УДК 004.75

**Богданов А.Ю.**
(ЗАО «МЦСТ»)

# Реализация межмашинных связей в распределенных вычислительных системах на базе микропроцессоров разработки ЗАО «МЦСТ»

*Рассматривается проблема выбора коммуникационных интерфейсов для реализации распределенных вычислительных систем на базе разработок эльбрусовской серии. Приводится описание спроектированного контроллера моста RDMA–RapidIO.*

***Ключевые слова: RapidIO, RDMA, коммуникационная сеть, обмен сообщениями.***

## Введение

В настоящее время при межмашинном обмене в распределенных информационно-вычислительных системах на базе вычислительных комплексов (ВК) семейства «Эльбрус» используется оригинальный протокол RDMA. Его применение в качестве альтернативы стандартному протоколу была вызвана в свое время сжатыми сроками разработки.

Протокол RDMA рассчитан на различную конфигурацию (кольцо, тор) подключения вычислительных средств и позволяет создавать многопроцессорные системы, которые включают в общей сложности до 32 процессорных ядер. Однако возможности дальнейшего расширения конфигураций системы весьма ограничены. В связи с этим было принято решение в дальнейших проектах перейти к организации связи между машинами через промежуточную среду на базе коммутирующих узлов (коммутаторов), позволяющую строить распределенные вычислительные системы произвольного масштаба. Таким образом, возникла проблема использования нового интерфейса для связи ВК с коммуникационной средой и, соответственно, нового протокола, сменяющего RDMA.

Анализ возможных вариантов привел к выбору стандартного интерфейса RapidIO, который, позволяя в перспективе перейти к созданию открытых систем, включающих вычислительные средства других архитектур, обладает характеристиками, вполне соответствующими ожидаемым параметрам коммуникационных трафиков эльбрусовской системы.

## 1. Текущее решение

RDMA-интерфейс расположен непосредственно в процессоре и позволяет связывать вычислительные комплексы высокоскоростным DMA-каналом с пропускной способностью 667 Мбайт/с (для микропроцессора МЦСТ-R500S) и 1 Гбайт/с (для систем на кристалле МЦСТ-R1000, ЭЛЬБРУС-2С+) [1] в каждом направлении на уровне межмодульной и внутримодульной связей.

Архитектура RDMA включает в себя три уровня [2] - физический, транспортный и логический. Каждый из них является заменяемым и выполняет определенный набор функций:

1. Согласно спецификациям *физического уровня*, порт RDMA состоит из 10 параллельных дифференциальных полнодуплексных линий связи. Восемь линий предназначены для данных, одна – для управления и одна – для передачи сигнала синхронизации. Помимо этих, типичных для физического уровня установок, в него включены функции управления потоком, контроля целостности данных и управления при ошибках.

2. *Транспортный уровень* отвечает за адресацию при взаимодействии групп абонентов. RDMA-абоненты (вычислительные комплексы) могут объединяться в сети типа «кольцо» (рис. 1) (микропроцессор МЦСТ-R500S) и тор (системы на кристалле МЦСТ-R1000, ЭЛЬБРУС-2С+).

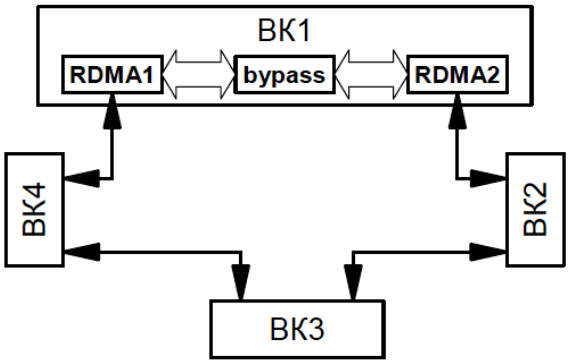


Рис. 1. Организация сети типа «кольцо»

3. *Логический уровень* обеспечивает работу двух основных режимов: BUS и DMA. Режим BUS предназначен для работы с внешней периферией, имеющей выходы на стандартные интерфейсы или встроенной видеосистемой с общей памятью. Режим DMA предназначен для межмашинного обмена.

Высокий темп обмена и минимальное время доступа за счет расположения в процессоре обеспечивали успешное применение RDMA в составе нескольких типов микросхем эльбрусовской серии до тех пор, пока не возникла необходимость построить на их основе распределенные системы с большим количеством оконечных узлов. В принципе она

была предусмотрена в спецификации компании ЗАО «МЦСТ», предполагающей возможность использования коммутатора RDMA, но, как отмечалось выше, альтернативой его разработке и верификации «от нуля» стало решение о выборе стандартного интерфейса.

## 2. Выбор стандартного интерфейса

В состав требований к новому интерфейсу входили: малые времена задержки, надежность, масштабируемость, высокая пропускная способность, минимум служебных данных и накладных затрат (использования процессорного времени), построение сети с использованием коммутаторов (в пределах вычислительного комплекса), наличие готовых устройств на рынке.

В этом контексте рассматривались интерфейсы InfiniBand, Ethernet 10Gb, RapidIO. Стандарт InfiniBand изначально разработан для задач высокоскоростного обмена в серверах и суперЭВМ. Поэтому, его применение в качестве внутри и межмодульной связи является неэффективным. Ethernet 10Gb обеспечивает высокую скорость передачи данных и поддерживается в промышленных высокоскоростных сетевых коммутаторах, однако его использование связано со значительными накладными расходами на программную обработку транзакций и задержками в контроллере доступа к среде, обычно измеряемыми в микросекундах. В силу этих и некоторых других причин выбор был сделан в пользу интерфейса RapidIO, во многом удовлетворяющего сформулированным выше требованиям.

В табл. 1 проведено сравнение систем, использующих Ethernet 10Gb и RapidIO, по некоторым важным показателям.

Таблица 1

Сравнение Ethernet и RapidIO

| **Система** | **Ethernet 10Gb** | **RapidIO** |
|---|---|---|
| Скорость передачи данных на один порт | 10 Гбит/с | 20 Гбит/с (4x) |
| Время передачи пакета | >10 мкс | ~1–2 мкс |
| Обработка сообщений | ПО | Аппаратная |
| Надежность | Потеря пакетов из-за ошибок или конфликтов | Встроенное обнаружение и исправление ошибок |

Функциональная модель интерфейса RapidIO поддерживается операциями прямого доступа в память (RDMA) и обмена сообщениями, которые выполняются на аппаратном уровне. Коммуникационная сеть, использующая RapidIO, образуется межсоединениями типа «точка-точка» и может включать коммутаторы; в ней можно связать до 16К процессоров. Длина передачи данных составляет ~80-100 см + 2 разъема.

В итоге проведенного анализа было установлено, что интерфейс RapidIO в достаточной степени соответствуют концепции применения вычислительных средств семейства «Эльбрус» в распределенных вычислительных системах. Проблема внедрения этого интерфейса в действующие и перспективные системы решена разработкой моста RDMA–RapidIO.

### 3. Мост RDMA–RapidIO

Системная конфигурация с использованием моста RDMA–RapidIO приведена на рис. 2. В случае его успешного применения в дальнейшем планируется исполнение интерфейса RapidIO непосредственно в микропроцессоре. Это решение позволит напрямую коммутировать процессоры.

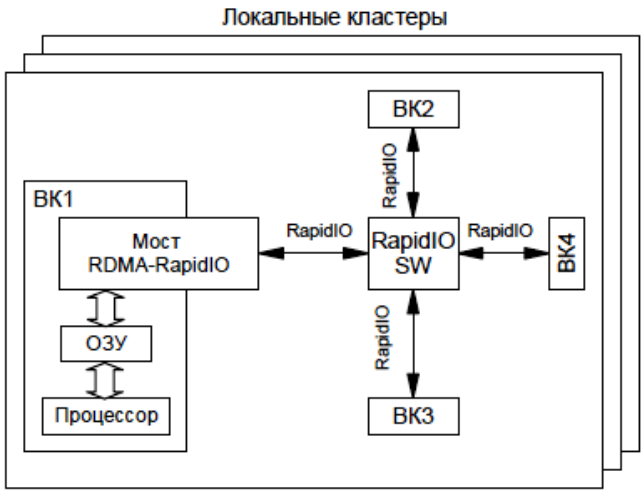


Рис. 2. Применение моста RDMA–RapidIO

Для реализации интерфейса RapidIO в составе моста выбрана спецификация версии 2.1 [3], как наиболее распространенная на данный момент.

В качестве блока физического уровня интерфейса используется IP-ядро компании Altera [4] с количеством каналов 1x/4x и скоростью передачи данных 1, 2, 2.5, 4 Гбит/с на каждую линейную пару.

Связь процессоров в коммуникационной сети через интерфейс RapidIO осуществляется путем обмена сообщениями данных (message passing) размером от 256 до 4096 байт и сигналами прерываний (сообщения-уведомления doorbell). В системных целях поддерживаются конфигурационные операции (maintenance) и аварийные сообщения (port-write) на случай сбоя.

Функциональная схема моста RDMA–RapidIO представлена на рис. 3.

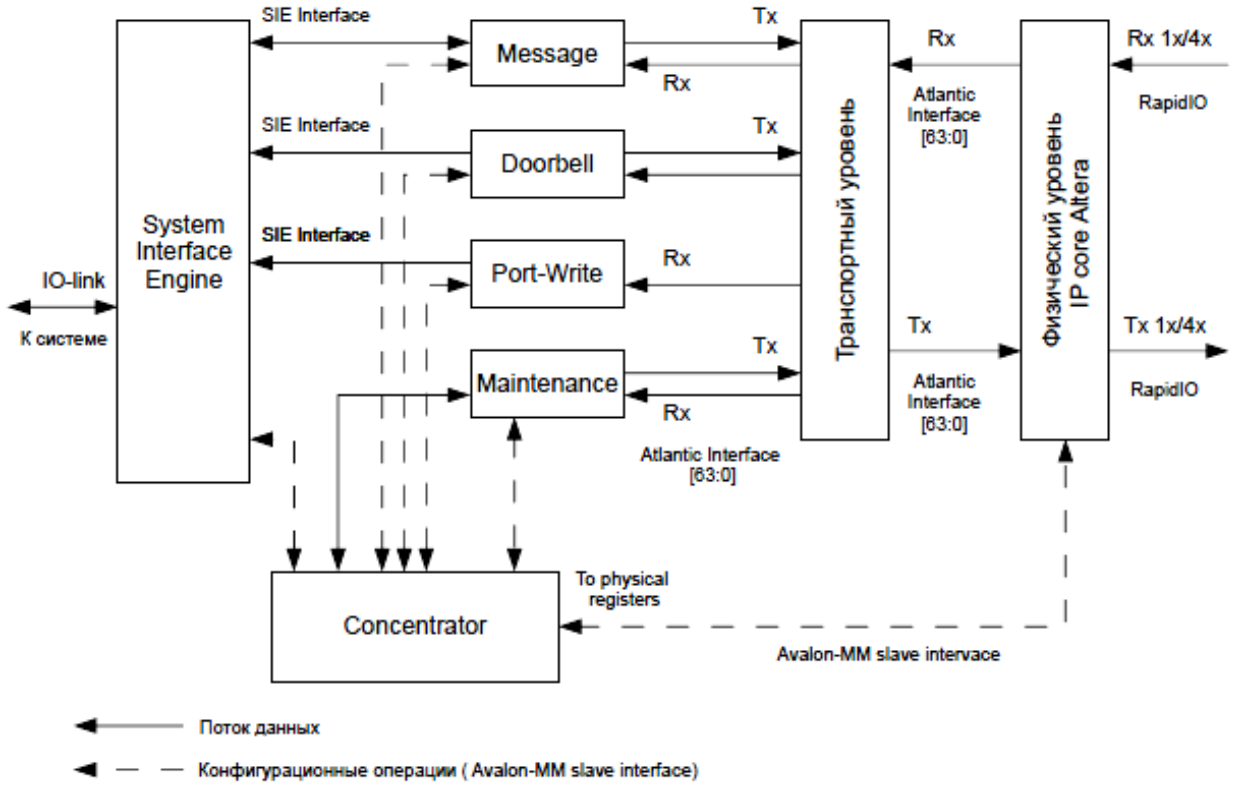


Рис. 3 Функциональная схема моста RDMA–RapidIO

Мост подключается к системе через интерфейс IO-link, входящий в состав блока System Interface Engine (SIE). Операции связанных с SIE функциональных модулей *Message* (прием и передача сообщений), *Doorbell* (прерывания), *Port-Write* (сообщения о сбое в сети) и *Maintenance* (служебные конфигурационные сообщения) выполняются независимо друг от друга. Для каждой операции в SIE-интерфейсе определена своя ширина данных. В модулях располагаются операционные и статусные регистры, связь которых с блоком физического уровня RapidIO осуществляется через интерфейс Avalon-MM Slave (далее по тексту Avalon-интерфейс) и модуль *Concentrator.*

Модуль *Concentrator* управляет двунаправленной передачей сообщений между функциональными модулями и системой. Его связь с модулями и регистрами физического уровня выполняется через Avalon-интерфейс. Он содержит в себе конфигурационные регистры, задающие возможности и текущий статус RapidIO.

В режиме передачи транспортный уровень выполняет арбитраж запросов на передачу по принципу «round-robin». В результате выбора запроса он осуществляет передачу данных через Atlantic-интерфейс.

В случае приема данных транспортный уровень осуществляет проверку принимаемого

сообщения на совпадения ID назначения пакета. При отрицательном результате пакет аннулируется и выставляется статус «*rx_packet_dropped*». При успешной проверке анализируются поля *ftype* и *transaction* для определения функционального модуля назначения. Как только он готов принять данные, осуществляется их пересылка. Если во время приема данных возникает ошибка, то пакет аннулируется.

**Заключение**

На данный момент разработано RTL-описание контроллера RapidIO, выполнено его standalone-тестирование и подготавливается макет для дальнейшей отладки в существующих эльбрусовских системах. Этот контроллер позволяет снизить количество внешних сигналов ввода-вывода в 2,5 раза (для подключения вычислительных средств по RDMA-интерфейсу требуется 40 сигналов, а для подключения по RapidIO – 8 сигналов).

Эта разработка позволяет расширить круг заказчиков ВК семейства «Эльбрус» и область их использования за счет предоставления преимуществ открытых систем и возможности применения отработанных покупных решений (коммутаторы RapidIO).

**Литература**

1. Ким А.К., Перекатов В.И., Ермаков С.Г. Микропроцессоры и вычислительные комплексы семейства «Эльбрус». СПб., Питер, 2013.

2. Воронцов М.В., Гондарь А.В., Диденко В.Б. и др. Высокоскоростной межмашинный внутрисистемный интерфейс RDMA. Международная научная конференция «Гагаринские чтения», секция «Информационные технологии», 2005.

3. RapidIO Interconnect Specification. Revision 2.1, 2009.

4. RapidIO Megacore Function. User Guide. Altera. 2011.