

*Н.А. Щербина<sup>1,2</sup>*

<sup>1</sup> ЗАО «МЦСТ»

<sup>2</sup> Московский физико-технический институт (государственный университет)

### **Разработка системного коммутатора для микропроцессора МЦСТ-4R**

В настоящее время потребность в высокопроизводительных вычислительных комплексах привела к созданию нескольких классов многопроцессорных систем. Среди них можно выделить класс систем с неоднородным доступом к памяти (NUMA), где память физически распределена между процессорами, но логически общедоступна.

Вычислительные комплексы NUMA класса в основном строятся на базе систем на кристалле (System on Chip), в которых большинство системных контроллеров (контроллер оперативной памяти, контроллер удаленного доступа и т.д.) вместе с процессорами интегрируются в одном кристалле.

В настоящий момент компания ЗАО «МЦСТ» выполняет проект по созданию NUMA системы на базе процессора МЦСТ-4R. В нем параллельно с разработкой основных процессорных модулей проектируется подсистема памяти, включающая в себя оборудование распределенного чипсета (Chip Set). Оно обеспечивает аппаратную поддержку коммутационной среды, объединяющей отдельные процессоры. В данной работе рассматривается разработка системного коммутатора (СК), который является частью этой среды, отнесенной к одному процессору.

Одна из основных проблем создания распределенного чипсета заключается в тестировании на предмет логических ошибок. Так как верификация его модулей в составе всей системы возможна только на последних стадиях разработки rtl модели процессора, то изменение алгоритмов функционирования проектируемой аппаратуры в случае не корректной (либо неоптимальной в плане производительности) работы - очень трудоёмкий процесс. К этому можно добавить длительное время разбора ошибочных ситуаций, что, главным образом, является свойством процесса отладки оборудования в составе всей системы. Поэтому не менее важной и трудоемкой задачей, чем непосредственная разработка rtl модели коммутационного оборудования, включающего СК, является осуществление его автономного тестирования на наличие логических ошибок и на производительность.

Для тестирования производительности как всей коммутационной среды, так и СК, был использован метод, подробно описанный в [2] (Chapter23 Performance Analysis). Он основывается на генерации входного трафика, задаваемого некоторыми вероятностными параметрами. При этом информация, необходимая для измерений на выходах из СК, передается в полях пакетов, не используемых при штатной работе коммутационной среды. В результате доработки этого метода был получен автономный тест, производящий как измерения производительности коммутационной среды, так и выполняющий проверку на наличие логических ошибок. Он применим к любой коммутационной среде. Это позволяет активно применять его в процессе разработки коммутационного оборудования путем варьирования ключевых элементов, таких как арбитры [1],[2], механизмы контроля потока данных (flow control [1],[2]) и наблюдения за изменениями в поведении системы (изменениями производительности).

В результате, на данный момент разработана rtl модель СК, полностью удовлетворяющая всем функциональным требованиям. В них входит обеспечение обмена абонентов, находящихся на том же узле, что и коммутатор (L2-кэш, контроллер памяти и IO-зонд), между собой и с абонентами, расположенными на удаленных узлах многопроцессорной конфигурации, через межсистемные линки.

При выполнении своих функций СК решает ряд задач. В первую очередь, это получение пакетов от абонентов либо из системных линков; их десериализация (при необходимости); определение абонентов-получателей при помощи маршрутизирующей таблицы; коммутация пакетов к соответствующим выходным портам; их сериализация (при необходимости); отправка их получателю либо в один из системных линков.

К более сложным задачам можно отнести следующие:

- обеспечение обслуживания абонентов согласно политике приоритетов;
- обеспечение независимости пакетных потоков между различными парами абонентов (пакет из одного потока, по каким-либо причинам не получающий право на дальнейшее прохождение по коммутационной среде, не должен блокировать обработку пакетов из другого потока);
- максимально возможную загрузку пропускных способностей межсистемных линков при использовании уже разработанных для NUMA системы контроллеров межсистемного взаимодействия;
- обеспечение независимости потоков различных типов командных пакетов (запросы на различного рода чтения/записи, ответы на

выполненные команд, снупирование и.т.д.), что осуществляется по средствам механизма виртуальных каналов (virtual channel [3]).

### Литература

1. *Dally W.J., Towles B.* Principles and practices of interconnection networks. — Morgan Kaufmann, 2004.
2. *Webber M.* Arbiters: design ideas and coding styles // SNUG Boston – 2001.
3. *Dally W.J.* Virtual-channel flow control // IEEE Transactions on Parallel and Distributed Systems. - March 1992. - V. 3, N. 2. - P. 194-205.