследует две цели:

- демпфирование всплесков поступления пакетов в один выходной порт одновременно с нескольких входных портов;
- хранение уже отправленных пакетов до получения подтверждения их получения, чтобы сохранялась возможность повторной передачи.

Служебные порты I<sup>2</sup>C и JTAG дают возможность конфигурировать коммутатор, программно настраивая типы и скорости передачи данных внешних рабочих портов (оба эти параметра могут также устанавливаться автоматически по «договоренности» между абонентами звена SRIO в соответствии с протоколом физического уровня), а также задавать режимы коммутации, способы арбитража и др. Порт I<sup>2</sup>C способен работать в режиме ведущего и ведомого. В первом случае коммутатор после включения питания самостоятельно конфигурируется через подключенную по интерфейсу I<sup>2</sup>C последовательную память EEPROM. Во втором случае конфигурирование коммутатора выполняется внешним устройством: либо выделенным конечным абонентом, либо специальным административным микропроцессором, имеющим порт I<sup>2</sup>C.

Коммутаторы компании IDT (см. рис. 8) в целом аналогичны коммутаторам компании Tundra, но имеют два существенных отличия. Во-первых, внешние порты коммутаторов более универсальны в том смысле, что каждый из них может быть сконфигурирован либо как один порт типа 4x, либо как четыре порта типа 1x. Во-вторых, некоторые приборы этой компании имеют по несколько встроенных препроцессоров, которые способны осуществлять предварительную обработку «на лету» коммутируемых данных, в частности оцифрованных сигналов. Эта обработка может варьироваться от простейшего пооктетного бит-реверса коммутируемых данных до цифровой фильтрации, например с помощью КИХ-фильтра.

То, что препроцессоры на рис. 8 как будто заняли место механизма размножения на рис. 7, не означает, что коммутаторы IDT не способны к размножению многоадресных пакетов. Эта стандартная для интерфейса SRIO функция реализуется внутри коммутационной структуры.

Характеристики некоторых коммутаторов SRIO компаний Tundra и IDT сопоставлены в таблице 2.

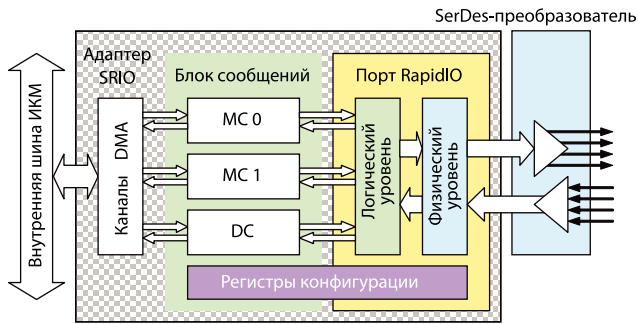


Рис. 9. Упрощенная структура адаптера SRIIO в ИКМ компании Freescale

### РЕАЛИЗАЦИЯ ИНТЕРФЕЙСА SRIIO В ИКМ

Компания Freescale включает адаптер интерфейса SRIIO во многие свои ИКМ последнего поколения PowerQUICC III. В целом реализация этого адаптера соответствует, с небольшими не принципиальными ограничениями, стандартной спецификации интерфейса SRIIO. В соответствии с этой спецификацией адаптер ИКМ поддерживает все стандартные скорости передачи данных и все режимы, свойственные конечному абоненту: адресный доступ и обмен сообщениями. Ниже будут затронуты только основные аспекты реализации адаптера, связанные с передачей сообщений.

Упрощенная структура адаптера интерфейса SRIIO, используемого в семействе ИКМ PowerQUICC III, показана на рисунке 9.

Блок сообщений адаптера SRIIO в ИКМ PowerQUICC III включает три контроллера:

- два контроллера сообщений (message controllers) MC0 и MC1;
- один контроллер звонков (doorbell controller) DC.

Все три контроллера статически конфигурируются программным исполнением адресуемых регистрационных регистров до начала работы. В процессе работы блок сообщений, как и большинство других адаптеров в ИКМ компании Freescale, может работать в двух режимах: пассивном и активном.

В пассивном режиме для передачи каждого сообщения программа пользователя заполняет соответствующие рабочие адресуемые регистры блока сообщений, в т.ч. дает ссылку на готовое к передаче сообщение в оперативной памяти конечного абонента, после чего запускает этот блок на передачу.

В активном режиме блок сообщений освобождает пользовательскую программу от «общения» со своими рабочими адресуемыми регистрами. Вместо этого программа организует в оперативной памяти циркулярный буфер с цепочкой описателей, готовых к передаче сообщений. Описатель сообщения содержит ссылку на буфер в памяти, хранящий само сообщение, а также прочую релевантную для данного сообщения информацию. После включения блок сообщений поочередно вычитывает описатели из цепочки, загружает свои регистры необходимой информацией и без вмешательства программы передает сообщение за сообщением до исчерпания описателей в цепочке. Однако цепочка описателей сообщений, сама организованная как циркулярный буфер, может никогда не исчерпаться, если пользовательская программа вовремя восполняет пустоты в цепочке от использованных описателей. Для этого предусмотрены прерывания программируемого ядра

Таблица 2. Характеристики коммутаторов SRIIO

Параметр	Компания, прибор				
	Tundra		IDT		
	Tsi572	Tsi578	CPS-8	CPS-10Q	PPS-10Q
Количество портов 4x	2	8	2	10	10
Количество портов 1x	8	16	8	16	22
Количество препроцессоров	—	—	—	—	10
Общая пропускная способность, Гбит/с	30	80	20	100	100
Корпус	BGA-399 (21x21 мм)	BGA-675 (27x27 мм)	BGA-324 (19x19 мм)	BGA-676 (27x27 мм)	BGA-676 (27x27 мм)

POWER INTEGRATIONS, INC.



МАКРО ГРУПП  
Внедряя Качество

## ПРОЕКТИРОВАНИЕ ИМПУЛЬСНЫХ ИСТОЧНИКОВ ПИТАНИЯ

по вашему техническому заданию на основе микросхем Power Integrations!

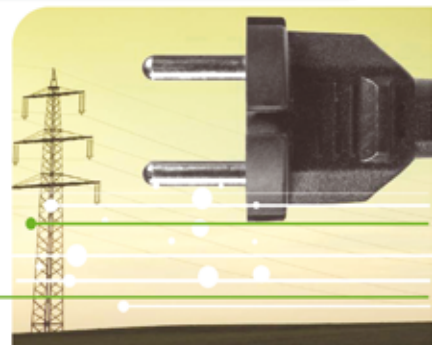
- Разработка электрической принципиальной схемы источника питания
- Проектирование моточных изделий
- Подбор комплектующих элементов
- Расчёт стоимости блока питания
- Разводка печатной платы
- Сборка действующего макета источника питания
- Обеспечение образцами заказчика
- Дальнейшее сопровождение и техническая поддержка проекта



Проектирование выполняется бесплатно при планируемом объеме производства более 10 000 изделий/год. При объеме менее 10 000 изделий/год стоимость работ согласуется индивидуально.

Макро Групп — официальный дистрибьютор Power Integrations на территории России и СНГ

www.macrogroup.ru



С-Петербург  
тел.: (812) 370 60 70

Ростов-на-Дону  
тел.: (863) 227 03 93

Екатеринбург  
тел.: (343) 218 74 08

Москва  
тел.: (495) 975 79 18

Чебоксары  
тел.: (835) 237 10 76



## КОМПОНЕНТЫ СИСТЕМ ЭЛЕКТРОПИТАНИЯ



- AC/DC ПРЕОБРАЗОВАТЕЛИ
- DC/DC ПРЕОБРАЗОВАТЕЛИ
- ФИЛЬТРЫ
- ИСТОЧНИКИ БЕСПЕРЕБОЙНОГО ПИТАНИЯ
- ЗАКАЗНЫЕ СИСТЕМЫ ЭЛЕКТРОПИТАНИЯ
- АКСЕССУАРЫ
- РАЗРАБОТКА  
И СЕРВИСНОЕ ОБСЛУЖИВАНИЕ
- ПРИЁМКА ОТК И ПЗ

НОВЫЕ ЦЕНЫ  
НИЖЕ!  
ДО 50%

ВОРОНЕЖ:  
ТЕЛ./ФАКС: (4732) 519-518  
ALEXDON@YMAIL.RU

МОСКВА:  
ТЕЛ./ФАКС: (495) 674-93-70  
AEI@AEDON.RU

ПРАГА:  
ТЕЛ.: +420 266-107-455  
ФАКС: +420 266-107-609  
AEPS@AEPS-GROUP.COM

WWW.AEPS-GROUP.RU



ИКМ адаптером SRIO по завершению передачи отдельного сообщения и/или всей цепочки сообщений, а также в случае возникновения тех или иных ошибок передачи. Таким образом, в активном режиме при корректном программировании блок сообщений может обеспечивать непрерывную передачу сообщений сколь угодно долго.

Контроллеры MC0, MC1 и DC блока сообщений работают параллельно и независимо друг от друга. Все сообщения, адресованные в почтовые ящики №0, проходят на передаче и приеме через контроллер MC0, а все сообщения, адресованные в другие почтовые ящики, — через контроллер MC1. Все исходящие и входящие звонки обслуживает контроллер DC.

При передаче каждого отдельного сообщения блок сообщений разбирает его на сегменты и передает последовательность сегментов порту RapidIO, который из них формирует и передает пакеты SRIO. Контроллеры MC в ИКМ обладают нестандартной способностью к рассылке многоадресных сообщений размером не более 256 байтов (т.е. одного пакета). В отличие от стандартной процедуры одновременной параллельной рассылки, реализуемой обычно коммутаторами SRIO, контроллеры MC многоадресную рассылку выполняют автоматическим повтором передачи одного и того же сообщения поочередно по всем адресам группы. Группы могут объединять до 32-х конечных абонентов в произвольных конфигурациях, создаваемых пользователем.

При приеме контроллер MC блока сообщений формирует сообщения из полученных пакетов следующим образом.

Для приема сообщений пользовательская программа изначально создает в оперативной памяти конечного абонента циркулярный буфер приема размером, достаточным для размещения всех поступающих сообщений. По заголовку первого пакета входящего сообщения контроллер MC определяет его размер как произведение содержимого полей **ssize** и **msglen** и резервирует в циркулярном буфере приема область соответствующего размера. В дальнейшем все пакеты, отно-

сящиеся к данному сообщению, помещаются контроллером MC в эту область, причем смещение внутри нее для каждого конкретного пакета вычисляется как произведение содержимого полей **ssize** и **msgseg** его заголовка. После приема всех пакетов (т.е. **msglen** пакетов в любом порядке) в отведенной области буфера собирается полное сообщение. Блок сообщений адаптера SRIO имеет возможность прерывать программируемое ядро ИКМ после приема каждого отдельного сообщения и/или по заполнению всего циркулярного буфера приема, а также в случае возникновения каких-либо ошибок.

Оконечный блок адаптера — SerDes-преобразователь параллельного кода в последовательный и обратно (serializer/deserializer) — поддерживает все три стандартные скорости (см. табл. 1). К сожалению, блок SerDes-преобразователя в ИКМ является, как правило, универсальным компонентом, используемым другими последовательными интерфейсами, в частности PCI Express и SGMII. Возможно, вследствие этого он, хотя и работает в любом из режимов 1x или 4x, но в обоих вариантах обеспечивает всего лишь единственный порт SRIO, что безусловно ограничивает возможности использования ИКМ и может рассматриваться как его недостаток. Тем не менее и наличие единственного порта SRIO открывает для ИКМ новые возможности.

Например, ИКМ MPC8548 имеет четыре встроенных коммуникационных порта трехскоростного Ethernet (10/100/1000 Мбит/с) и один порт SRIO. Приблизительные оценки показывают, что суммарный поток данных через все четыре порта Ethernet в каждом направлении не превысит 4 Гбит/с. Даже если все четыре потока коммутируются на порт SRIO, тот (в варианте 4x) окажется в состоянии пропустить весь суммарный поток даже на самой низкой скорости (4 × 1 Гбит/с). Объединение, например, 16-ти ИКМ MPC8548 с помощью коммутатора SRIO Tsi578 позволит организовать коммутатор Gigabit Ethernet с 64-мя портами, в каждом узле которого (конечном абоненте SRIO) окажется мощный процессор с архитектурой PowerPC и рабочей частотой до 1 ГГц. Такой коммутатор может попутно выполнять множество полезных функций от маршрутизации и защиты информации до управления сетью.

Сервисы передачи сообщений и потоковой инкапсуляции на интерфейсе SRIO вполне соответствуют требованиям современной пакетной коммутации. В частности, максимальная длина сообщений 2 Кбайт достаточна для инкапсуляции типичного IP-пакета размером 1500 байт, а ограничение на объем блока данных при потоковой инкапсуляции в 64 Кбайт позволяет инкапсулировать даже Ethernet-кадры jumbo длиной до 9000 байт.

С ростом популярности интерфейса SRIO растет и разнообразие технических и программных средств его поддержки. Большинство производителей микропроцессоров и микроконтроллеров с интерфейсом SRIO, включая ПИОС и ИКМ, предлагает его программную поддержку на разных уровнях. Например, компания Texas Instruments включила поддержку SRIO в свои операционные системы RTOS и DSP/BIOS на уровне прикладных интерфейсов программирования (application programming interfaces), что позволяет разрабатывать программное обеспечение, использующее возможности SRIO, при высоких уровнях абстрагирования от аппаратуры. Компания Freescale, как мы уже видели, использовала для облегчения пользовательского программирования интерфейса SRIO свой фирменный механизм управления внешними коммуникационными портами ИКМ через механизм цепочек описателей буферов/сообщений. Кроме того, поддержка SRIO включена компанией в популярную интегрированную среду разработки CodeWarrior.

### ЛИТЕРАТУРА

1. [www.tundra.com/products/rapidio-switches](http://www.tundra.com/products/rapidio-switches)
2. [www.idt.com/?catid=18446178](http://www.idt.com/?catid=18446178)
3. [www.idt.com/?catid=18388262](http://www.idt.com/?catid=18388262)
4. [www.freescale.com/files/32bit/doc/app\\_note/AN2923.pdf?srch=1](http://www.freescale.com/files/32bit/doc/app_note/AN2923.pdf?srch=1)
5. [focus.ti.com/lit/ds/symlink/tms320c6455.pdf](http://focus.ti.com/lit/ds/symlink/tms320c6455.pdf)