

УДК 004.335

ФУНКЦИОНАЛЬНАЯ ОРГАНИЗАЦИЯ И АЛГОРИТМЫ РАБОТЫ АППАРАТНОГО БУФЕРНОГО УСТРОЙСТВА ПАМЯТИ МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Мартышкин А.И.

ФГБОУ ВО «Пензенский государственный технологический университет», Пенза,
e-mail: rector@penzgtu.ru

Статья посвящена решению ряда вопросов, связанных с проблемой так называемых «узких мест» многопроцессорных вычислительных систем, а именно конфликтам за доступ процессоров к общей системной шине. Показана возможность размещения между процессором и памятью аппаратно реализованного модуля буферного устройства памяти, который необходим для быстрого доступа к памяти (в структуре буферного устройства используется ассоциативная память) многопроцессорной вычислительной системы с широко используемым интерфейсом «общая шина». Описанный модуль базируется на использовании регистровой памяти и содержит в своем составе две части: первая составляющая модуля отвечает за процедуру записи данных, вторая составляющая контролирует процедуру чтения. В ходе работы определена функциональная организация блока аппаратного буферного устройства, продуманы и реализованы в проекте алгоритмы его работы, создан и отлажен VHDL-файл, описывающий функционирование устройства, проведено моделирование правильности работы в программе ISE Web Pack. Благодаря использованию современной элементной базы, а именно программируемых логических интегральных схем (ПЛИС), описанное буферное устройство является реконфигурируемым (можно в любой момент внести коррективы в VHDL-файл для изменения параметров работы, а также структуры и функциональных возможностей) и кроссплатформенным, из-за универсальности VHDL-кода устройство может быть реализовано на ПЛИС различных производителей. С применением рассмотренного блока частично удается решить проблему «узкого места» многопроцессорной системы с интерфейсом «общая шина». В результате практического использования описанного устройства увеличится пропускная способность подсистемы «процессор-память» и, соответственно, производительность всей многопроцессорной системы в целом.

Ключевые слова: многопроцессорная система, структурная организация, функциональная организация, алгоритм работы, буферное устройство памяти, режим расщепления транзакций, распределенная память, ассоциативная память, временные диаграммы работы, режим записи, режим чтения

FUNCTIONAL ORGANIZATION AND OPERATION ALGORITHMS OF HARDWARE MEMORY BUFFER OF MULTIPROCESSOR COMPUTER SYSTEM

Martyshkin A.I.

Penza State Technological University, Penza, e-mail: rector@penzgtu.ru

The article is devoted to solving a range of issues related to the problem of so-called «bottlenecks» multiprocessor computer systems, namely, conflicts over access to the shared processor system bus. The possibility of accommodation between the processor and memory hardware-implemented module buffer memory device, which is required for faster memory access (in the structure of the buffer device is used associative memory) of multiprocessor systems with widely used interface «shared bus». The described module is based on the use of registered memory and contains in its composition of two parts: the first part of the module is responsible for the procedure of data recording, the second component controls the reading procedure. In the course of work is defined as the functional organization of the hardware module of the buffer device, designed and implemented in the project algorithms of its work, created and debugged VHDL file that describes the operation of the device, the simulation of the correct operation of the program ISE Web Pack. Through the use of modern element base, namely, Field-Programmable Gate Array (FPGA) described the buffer device is reconfigurable (can be in any time to make adjustments in the VHDL-file to change the parameters, the structure and functionality) and cross-platform, because of the versatility of VHDL-code device can be implemented on FPGAs from different manufacturers. With the application of the considered block are partially able to solve the problem of bottlenecks in a multiprocessor system interface «shared bus». As result of practical use of the device described to increase the throughput of the subsystem «processor-memory» and, accordingly, the performance of the entire multiprocessor system as a whole.

Keywords: multiprocessor system, structural organization, functional organization, work algorithm, buffer memory device, transaction splitting mode, shared memory, associative memory, work diagrams, write mode, read mode

Вся сегодняшняя жизнь современного человека буквально пропитана вычислительной техникой: компьютерами и создаваемыми на их базе вычислительными системами (ВС). Они проникли повсеместно: в бытовой технике, приборах, устройствах связи и т.д. Список можно

продолжать долго. Среди всех ВС особняком стоят многопроцессорные вычислительные системы (МПВС), которые активно используются для трудоемких вычислений, например, при моделировании сложных процессов и других научных расчетов, которым необходимы ко-

лоссальное быстродействие и слаженная работа подсистемы «процессор-память».

Основной целью, которую преследует эта статья является проработка функциональной организации и алгоритмов работы аппаратно реализованного буферного устройства памяти МПВС с общей шиной. Описываемое здесь устройство необходимо, в первую очередь, для быстрого доступа к памяти и разгрузки процессоров. В результате применения данного модуля снижается загрузка памяти, повышается пропускная способность подсистемы «процессор-память» и возрастает быстродействие МПВС в общем и целом.

Представляемая статья в целом носит исследовательский характер. В ходе изучения предметной области был проанализирован ряд литературных источников [1, 9, 10] с целью поиска незатронутых вопросов и нерешенных проблем. Ряд проблемных вопросов, связанных с возможностью аппаратной реализации буферного устройства памяти многопроцессорных систем для разгрузки подсистемы «процессор-память», не нашел должного отражения в существующих публикациях, частично проблемные вопросы были проанализированы в работах [2–7, 10].

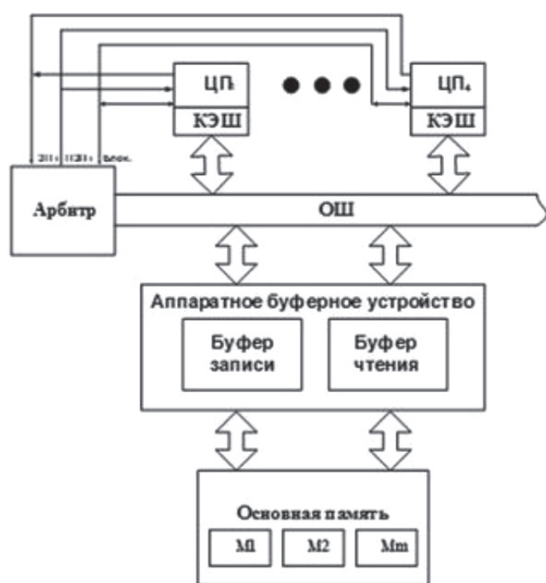


Рис. 1. Четырехпроцессорная система с АБУ памяти

Целью работы является описание рассмотренных возможных алгоритмов функционирования аппаратно реализованного буферного устройства (АБУ) памяти МПВС с интерфейсом «общая шина» (ОШ), включающей в свой состав 4 процессора (ЦП) (рис. 1). Данный вопрос сегодня является

актуальным ввиду глобальной информатизации и практически повсеместном оперировании колоссальными объемами данных. Для достижения поставленной цели в статье решаются задачи по определению структуры устройства и принципов и алгоритмов его функционирования. В существующих МПВС на занятие ОШ одновременно могут претендовать несколько устройств, однако в любой момент времени это возможно только одному из них. Для исключения возможных конфликтов ОШ должна выбирать механизмы арбитража запросов и правила предоставления шины одному конкретному устройству из запросивших [8].

ОШ взята в соответствии со спецификацией АМВА [11], которая разработана для создания высокопроизводительных систем. Возможная структура и принципы функционирования АБУ показаны в работах [3–5]. Здесь подробно остановимся на функциональной организации и алгоритмах работы АБУ.

При написании VHDL-кода схемной реализации как таковой не получается, если не придерживаться описания конкретного элемента для VHDL (например, описание функционирования регистра, счетчика, дешифратора и т.п.). Но все же попробуем выделить некоторые функциональные блоки (рис. 2):

- блок добавления транзакции в очередь чтения;
- блок добавления транзакции в очередь записи;
- блок увеличения указателя на голову буфера чтения и увеличения очереди чтения;
- блок увеличения указателя на голову буфера записи и увеличения очереди записи;
- блок поиска данных в буфере записи при совпадении адресов с буфером чтения;
- блок чтения из памяти по указанному адресу;
- блок увеличения указателя на хвост очереди обработанных сообщений;
- блок записи в память по указанному адресу;
- блок увеличения указателя на хвост буфера записи;
- блок выдачи данных ЦП, инициировавшему запрос на чтение.

На рис. 2 все блоки показаны в общем виде.

Блок добавления транзакции в очередь чтения работает следующим образом. На внутренний регистр RGA1 поступает 32-разрядный адрес от ЦП, по которому должны быть найдены данные. На регистр RGMID подается 2-разрядный идентификатор ЦП (принимает значение от 0 до 3 по числу ЦП в системе). Разрешающим сигналом для работы данных регистров является комбинация сигналов.

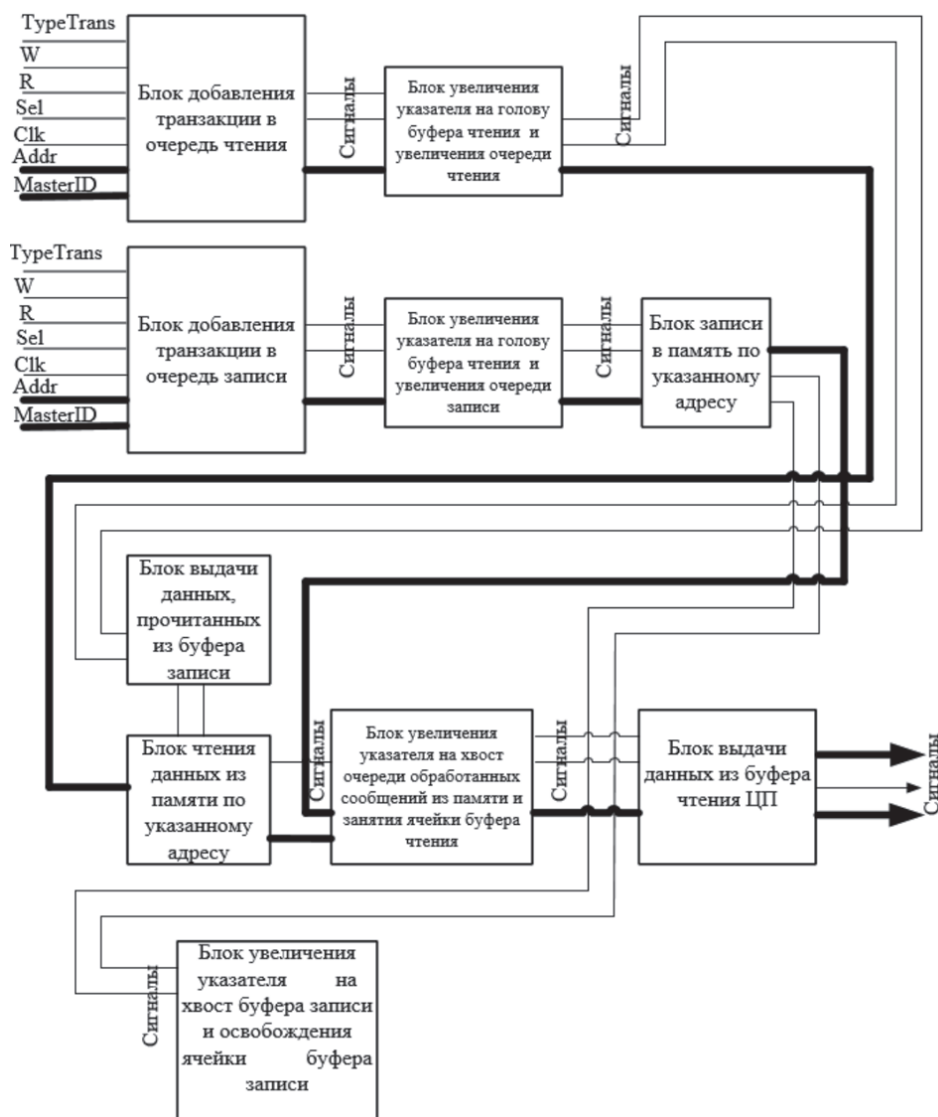


Рис. 2. Функциональные блоки АБУ памяти

Блок добавления транзакции в очередь записи работает следующим образом. На внутренний регистр RGA2 поступает 32-разрядный адрес от ЦП, по которому должны быть записаны данные. На регистр RGD подаются 32-разрядные данные от ЦП. Разрешающим сигналом для работы данных регистров является комбинация сигналов.

Блок увеличения указателя на голову буфера чтения и увеличения очереди чтения работает следующим образом. Вся очередь заявок представим состоящей из «головы очереди», «хвоста очереди» и указателя на ячейку в данный момент времени. При добавлении следующей заявки в очередь чтения «голова» увеличивается на едини-

цу и вместе с тем увеличивается сама очередь чтения. Этот блок можно представить в виде счетчиков.

Блок поиска данных в буфере записи при совпадении адресов с буфером чтения можно реализовать на 32-разрядном компараторе, на один вход которого подается адрес, выставленный ЦП в буфере чтения, на другой последовательно подаются адреса из буфера записи. При совпадении адресов ЦП возвращаются данные из буфера записи. Если же адреса не совпали, т.е. ассоциативный поиск не дал результатов, то необходимо обратиться к памяти по нужному адресу. Это действие производит следующий блок.

Блок чтения из памяти по указанному адресу работает следующим образом. Из 32-разрядного регистра, в котором хранится адрес, выставленный ЦП, этот адрес поступает на выход АБУ, на адресные входы памяти. Два старших разряда адреса используются для выбора модуля памяти, откуда необходимо прочитать данные по указанному адресу. Подается сигнал чтения из памяти MemR, который будет держаться в единице на протяжении 50 ns – время поиска данных в памяти по указанному адресу и чтение в буфер. Данные сохраняются в регистре данных в буфере чтения, и когда ЦП, который выставил адрес на чтение, вновь подключится к ОШ для получения данных, из буфера чтения им будут считаны необходимые данные, предназначенные ему. Выдается сигнал Split (номер ЦП).

Блок увеличения указателя на хвост очереди обработанных сообщений работает следующим образом. Когда произошла операция чтения из памяти, то данные не сразу выдаются ЦП. Вначале создается так называемая «очередь обработанных сообщений». С каждой новой обработанной транзакцией увеличивается очередь сообщений. Блок можно реализовать на счетчике.

Блок записи в память по указанному адресу работает следующим образом. Из 32-разрядных регистров, в которых хранятся адрес и данные в буфере записи, происходит запись в память. Здесь также два старших разряда адреса используются для выбора модуля памяти, куда будут записываться данные. Подается сигнал записи в память MemW, который будет держаться на протяжении 10 ns – время записи данных в память.

Блок увеличения указателя на хвост буфера записи работает следующим образом. При записи в память указатель на хвост буфера увеличивается на единицу с каждой обработанной транзакцией. После этого в буфере записи освобождается одна ячейка.

Блок выдачи данных ЦП, инициировавшему запрос на чтение, работает следующим образом. Как только очередь готовых заявок сформирована, т.е. все заявки очереди обработаны, данные из памяти помещены в буфер чтения, то ЦП могут забирать заявки, предназначенные им. Данные из регистров поступают на выход DataRead и ЦП считывает данные.

Теперь покажем вариант алгоритма функционирования АБУ. Представим подсистему «процессор-память» как две подсистемы: «процессор-АБУ» и «АБУ-память». Опишем алгоритм функционирования подсистемы «процессор-АБУ». Вначале ЦП проверяет линию блокировки и определяет, свободна либо занята ОШ в данный момент

времени. Допустим, высокий потенциал на линии блокировки соответствует состоянию, при котором ОШ свободна. Если же опрашивающие линию блокировки ЦП обнаруживают там высокий потенциал, они отправляют запросы в шинный арбитр. ЦП с наивысшим приоритетом получит сигнал, подтверждающий запрос. После захвата ЦП ОШ выбирается тип операции: чтение или запись. В случае выполнения операции записи, буфер записи проверяется на наличие свободного места и в случае, если он полон, ЦП переводится в режим ожидания, где находится до появления свободной ячейки. В случае, если место есть, происходит запись адреса и данных, после чего ЦП освобождает ОШ. В случае выбора операции чтения, сначала буфер чтения проверяется на наличие свободного места и в случае их отсутствия ЦП переводится в режим ожидания пока не появятся свободная ячейка. Если свободная ячейка имеется, происходит запись адреса, по которому должны быть предоставлены данные для чтения. После процедуры физического чтения из памяти или поиска в ассоциативной памяти буфера записи по выставленному адресу считываются данные в буфер чтения. АБУ оповещает ЦП, которому подготовлены считанные данные, о готовности, и он забирает их из АБУ.

Опишем возможный алгоритм работы подсистемы «АБУ-память». Вначале выбирается тип операции: чтение или запись. При выборе операции записи осуществляется запрос j -го модуля памяти, куда будут записываться данные, пересланные в АБУ из ЦП. Далее j -й модуль памяти проверяется на занятость, и если он занят, подсистема переходит в режим ожидания, если свободен, происходит запись данных по нужному адресу. Далее модуль памяти, как и одна ячейка буфера записи освобождаются. При выполнении операции чтения запрашивается k -й модуль памяти, откуда будут считываться данные в АБУ, для дальнейшей передачи их соответствующему ЦП. После этого проверяется k -й модуль памяти на занятость, и если он занят, подсистема переходит в режим ожидания, если свободен, происходит чтение данных по нужному адресу. После этого считанные данные записываются в ячейку буфера чтения. Затем модуль памяти высвобождается, а АБУ оповещает запросивший ЦП о выполнении операции.

По приведенным алгоритмам работы АБУ в среде ISE WebPack устройство было промоделировано и были получены диаграммы, показанные на рис. 3, по которым можно говорить о правильном функционировании устройства согласно описанным выше алгоритмам.

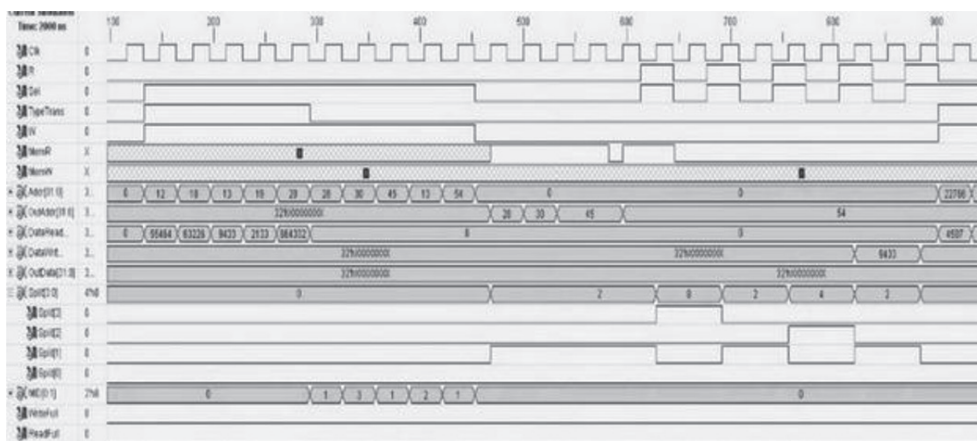


Рис. 3. Временные диаграммы работы АБУ памяти

Выводы

В работе затронуты вопросы функциональной организации аппаратного буферного устройства и его алгоритмов работы. Рассматриваемое устройство отличается от ранее существующих тем, что ранее в МПС задача решалась применением памяти с архитектурой NUMA, либо памяти архитектуры UMA с чередованием адресов, что не позволяло использовать режим расщепления транзакций на ОШ.

В результате использования АБУ, реализованного на ПЛИС, понижается нагрузка памяти, возрастает пропускная способность подсистемы «процессор-память» и быстродействие всей МПС в целом.

Работа выполнена при финансовой поддержке РФФИ (грант № 16-07-00012).

Список литературы

1. Бикташев Р.А., Князков В.С. Многопроцессорные системы. Архитектура, топология, анализ производительности: учеб. пособие. – Пенза: Изд-во Пенз. гос. ун-та, 2004. – 107 с.
2. Мартышкин А.И. Исследование подсистем памяти с буферизацией транзакций на моделях массового обслуживания // XXI век: итоги прошлого и проблемы настоящего плюс. – 2011. – № 3 (03). – С. 124–131.
3. Мартышкин А.И. Математическое моделирование аппаратного буфера памяти многопроцессорной системы // Сборник материалов XII Международной научно-технической конференции «Опико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символической информации. Распознавание-2015». – Курск: Юго-Зап. гос. у-нт, 2015. – С. 247–249.
4. Мартышкин А.И. Реализация аппаратного буфера памяти многопроцессорной системы // Труды XII Международной научно-технической конференции «Новые информационные технологии и системы». – Пенза: ПГУ, 2015. – С. 96–99.
5. Мартышкин А.И. Разработка аппаратного буферного устройства памяти многопроцессорной системы // Фундаментальные исследования. – 2015. – № 12–3. – С. 485–489.
6. Мартышкин А.И. Разработка и исследование разомкнутых моделей подсистемы «процессор-память» многопроцессорных вычислительных систем архитектур UMA и NUMA // Вестник Рязанского государственного радиотехнического университета. – 2015. – № 54–1. – С. 121–126.

7. Мартышкин А.И. Функциональная организация модуля ассоциативного сопроцессора для специализированных вычислительных систем // Модели, системы, сети в экономике, технике, природе и обществе. – 2016. – № 3 (19). – С. 157–166.

8. Суворова Е.А., Шейнин Ю.Е. Проектирование цифровых систем на VHDL. – СПб.: БХВ-Петербург, 2003. – 576 с.

9. Хамакер К., Вранешич З., Заки С. Организация ЭВМ 5-е издание. Перевод с английского О. Здир. – СПб.: Питер; Киев: Издательская группа BHV, 2003. – 848 с.

10. Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем (2-е изд.). – СПб: Питер, 2011. – 688 с.

11. AMBA Specification. Rev 2.0. – ARM Limited, 1999.

References

1. Biktashev R.A., Knjazkov V.S. Mnogoprocessornye sistemy. Arhitektura, topologija, analiz proizvoditel'nosti: ucheb. posobie. Penza: Izd-vo Penz. gos. un-ta, 2004. 107 p.
2. Martyshekin A.I. Issledovanie podsystem pamjati s buferizacijej tranzakcij na modeljah massovogo obsluzhivanija // XXI vek: itogi proshlogo i problemy nastojashhego pljus. 2011. no. 3 (03). pp. 124–131.
3. Martyshekin A.I. Matematicheskoe modelirovanie apparatnogo bufera pamjati mnogoprocessornoj sistemy // Sbornik materialov XII Mezhdunarodnoj nauchno-tehnichesknoj konferencii «Optiko-jelektronnyje pribory i ustrojstva v sistemah raspoznavanija obrazov, obrabotki izobrazhenij i simvolnoj informacii. Raspoznavanje-2015». Kursk: Jugo-Zap. gos. u-nt, 2015. pp. 247–249.
4. Martyshekin A.I. Realizacija apparatnogo bufera pamjati mnogoprocessornoj sistemy // Trudy XII Mezhdunarodnoj nauchno-tehnichesknoj konferencii «Novye informacionnye tehnologii i sistemy». Penza: PGU, 2015. pp. 96–99.
5. Martyshekin A.I. Razrabotka apparatnogo bufernogo ustrojstva pamjati mnogoprocessornoj sistemy // Fundamentalnye issledovanija. 2015. no. 12–3. pp. 485–489.
6. Martyshekin A.I. Razrabotka i issledovanie razomknyh modelej podsystemy «processor-pamjat» mnogoprocessornyh vychislitelnyh sistem arhitektur UMA i NUMA // Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta. 2015. no. 54–1. pp. 121–126.
7. Martyshekin A.I. Funkcionalnaja organizacija modulja associativnogo soprocessora dlja specializirovannyh vychislitelnyh sistem // Modeli, sistemy, seti v jekonomike, tehnike, prirode i obshhestve. 2016. no. 3 (19). pp. 157–166.
8. Suvorova E.A., Shejnin Ju.E. Proektirovanie cifrovyyh sistem na VHDL. SPb.: BHV-Peterburg, 2003. 576 p.
9. Hamaker K., Vraneshich Z., Zaki S. Organizacija JeVM 5-e izdanie. Peregod s anglijskogo O. Zdir. SPb.: Piter; Kiev: Izdatelskaja gruppya BHV, 2003. 848 p.
10. Cilker B.Ja., Orlov S.A. Organizacija JeVM i sistem (2-e izd.). SPb: Piter, 2011. 688 p.
11. AMVA Specification. Rev 2.0. ARM Limited, 1999.