Звонарева Г.А.¹, Конец М.Ю.², Тухбатуллин В.В.¹
Организация многоканальной передачи данных в специализированной
вычислительной системе

¹Московский авиационный институт (НИИУ)

²АО «НПО «Электронное приборостроение»
(Россия, Москва)

doi: 10.18411/sr-10-04-2019-71

idsp: sciencerussia-10-04-2019-71

Аннотация

Рассматриваются аспекты разработки программного обеспечения для организации многоканальной передачи данных от множества параллельно обрабатывающих информацию цифровых сигнальных процессоров серии TMS320C64xx со сверхдлинным набором команд к ядру ЭВМ специального назначения через магистраль VME. При этом представляется организация блочного метода передачи по прерываниям и потокового метода под управлением контроллера прямого доступа к памяти в условиях необходимости решения задачи диспетчеризации запросов на ввод/вывод как в пределах одного агрегирующего несколько сигнальных процессоров узла с помощью контроллера Ad6414, так и задачи арбитража магистрали VME между данными узлами посредством контроллера Tundra Tsi148. Отладка и тестирование программного обеспечения проводилась в среде разработчика Borland C++ Builder 6.0 с использованием набора разработки драйверов для Windows окружения Driver Development Kit 7.1.0 и WinDbg и на аппаратно-программном комплексе.

Ключевые слова: многоканальная передача данных, ЭВМ специального назначения, блочный метод передачи данных по прерываниям, потоковый метод передачи данных под управлением контроллера прямого доступа к памяти, DSP, VME.

Abstract

Herein are presented development aspects of software for organization of multichannel data transmission from a parallel processing set of TMS320C64xx generation digital signal processors based on Very Long Instruction Word architecture to random-access memory of a specialized computational facility via a VME bus. On top of it, the organization of block-oriented transmission method with usage of interrupts and stream-oriented transmission method under control of Direct Memory Access controller have been reviewed in the conditions of the need for the input/output request dispatching both within the one board aggregating several signal processors under control of Ad6414 controller and within the task of VME bus arbitration of such boards by means of Tundra Tsi148 controller. Developed software was debugged and tested in Borland C Builder 6.0 developer environment with usage of Windows Driver Development Kit 7.1.0 and WinDbg and on a hardware-software complex.

Keywords: multichannel data transmission, specialized computational facility, block-oriented transmission method with usage of interrupts, stream-oriented transmission method under control of Direct Memory Access controller, DSP, VME.

Список основных специальных терминов и сокращений

DSP (англ. Digital Signal Processor) – цифровой процессор обработки сигналов (ЦПОС);

VME (англ. Versa Module Europa bus) – стандарт организации шины передачи данных;

DMA (англ. Direct Memory Access) – прямой доступ к памяти;

PnP (англ. Plug And Play) – технология, предназначенная для быстрого определения и конфигурирования устройств в компьютере;

IRP (англ. Input/Output Request Packet) – пакет запроса на ввод/вывод;
IRQL (англ. Interrupt Request Level) – механизм программно-аппаратной приоритизации;
PDO (англ. Physical Device Object) – объект физического устройства;
FDO (англ. Functional Device Object) – объект функционального устройства;
HAL (англ. Hardware Abstraction Layer) – слой аппаратных абстракций;
ПО – программное обеспечение;
ЦП – центральный процессор;
ОП – оперативная память.

Современные вычислительные системы с многоканальной передачей информации на базе технологий цифровой обработки сигналов находят все большее применение при решении задач специального назначения. При этом программная и аппаратная архитектура данных систем определяется характерным образом необходимостью соответствия предъявляемым к ним требованиям, а также напрямую связана с самим технологическим процессом параллельного формирования, обработки и передачи потоковых данных от множества цифровых источников к главному узлу специализированной ЭВМ за конечный промежуток времени. Успешное выполнение данного процесса обеспечивается последовательным решением специфических подзадач, реализуемых конкретными этапами обработки информации на протяжении всего тракта данных (канала).

Применение универсально программируемых специализированных DSP в качестве данных цифровых источников позволяет обеспечить выполнение всех характерных потребностей распределенного процесса подготовки информации на начальных этапах ее обработки в виде разделенных независимых потоков данных. Необходимость обработки большого количества таких потоков обуславливает трудности их вывода на центральный узел сбора информации через общую шину, ведя к аппаратному усложнению вычислительной системы за счет ввода промежуточных узлов контроля передаваемой информацией в виде отдельных контроллеров на тракте данных. Усложненная инфраструктура выдвигает новые требования по синхронизации взаимодействия DSP источника и приемника, централизованно осуществляющего последующую обработку цифровой информации на процессорном модуле ЭВМ специального назначения на основе комплексного программного обеспечения. Таким образом, при разработке данного ПО поднимаются не только вопросы программного представления процесса передачи данных, но и организации многоуровневой иерархии доступа с учетом арбитража и аппаратных особенностей на соответствующих уровнях-контроллерах при помощи событий-прерываний, вопросы реализации блочного метода передачи с использованием прерываний под управлением ЦП и потоковой передачи под управлением специализированного контроллера прямого доступа к памяти с учетом характеристик шин. Существенной к рассмотрению становится тема организации взаимосвязи ПО уровня пользователя и драйверов данных контроллеров.

Решение задачи сбора данных, сопряженной с решением проблем параллельных вычислений таких, как синхронизация при использовании данных и многоуровневый арбитраж, также, как правило, подразумевает необходимость исполнения требования согласованных временных характеристик системы, обеспечивающего эффективным методом управления потоками данных через соотнесение параметра пропускной способности центральной магистрали с требованием того, что все операции на процессорном модуле ЭВМ специального назначения должны быть выполнены за время, которое бы не превышало период дискретизации поступающего на нее сигнала.

Структурная схема специализированной вычислительной системы с многоканальной передачей данных представлена на рисунке 1.

Рисунок 1. Структурная схема специализированной вычислительной системы с многоканальной передачей данных

Данная система предназначена для сбора информации с нескольких десятков параллельно обрабатывающих DSP серии TMS320C64xx со сверхдлинным набором команд для ее последующей передачи в оперативную память процессорного модуля ЭВМ специального назначения через магистраль VME. Одновременно с этим организуется двухуровневая схема арбитража для образованного тракта данных:

- диспетчеризация, обеспечивающая согласованную работу ряда устройств в пределах одного DSP узла и проводимая локальным контроллером Ad6414;
- решение конфликтных ситуаций, возникающих при регулировании процесса распределения приоритетов передачи по магистрали VME между данными узлами, которое осуществляется контроллером Tundra Tsi148.

Разработанное ПО принадлежит и встраивается в данный комплекс, обеспечивая выполнение его задач, и обуславливая таким образом свою актуальность. При этом оно строится на базе активного использования имеющихся, разработанных ранее драйверов устройств Tundra Tsi148 и Ad6414 при помощи такого компонента ОС как менеджер ввода/вывода. Менеджер ввода/вывода является работающим в привилегированном режиме ядра модулем, встроенным в вертикальную иерархию взаимодействия между пользовательским ПО и интерфейсами доступа к устройствам, предоставляемыми данными драйверами. Комплекс ПО реализует и соответствует данному разделению, разбиваясь на ПО пользовательского уровня привилегий (рисунок 2) и ПО системного уровня привилегий (режима ядра) (рисунок 4) в соответствии с используемым ОС Windows архитектурным механизмом информационной безопасности и функциональной отказоустойчивости – кольцами защиты RING3 и RING0 соответственно.

Рисунок 2. Схема алгоритма пользовательского уровня ПО

Разработанное ПО предусматривает реализацию 2 способов передачи данных от сигнальных процессоров на процессорный модуль ЭВМ специального назначения, использование которых обосновывается требованием исполнительного характера о предъявляемой роли к ЦП или контроллеру Tundra Ts148 как регулирующим механизм передачи:

- блочный метод передачи информации с использованием прерываний под управлением центрального процессора процессорного модуля специализированной ЭВМ основан на разбиении исходного сообщения на блоки равные ширине выборки в соответствии с шириной шины данных VME по 32 бита. Блоки последовательно передаются после генерации запроса на прерывание, направленного от DSP и регистрируемого ЦП, который, в свою очередь, берет управление над данным процессом передачи;
- потоковый метод передачи информации под управлением контроллера DMA, когда те же действия по управлению осуществляются встроенным в контроллер Tundra Ts148 заранее программируемым блоком DMA после регистрации сгенерированного от контроллера Ad6414 события запроса на передачу из памяти DSP в заданную область ОП процессорного модуля ЭВМ специального назначения.

ПО пользовательского уровня для обоих методов реализуется одинаковым образом. Оно выполняет задачу организации общего алгоритма работы всей системы, обеспечивая логикой выполнения «запрос на передачу – ответные действия ЦП», включает цифровую обработку на ЦП и отображение результатов пользователю, регулирует процесс передачи запросов, конфигурирующих работу контроллеров Tundra Ts148 и Ad6414: подключение/отключение мониторов запросов на передачу от конкретного DSP, чтение/запись массивов данных с DSP. Данное регулирование происходит через сообщения унифицированного формата, которые описывают необходимое к выполнению действие параметрами кода действия, адресов чтения-записи, номера узла (дескриптора), номера DSP, длины передаваемых буферных массивов. ПО пользовательского уровня выполняет работу тандемного характера с обращением к исполнительным службам ОС Windows системного уровня данными сообщениями

ввода/вывода. При этом организация согласованной работы верхнего и нижнего ПО осуществляется централизованным механизмом менеджера ввода/вывода по передаче драйверам запросов на ввод/вывод при помощи асинхронного высокопроизводительного пакетного механизма IRP. Наличие такого универсального интерфейса взаимодействия способствует переносу ПО уровня пользователя на системы с иной архитектурой.

Разработанное ПО пользовательского уровня также включает программную организацию механизма приема информации с циклических буферов DSP RAM памяти, что является одним из способов синхронизации при доступе управляющих процессов к используемым исходным массивам информации через блочно-ориентированный механизм передачи. Циклическая буферизация данных DSP, наряду с аппаратными средствами контроллеров Ad6414 и Tundra Tsi148 и программно настраиваемой буферизацией данных, осуществляемой ОС при использовании рабочих процедур менеджера ввода/вывода и IRP-механизма, поддерживают согласование временных характеристик при переносе информации с DSP на процессорный модуль ЭВМ специального назначения, нивелируя несоответствие скоростей записи и чтения используемых устройств системы на тракте передачи данных.

Характерной особенностью работы пользовательского ПО является использование прозрачного для своего уровня механизма адресации DSP объединенных единым виртуальным адресным пространством (рисунок 3).

Рисунок 3. Адресация на узлах с DSP устройствами

Проблема согласования доступа к памяти DSP решается средствами драйверов устройств Ad6414 и Tundra Tsi148 через вызов соответствующих функций пользовательского уровня ПО. Взаимодействие данного ПО со стратегией распределения виртуальной памяти соответствующей компонентой ОС Windows менеджером виртуальной памяти на основе определяющей механизм трансляции адресов работы с PnP таблицей ресурсов устройств (описываемой реестром ОС) позволяет организовывать однородное отображение физических DSP адресов в пользовательское пространство исполняющегося ПО, что является принципиальным в процессе считывания информации с DSP. Данный процесс учитывает особенности созданного с их помощью единого адресного пространства, файлового механизма представления устройств (физические устройства распознаются пользовательскими приложениями в качестве именованных файловых объектов с символьными ссылками) и включает последовательное преобразование виртуальных адресов в физические. Обеспечение данным однотипным виртуальным доступом ко всем DSP устройствам не только способствует изоляции процессов пользовательского уровня от процессов режима ядра (что в целом соответствует многокомпонентной иерархической практике организации ОС Windows), но и обеспечивает защиту независимого виртуального пользовательского пространства процесса, предоставляет возможности масштабируемости системы за счет гибкого включения своих адресных пространств новых модулей-устройств DSP.

На рисунке 4 изображена реализация ПО системного уровня, основной задачей которого является выполнение функций администрирования унифицированных системных вызов контроллеров Ad6414 и Tundra Tsi148 по управлению конкретным соответствующим физическим устройством через перехват возникающих запросов на ввод/вывод информации DSP и их интерпретацию при помощи менеджера ввода/вывода в высокоуровневые файловые команды.

Рисунок 4. Схема алгоритма системного уровня ПО

Программная реализация предусматривает разбор запросов пользовательского приложения драйвером контроллера Ad6414 для последующего расширения уже созданной структуры адресов DSP и массивов определенной буферной структуры с помощью конфигурирования добавляемых характеристик передачи (таких как код записи/чтения Tundra Tsi148, установка типа адресации и используемой ширины выборки данных VME, установка кода использования роли регулирующего механизма передачи по VME для потокового метода под управлением DMA или блочной передачи под управлением ЦП процессорного модуля ЭВМ специального назначения). Данное расширенное сообщение передается в виде IRP нижележащему драйверу Tundra Tsi148, который в свою очередь выполняет запрограммированный таким образом процесс непосредственного перемещения данных с DSP на процессорный модуль, транслируя принятые в обобщенной форме запросы в формат, понятный для связанных шин ввода/вывода, адресации с последующим уведомлением о результате, и реализуя механизмы блочной передачи и DMA. Передача IRP осуществляется в рамках организованного стека драйверов Ad6414 и Tundra Tsi148 под управлением защищенной подсистемы, реализующей программный интерфейс Win32, при этом работа в нескольких слоях драйверной модели устройств также предполагает необходимость программирования набора процедур их взаимодействия для стандартной поэтапной обработки на соответствующем уровне-устройстве. Этапы алгоритма метода блочной передачи на рисунке 4 помечены флагом «PIO», а блоки программной реализации потокового метода – флагом «DMA». Различие данных блоков кода определяется передаваемыми параметрами вызываемым ими функций: кодовая основа функций блочной передачи описывает явным образом действия ЦП после получения последовательно обработанного на контроллерах и переданного ему запроса на прерывание для пересылки информации через VME (ЦП проводит разбиение на блоки, уменьшает счетчик количества переданных блоков), в то время как вызываемые функции потокового метода передачи используют специальное окно управления определенной

структуры [1] для программирования DMA контроллера, ведущего запись в ОП по определенным адресам своими аппаратными средствами без прерывания ЦП.

ПО, изображенное на рисунке 4, включается в программный комплекс системного уровня на основе разработанных ранее драйверов, используя при реализации необходимых аспектов организации многоканальной передачи данных. К этому относится реализация аппаратных процедур подключения/отключения шинных устройств PDO и функциональных устройств FDO Ad6414 и Tundra Tsi148 на основании дерева устройств, строящегося в ходе взаимодействия PnP менеджера и драйверов, что способствует аппаратной расширяемости системы в случае ввода в эксплуатацию новых параллельно обрабатывающих DSP/DSP-узлов. Процесс построения дерева устройств выполняется, с одной стороны, с помощью создания соответствующих необходимых для работы с контроллерами на пользовательском уровне файловых объектов устройств (предоставление дескрипторов), стеков устройств – на системном при помощи PnP менеджера и организуемым таким образом стеком драйверов Ad6414 и Tundra Tsi148, а с другой, производит создание символьных ссылок для подсистемы Win32 на соответствующие имена файловых объектов устройств. При этом на данном уровне была программно определена обработка стандартных IRP запросов, PnP запросов, запросов на управление электропитанием, их передача через все стеки устройств PnP дерева до достижения целевого PDO и ассоциированного с ним драйвера. Другой выполняемой задачей данного комплекса является реализация инфраструктуры взаимодействия с внутренними модулями ОС Windows (такими, как менеджеры ввода/вывода, PnP, электропитания) и с механизмами организации единого виртуального адресного пространства с помощью драйверов и менеджера виртуальной памяти. Разработанное ПО также включается в процесс обработки запросов на прерывание, рассматриваемой в свете согласованной по времени и приоритету передаче данных по тракту данных при двухуровневой схеме арбитража на контроллерах Ad6414 и Tundra Tsi148. При этом взаимодействие всего комплекса системного уровня с аппаратной частью данных контроллеров, включая DMA, активно происходит через их управляющие регистры. Примером такого взаимодействия является регулирование динамической трансляции адресов аппаратными средствами моста «PCI-X to VME», программный код которого определен соответствующими процедурами драйвера Tundra Tsi148.

Каждая часть ПО, пользовательская и системная, выполняет свойственные для своего уровня задачи, только их взаимодействие организует многоканальную передачу данных, совокупный алгоритм работы которой описывается на основании цикла обмена информацией между сигнальными процессорами и процессорным модулем ЭВМ специального назначения. Алгоритм формализуется и был программно реализован следующими этапами:

1. Подключение мониторов событий запросов на передачу данных от параллельно обрабатывающих DSP на процессорном модуле ЭВМ специального назначения.

На данном этапе происходит программная инициализация узлов и DSP устройств, построение PnP дерева подключенных устройств, загрузка управляющих последовательностей контроллеров Ad6414 и Tundra Tsi148, создание единого адресного пространства DSP, программное включение возможности регистрации событий запросов на передачу.

2. Генерация запроса на передачу данных от DSP.

По выполнению этапа 1 формируется инфраструктура для организации параллельных каналов ввода/вывода, обеспечивающая возможностью передачи данных от DSP устройств к процессорному модулю ЭВМ специального назначения в рамках зарегистрированной операционной системой архитектуры соединений устройств. На данном этапе становится принципиальной обработка запросов на прерывание, ассоциируемых с драйверными структурами определенного формата и генерируемых на соответствующих аппаратных уровнях контроллеров. При этом драйверы взаимодействуют с входящим в состав менеджера ввода/вывода системным модулем

обращения к диспетчеру прерываний через вызываемые функции, что необходимо для начальной обработки запросов на прерывание и формировании их корректной адресации.

Таким образом, передача данных начинается с генерации запроса на прерывание DSP устройством посредством установки бита прерывания на свой соответствующий порт, что, в свою очередь, фиксируется контроллером Ad6414 [2]. Обработка данного запроса на Ad6414 осуществляется согласно FIFO дисциплине обработки запросов на прерывание через его помещения в список LIST_ENTRY. Список сохраняется до разрешения передачи данных, что сопровождается генерацией лишь одного запроса на прерывание от Ad6414 к контроллеру Tundra Tsi148.

Последующая обработка запросов на прерывание от множества DSP узлов (от контроллеров Ad6414) выполняется Tundra Tsi148 в соответствии с механизмом программно-аппаратной приоритизации IRQ, применяемым при синхронизации приоритетов исполняемых на ЦП процессорного модуля кодов. IRQ присваивается ОС на предыдущем этапе каждому из DSP узлов. При этом в зависимости от загруженной управляющей последовательности Tundra Tsi148 запрос на прерывание приведет к передаче контроля над управлением пересылкой данных либо ЦП процессорного модуля ЭВМ специального назначения, либо DMA контроллеру. Для случая ЦП обработка незамаскированных запросов на прерывание выполняется по относительной схеме (когда исполняемая на ЦП программа не прерывается запросом с более высоким приоритетом, что достигается маскированием сигналов прерывания на время обработки прерывания) и сопровождается сохранением слова состояния (полностью описывающим текущий выполняемый процесс на момент прерывания наряду со счетчиком команд и регистром флагов), сохранением регистров общего назначения ЦП, загрузкой адреса программы-обработчика (указываемого вектором прерывания), и по окончании выполнения программы обработки запросов на прерывание – восстанавливающим процессом обратной последовательности.

3. Разрешение передачи данных для DSP.

На данном этапе происходит установка на соответствующих регистрах контроллеров Tundra Tsi148 и Ad6414 значений разрешения процесса передачи. При этом учитываются такие аспекты как очередность созданного Ad6414 списка запросов FIFO, готовность к работе заранее запрограммированного DMA контроллера.

4. Передача данных из памяти DSP в ОП процессорного модуля ЭВМ специального назначения.

Заключительный этап сопровождается работой циклов блочной пересылки по VME [3] с характерным использованием шин данных, адреса, арбитража (и их освобождением). Управление данным процессом производится при помощи ПО системного уровня через неявное (передаваемое более глубоким слоям как HAL) определение параметров передачи.

Разработка ПО для организации многоканальной передачи данных в специализированной вычислительной системе велась с использованием наработанного ПО богатого инструментария языков программирования высокого уровня C и C++. ПО пользовательского уровня разрабатывалось в среде Borland C++ Builder 6.0, а разработка ПО уровня ядра осуществлялась с использованием набора из средств разработки драйверов окружения Windows Driver Development Kit 7.1.0 и WinDbg. Отладка производилась как в автономном программном режиме, подтвердившим правильность исполнения отдельных программных модулей, отсутствия ошибок компиляции, так и на аппаратно-программном комплексе с различным количеством DSP узлов, обеспечивавшим тестовыми данными, что также подтвердило корректность исполнения управляющего программного комплекса в условиях близких к реальным. При этом использование ОС Windows в качестве основной среды выполнения ПО обуславливается и является допустимым текущим НИОКР этапом разработки всей вычислительной системы специального назначения.

На основании вышеизложенного можно сформулировать следующие выводы:

1. Решена задача сбора и передачи информации с большого количества параллельно обрабатывающих цифровых процессоров обработки сигналов

на процессорный модуль через общую магистраль VME как с одного узла DSP, так и с нескольких, в рамках чего было выполнено:

- разработаны и реализованы алгоритмы пользовательского и системного уровней для блочной передачи данных при использовании прерываний;
- разработаны и реализованы алгоритмы пользовательского и системного уровней потокового метода передачи данных.

2. Разработаны и реализованы алгоритмы для решения задачи организации многоуровневой иерархии доступа с учетом двух вложенных уровней арбитража на основе существующего программного обеспечения драйверов управляющих контроллеров Ad6414 и Tundra Tsi148.

Результат работы представляет из себя программный продукт, включенный в состав специализированной вычислительной системы и прошедший тестирование в АО «НПО «Электронное приборостроение».

1. Tsi148 PCI/X-to-VME Bus Bridge Programming Manual. URL: https://www.slac.stanford.edu/grp/lcls/controls/global/hw/users_guides/cpu/pci-bridge/Tsi148.pdf (дата обращения: 30.03.2019).
2. TMS320C64x/C64x+ DSP CPU and Instruction Set. Reference Guide. URL: <http://www.ti.com/lit/ug/spru732j/spru732j.pdf> (дата обращения: 30.03.2019).
3. ГОСТ Р МЭК 821-2000. Магистраль микропроцессорных систем для обмена информацией разрядностью от 1 до 4 байтов (магистраль VME). URL: <http://vak.ru/pub/gost/gost-r-mek-821-2000.pdf> (дата обращения: 30.03.2019).