

ФУНКЦИОНАЛЬНАЯ ОРГАНИЗАЦИЯ АППАРАТНОГО АРБИТРА ОБЩЕЙ ШИНЫ МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ¹

А. И. Мартышкин

FUNCTIONAL ORGANIZATION OF HARDWARE ARBITER OF COMMON BUS OF MULTIPROCESSOR SYSTEM

A. I. Martyshkin

Аннотация. *Актуальность и цели.* Одним из путей к определенному решению проблемы сложности решаемых научно-технических задач стало появление современных высокопроизводительных вычислительных систем. Здесь немаловажное значение имеет организация шинного арбитра, который управляет процессом занятия и освобождения общей шины в вычислительных системах. Этим обусловлена актуальность статьи. *Цель работы* – создание, описание и исследование функционирования аппаратного арбитра общей шины многопроцессорной системы. *Материалы и методы.* Для достижения поставленной в работе цели автором применена САПР Web pack ISE фирмы Xilinx, где наряду с языком VHDL можно использовать схемотехнический редактор для создания и моделирования правильности функционирования блока аппаратного арбитра, что облегчает синтез проектов, направленных на использование современной элементной базы – ПЛИС. *Результаты.* Предложен вариант реализации и функционирования аппаратного арбитра. Описан состав предлагаемого устройства. Представлена созданная функциональная схема модуля арбитра, которая синтезирована в САПР Web pack ISE для проверки работоспособности. В качестве подтверждения работоспособности приведены временные диаграммы функционирования компонента выбора ведущего процессора аппаратным арбитрам. *Выводы.* Предполагается реализовать аппаратный арбитра в виде платы расширения с применением ПЛИС. Областью применения устройства являются высокопроизводительные вычислительные системы различной специализации и назначения.

Ключевые слова: аппаратный арбитраж, децентрализованный арбитраж, интерфейс «общая шина», многопроцессорная система, ПЛИС, принцип работы, приоритеты, функциональная схема, централизованный арбитраж.

Abstract. *Background.* One of the ways to solve the problem of the complexity of the scientific and technical problems being solved was the emergence of modern high-performance computing systems. It is important to organize the bus arbiter, which manages the process of occupying and releasing the common bus in computer systems. This is due to the relevance of the article. The purpose of the work is to create, describe and study the functioning of the hardware arbiter of the common bus of the multiprocessor system. *Materials and methods.* To achieve the goal set, the author applied CAD pack Web pack ISE from Xilinx is used, where along with the VHDL language, a circuit designer can be used to create and simulate the correct functioning of the hardware arbiter

¹ Работа выполнена при финансовой поддержке РФФИ (грант № 16-07-00012).

unit, which facilitates the synthesis of projects aimed at using the modern element base – PLD. *Results.* In the article the author suggests an option for the implementation and functioning of the hardware arbiter. The composition of the proposed device is described. The created functional diagram of the arbitrator module, which is synthesized in the CAD Web pack ISE, is shown to test the functionality. As a confirmation of operability, the time diagrams of the functioning of the master processor selection component by the hardware arbiter. *Conclusions.* In conclusion, based on the results of studies and analysis of the further development of the project, the author presents the results and conclusions. It is proposed to implement a hardware arbiter in the form of an expansion board using PLDs. The field of application of the device are high-performance computing systems of various specialization and purpose.

Key words: hardware arbiter, decentralized arbitration, common bus interface, multiprocessor system, PLD, operating principle, priorities, functional scheme, centralized arbitration.

Введение

Вся жизнь современного человека пронизана вычислительной техникой. Пожалуй, нет такой сферы деятельности, где бы ни применялись компьютеры и разнообразная вычислительная техника. В настоящее время проблема производительности вычислительных машин актуальна в том числе и потому, что круг задач, решаемых на ЭВМ, существенно вырос, а также возросла сложность самих решаемых задач. Одним из путей к определенному решению этой проблемы стало появление современных высокопроизводительных [1, 2] и специализированных [3–5] вычислительных систем. Немалое влияние в таких системах имеет организация устройства – арбитра, который управляет процессом занятия и освобождения общей шины. От того, насколько правильно продуман функционал арбитра, зависит быстрдействие вычислительной системы в целом.

Постановка задачи

Целью статьи является разработка и описание функционирования аппаратного арбитра общей шины многопроцессорной системы.

Функциональная организация аппаратного арбитра

В работе [6] при описании структуры производилось проектирование и описание всей вычислительной системы в целом, в состав же функциональной схемы входит и исследуется только модуль аппаратного арбитра шины. Выделим следующие основные блоки, из которых состоит арбитр:

- регистр запросов;
- схема выделения статических приоритетов;
- схема выделения динамических приоритетов;
- регистр приоритетов.

Схема выделения статических приоритетов работает следующим образом.

На регистр запросов поступают запросы от процессоров (BR0–BR15) на захват и занятие шины. Затем они поступают на схему анализа приоритетов, где выделяется запрос с наивысшим приоритетом.

С целью оптимизации и простоты функционирования примем, что выделение наивысшего приоритета происходит по следующей схеме. Статический арбитр присваивает каждому из процессоров системы строго зафиксированный за ним уровень приоритета. В том случае, если одновременно поступило несколько запросов от разных процессоров, арбитр выделяет приоритет по следующему принципу:

- процессор № 0 получает разрешение доступа к шине сразу же после выдачи им запроса (если шина в данный момент свободна);
- процессор № 1 получает разрешение в том случае, если от него есть запрос и в этот момент нет запроса от процессора № 0;
- процессор № 2 получает разрешение в том случае, если от него есть запрос и в этот момент нет запросов от процессоров № 0 и № 1 и т.д.;
- процессор № 3 получает разрешение в том случае, если от него есть запрос и в этот момент нет запросов от процессоров № 0, № 1 или № 2 и т.д.

Этот принцип можно описать следующими выражениями:

$$\text{ПШ}_0 = \text{ЗШ}_0, \quad (1)$$

$$\text{ПШ}_1 = \overline{\text{ЗШ}_0} \text{ЗШ}_1, \quad (2)$$

$$\text{ПШ}_2 = \overline{\text{ЗШ}_0} \overline{\text{ЗШ}_1} \text{ЗШ}_2, \quad (3)$$

...

$$\text{ПШ}_{15} = \overline{\text{ЗШ}_0} \overline{\text{ЗШ}_1} \overline{\text{ЗШ}_2} \overline{\text{ЗШ}_3} \dots \overline{\text{ЗШ}_{13}} \overline{\text{ЗШ}_{14}} \text{ЗШ}_{15}, \quad (4)$$

После анализа всех запросов и выделения из них наивысшего результата работы заносится в регистр приоритетов (регистр ответа). Выходы регистра приоритетов соединены со входами готовности (READY) всех процессоров.

Схема выделения динамических приоритетов работает следующим образом. На регистр запросов поступают запросы на захват шины от процессоров (BR0-BR15). В схеме динамического арбитра имеется 4-разрядный двоичный счетчик, выходы которого соединены с четырьмя входами 4-16-разрядного дешифратора. По каждому сигналу освобождения шины счетчик производит инкремент и передает 4-разрядную выходную последовательность в дешифратор. На соответствующем этой последовательности выходе дешифратора появляется логическая единица, которая и определяет, какому из процессоров выдать разрешение на работу с общей шиной в данном цикле арбитража.

Регистры запросов и ответов являются общими для обеих схем выделения приоритетов. Тот процессор, который получил единичный сигнал на входе READY, получает в свое распоряжение общую шину и может записать данные в общую память и/или прочитать их оттуда согласно алгоритму функционирования. Кроме того, в логике работы обеих этих схем выдачи приоритетов (статических и динамических) принимают участие сигнал состояния шины (свободно/занято) и сигнал выбора режима работы арбитра. Оба эти сигнала задаются программным путем.

Сигнал состояния шины является общим для обеих схем выделения приоритетов. Его участие в работе схемы в том, что пока этот сигнал показы-

вает, что шина занята, арбитр не будет выдавать никаких приоритетов процессорам.

Сигнал выбора режима арбитража служит для того, чтобы в конкретный момент времени работала только одна из схем выдачи приоритетов (статическая или динамическая).

Описываемое устройство работает в соответствии со спецификацией шины AMBA АНВ [7]. На рис. 1 представлена упрощенная схема интерфейса ведущего устройства, ведомого устройства и арбитра на шине AMBA АНВ.

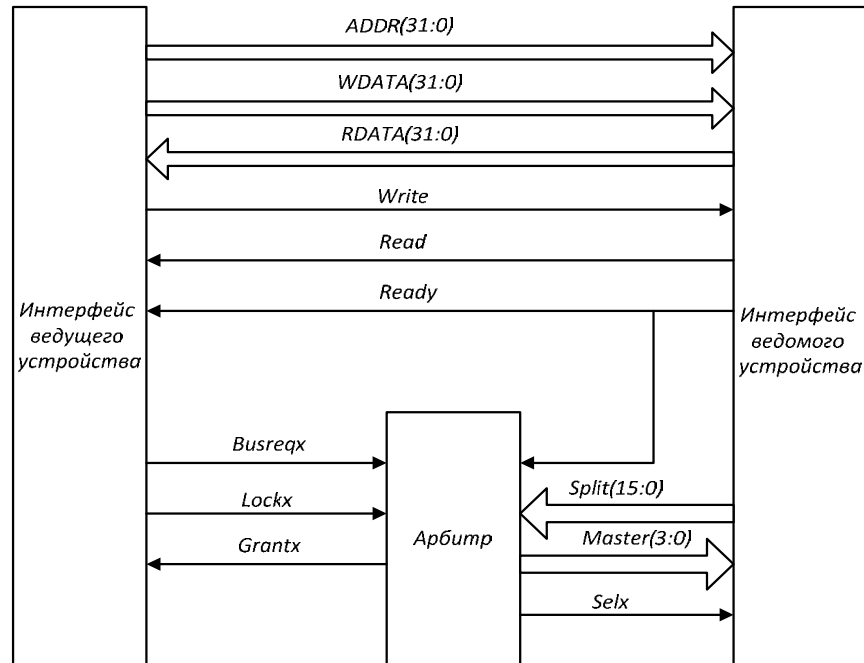


Рис. 1. Интерфейсы ведомого устройства, ведущего устройства и арбитра на шине AMBA АНВ

Шина данных AMBA АНВ разделена на две однонаправленные шины – rdata и wdata. Для ведущих устройств шина rdata является входной (по ней ведущее устройство читает, принимает данные из памяти), а шина wdata – выходной (по ней ведущее устройство пересылает данные для записи в ведомом устройстве). Для ведомых устройств, наоборот, шина rdata является выходной, а шина wdata – входной. Эта логика наименования шин данных понятна, если представить себе в качестве примера ведущего устройства процессор, а в качестве ведомого – память.

Стандартом AMBA АНВ предусматривается возможность различной разрядности шин данных. Протоколом шины определен числовой ряд возможной разрядности шин данных: 8, 16, 32, 64, 128, 256, 512 и 1024 разряда. Разрядностью шин данных считается 32 разряда. Разрядность шин rdata и wdata одинакова.

Сигнал ready используется ведомым устройством для указания его готовности завершить текущий обмен данными. Ведомое устройство определя-

ет завершение первой фазы адреса обмена по сигналу ready = «1». Первая фаза передачи адреса текущего обмена может совпадать с последней фазой передачи данных предыдущего обмена, в котором участвовало другое ведомое устройство. По этой причине, если в системе имеется более одного такого устройства, каждое из них должно иметь и входной сигнал ready для того, чтобы отслеживать состояние выходного сигнала ready от ведомого устройства, которое участвовало в предыдущем обмене.

Отладка аппаратного арбитра, рассматриваемого в статье, осуществлялась в программе Web pack ISE. В качестве способа ввода описания проектируемой схемы на ПЛИС наряду с языком VHDL-1993, посредством которого был описан функционал арбитра, использовался и схемотехнический редактор [6, 7]. Схема внутреннего устройства аппаратного арбитра общей шины, спроектированная на основе VHDL-модулей в программе Web pack ISE, представлена на рис. 2.

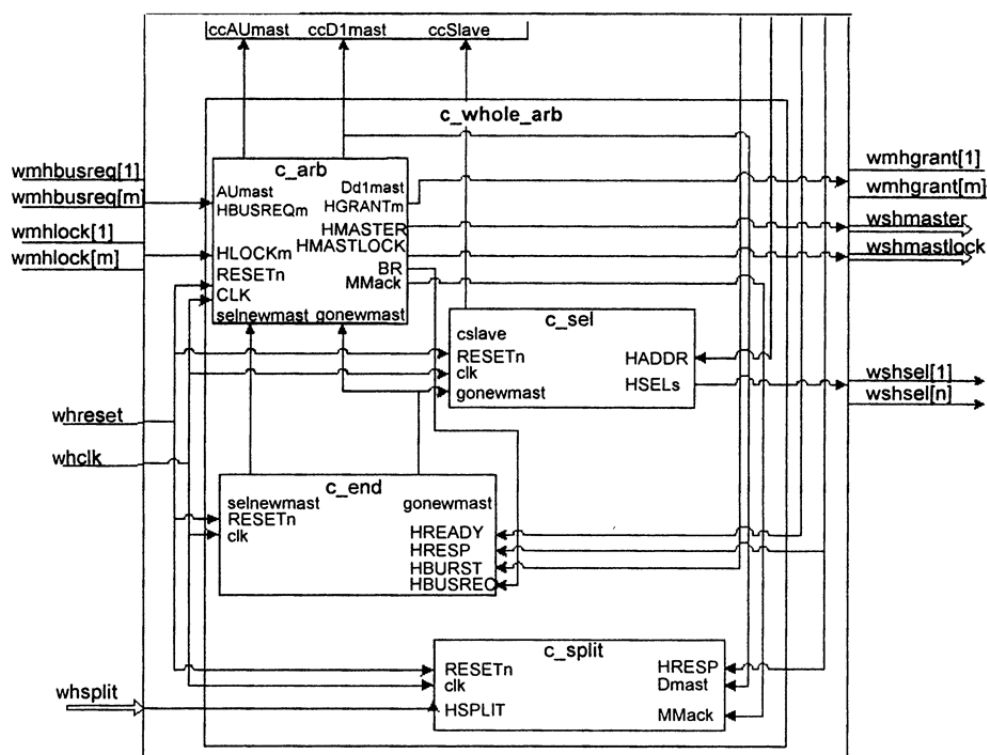


Рис. 2. Схема внутреннего устройства аппаратного арбитра общей шины

На рисунке в виде соединенных между собой блоков показано внутреннее устройство аппаратного арбитра. Основным его модулем является c_whole_arb, включающий в себя модули c_arb (блок выбора ведущего устройства), c_sel (блок выбора ведомого устройства), c_end (блок определения конца запроса) и c_split (блок контроля расщепленных транзакций [7]).

Организация блока выбора главного (ведущего) процессора зависит от используемой в системе схемы приоритетов (относительные, абсолютные,

смешанные и др.). В статье ввиду ограничения объема автором предложен к рассмотрению вариант, при котором процессоры имеют абсолютные приоритеты, определяющиеся их идентификаторами. Процессор с идентификатором 1 имеет наивысший приоритет и является ведущим по умолчанию. Ниже представлены фрагменты VHDL-кода описания процессов компонента выбора ведущего процессора:

```

p_selmast: process (HBUSREQm, MMack)
begin amast: = 1;
for i in Nm downto 1 loop
  If HBUSREQm(i) = '1' and MMack (i-1) = '0' then amast: = i; end if;
end loop;
if amast/= 0 then samast< = amast;
else for i in Nm downto 1 loop
  if MMack(i-1) = '0' then samast< = i;
end if;
end loop;
end if;
samast< = amast;
end process p_selmast;
p_lockmast: process (RESETn, clk, HREADY, gonewmast, selnewmast,
samast)
begin
if RESETn = '0' then AUmast< = 1; saum< = 1;
HMASTER< = '0';HMASTER< = "0001";
else if rising_edge(clk) then
  if HREADY = '1' then
    if gonewmast = '1' then HMASTERLOCK< = intlock;
end if;
    if selnewmast = '1' then
      if samast>0 then AUmast< = samast; saum< = samast;
        else AUmast< = 1;saum< = 1;
end if;
      if samast>0 then intlock< = HLOCKm(samast);
HMASTER< = conv_std_logic_vector(conv_unsigned(samast,4),4);
        else intlock< = HLOCKm(1);
HMASTER< = "0001";
end if;
end if;
end if;
end if;
end if;
end process p_lockmast;
p_BR: process(AUmast, HBUSREQm)
begin BR< = HBUSREQm(AUmast);
end process p_BR;

```

Приведенный выше фрагмент VHDL-кода описывает функционирование блока выбора следующего ведущего процессора, который как кандидат на использование общей шины выбирается каждый раз при изменении на линиях запроса шины или изменении сигнала Mmask. Если selnewmast = «1», то по восходящему фронту сигнала тактирования кандидат на использование общей шины становится владельцем шины адреса и управления. Если gonewmast = «1», то по восходящему фронту сигнала тактирования процессор, владеющий шинами адреса и управления, получает в свое распоряжение шину данных. Сигнал samast используется для хранения идентификатора процессора, выбранного для следующего запроса. Сигнал intlock используется для хранения информации о наличии блокировки в текущем запросе. Сигнал saum используется для хранения идентификатора процессора, который в данный момент времени владеет линиями адреса и управления.

Процесс p_selmast используется для выбора процессора, которому будет предоставлена шина адреса и управления после завершения очередного запроса. Этот процесс выполняется каждый раз, когда происходит изменение на линиях запроса шины от процессоров и изменение сигнала Mmask. Внутри процесса описана переменная amast, используемая при выборе процессора, который будет использовать шину. Сначала этой переменной присваивается значение 0. Далее в цикле просматривается наличие запросов от процессоров, начиная с последнего, имеющего самый низкий приоритет. При каждом выполнении цикла, если имеется сигнал запроса шины, а соответствующий разряд сигнала Mmask = «0» (что указывает на отсутствие для данного процессора незавершенной операции), переменной amast присваивается значение идентификатора этого процессора. Таким образом, после завершения цикла в переменной amast хранится идентификатор процессора, имеющего наивысший приоритет среди запросивших шину. Если значение amast осталось равным 0, значит, в данный момент ни один процессор не запрашивает шину. После того, как выбран процессор – кандидат на использование шины, его идентификатор сохраняется в сигнале samast.

Процесс p_lockmast используется для защелкивания идентификатора процессора, становящегося собственником линий адреса и управления. В этом процессе определяются также значения выходных сигналов HMASTER и HMASTLOCK. При наличии сигнала сброса в качестве ведущего по умолчанию выбирается процессор с номером 1. Если selnewmast = «1», то значение сигнала samast по восходящему фронту импульса фиксируется в saum и Aumast. В соответствии со значением samast определяется значение сигнала HMASTER. Значение сигнала HLOCK от процессора, который становится собственником линий данных и управления, фиксируется в сигнале intlock. При очередном запуске процесса, после того как этот процессор займет шину данных, значение сигнала intlock выдается на HMASTLOCK. Процесс p_BR используется для определения значения сигнала BR – запрос шины того процессора, который в данный момент владеет линиями адреса и управления.

В ходе моделирования описанного компонента получены временные диаграммы работы в точном соответствии приведенному выше описанию функционирования (рис. 3).

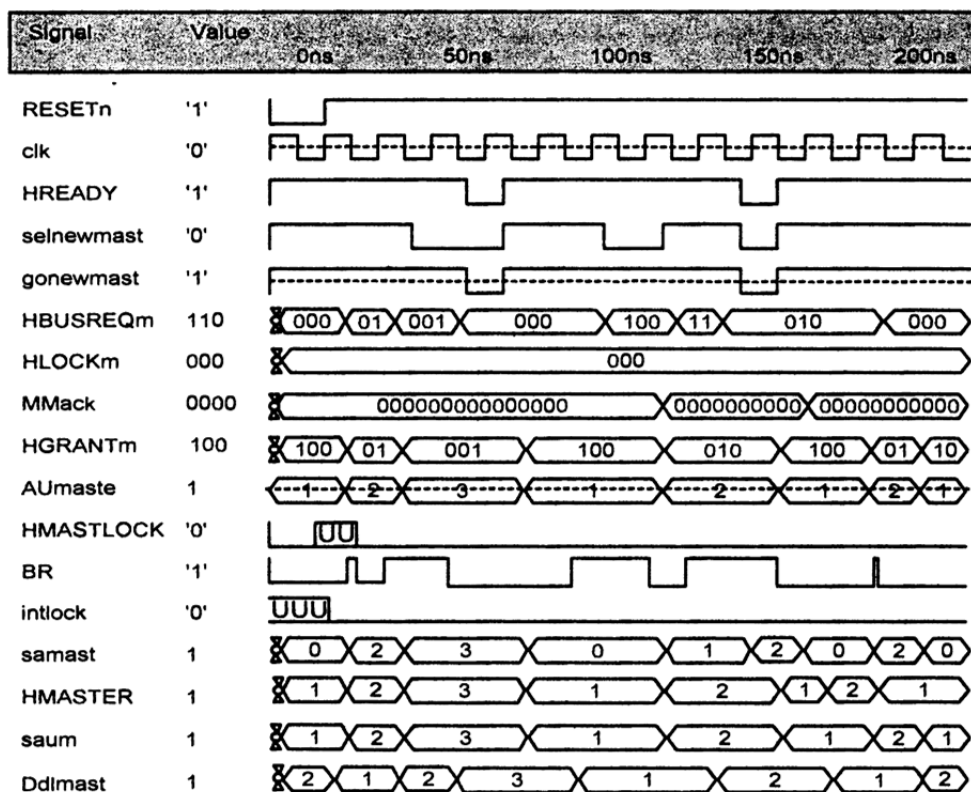


Рис. 3. Временные диаграммы работы компонента выбора ведущего процессора аппаратным арбитром

Заключение

В результате проведенных исследований предложен вариант построения аппаратного арбитра общей шины многопроцессорной системы на функциональном уровне. Исследован принцип его работы. Получены временные диаграммы работы компонента выбора ведущего процессора, подтверждающие работоспособность предложенного решения.

Дальнейшая работа в этом направлении предполагает проектирование и синтез модуля аппаратного арбитра на базе ПЛИС Spartan 3 фирмы Xilinx и реализацию его на плате расширения. Создание устройства на такой ПЛИС обеспечит приемлемое соотношение цена/быстродействие по сравнению с существующими аналогами. Кроме того, предлагаемый аппаратный модуль может быть использован в вычислительных системах среднего быстродействия.

Библиографический список

1. Бикташев, Р. А. Многопроцессорные системы. Архитектура, топология, анализ производительности : учеб. пособие / Р. А. Бикташев, В. С. Князьков. – Пенза : Изд-во ПГУ, 2004. – 107 с.
2. Таненбаум, Э. Архитектура компьютера / Э. Таненбаум, Т. Остин. – 6-е изд. – СПб. : Питер, 2013. – 816 с.

3. Мартенс-Атюшев, Д. С. Разработка и исследование реконфигурируемого вычислительного кластера для цифровой обработки сигнала / Д. С. Мартенс-Атюшев, А. И. Мартышкин // Современные информационные технологии. – 2015. – № 21. – С. 190–195.
4. Мартышкин, А. И. Функциональная организация и алгоритмы работы аппаратного буферного устройства памяти многопроцессорной вычислительной системы / А. И. Мартышкин // Фундаментальные исследования. – 2016. – № 12-3. – С. 518–522.
5. Мартышкин, А. И. Разработка аппаратного буферного устройства памяти многопроцессорной системы / А. И. Мартышкин // Фундаментальные исследования. – 2015. – № 12-3. – С. 485–489.
6. Мартышкин, А. И. Варианты реализации и структура аппаратного арбитра общей шины многопроцессорной системы / А. И. Мартышкин // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 2 (22). – С. 195–204.
7. Суворова, Е. А. Проектирование цифровых систем на VHDL / Е. А. Суворова, Ю. Е. Шейнин. – СПб. : БХВ-Петербург, 2003. – 576 с.

Мартышкин Алексей Иванович

кандидат технических наук, доцент,
кафедра вычислительных машин и систем,
Пензенский государственный
технологический университет
E-mail: alexey314@yandex.ru

Martyshkin Alexey Ivanovich

candidate of technical sciences,
associate professor,
sub-department of computers and systems,
Penza State Technological University

УДК: 004.31

Мартышкин, А. И.

Функциональная организация аппаратного арбитра общей шины многопроцессорной системы / А. И. Мартышкин // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 3 (23). – С. 151–159.