

Устройства серии RDMA. Руководство по программированию.

*Модули ЦОС серии RDMA.***Устройства ЦОС серии RDMA.****Руководство по программированию.****Версия 1.2.**

Аннотация

Устройства ЦОС серии RDMA имеют модульную конструкцию, позволяющую адаптировать их для различных задач. В этом документе содержится информация, необходимая для разработки программного обеспечения для процессоров TMS430C6416, расположенных на плате устройства и использовании библиотеки “DSP64162.DLL” в среде Microsoft Windows.

Документ предназначен для специалистов, разрабатывающих встроенное и управляющее программное обеспечение для модулей семейства RDMA.

Содержание

1	ОБЩИЕ СВЕДЕНИЯ	4
2	УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	4
3	КРАТКОЕ ОПИСАНИЕ RDMA	5
3.1	Структурная схема RDMA	5
3.1.1	Схема управления	5
3.1.2	Схема передачи данных	6
3.1.3	Схема обработки прерываний	7
3.2	Устройство FIFO	8
4	БИБЛИОТЕКА УПРАВЛЕНИЯ DSP64162	9
4.1	Общие сведения	9
4.2	Работа с библиотекой	10
4.2.1	Подключение библиотеки к проекту	10
4.2.2	Подключение к устройству	10
4.2.3	Инициализация устройства	10
4.2.4	Загрузка программ TMS320C6416	10
4.2.5	Загрузка прошивок ПЛИС	11
4.2.6	Доступ в адресное пространство TMS320C6416	11
4.2.7	Обработка прерываний	11
4.2.8	Генерация прерывания для TMS320C6416	12
4.2.9	Передача данных через PCI с использованием DMA	12
4.2.10	Получение справочной информации	12
4.2.11	Работа с submodule PIPU	12
4.2.12	Работа с submodule опорного генератора	12
4.2.13	Работа с мезонином ADC	12
4.2.14	Работа с цифровыми приемниками	13
4.2.15	Доступ в адресное пространство мезонинов	13
4.3	Перечень функций	14
4.3.1	Функции инициализации	14
4.3.2	Функции доступа к DSP через окна	18
4.3.3	Функции потокобезопасного доступа к DSP	20
4.3.4	Функции управления DSP	22
4.3.5	Функции загрузки ПЛИС	26
4.3.6	Функции сигнализации	29
4.3.7	Функции работы с контроллером DMA	30
4.3.8	Функции работы с мезонином ADC	32
4.3.9	Функции работы с submodule PIPU	34
4.3.10	Функции работы с submodule RF	36
4.3.11	Функции работы с DDC	39
4.3.12	Функции доступа к мезонином	42
5	ПРОГРАММА МОНИТОР DSP6416-2	45
5.1	Назначение программы	45
5.2	Управление модулем RDMA	45

5.2.1	Общие функции.....	45
5.2.2	Управление сигнальными процессорами.....	45
5.2.3	Управление DDC.....	45
5.2.4	Сигнал с АЦП.....	46
5.2.5	Сигнал с DDC.....	47
5.2.6	Управление submodule PPU.....	48
5.3	Проверка функционирования RDMA	49
5.3.1	Экспресс-проверка	49
5.3.2	Проверка HOST интерфейса TMS320C6416.....	49
5.3.3	Проверка встроенной памяти SDRAM.....	49
5.3.4	Проверка контроллеров PCIDMA	49
5.3.5	Проверка интерфейсов EMIFA, EMIFB TMS320C6416.....	49
5.3.6	Проверка последовательных портов TMS320C6416.....	49
5.3.7	Проверка мезонина ADC-8-14.....	50
5.3.8	Проверка submodule PPU.....	50
5.3.9	Проверка мезонинов PDDC.....	50
6	РАБОТА С ПРОЦЕССОРАМИ TMS320C6416.....	51
6.1	Общие сведения.....	51
6.2	Примеры программ.....	51
6.2.1	Простейшая программа	51
6.2.2	Использование системы прерываний PCI.....	51
6.2.3	Использование контроллера DMA на шине PCI.....	52
6.3	Работа с устройствами на плате RDMA.....	54
6.3.1	Управление FIFO	54
6.3.2	Считывание данных.....	55
6.3.3	Доступ в адресное пространство мезонинов PDDC	56
6.3.4	Использование последовательных портов.....	57
6.3.5	Использование внутрисхемных эмуляторов.....	57
7	РАБОТА С DDC	58
7.1	Общие сведения.....	58
7.2	Рекомендации по настройке DDC ISL5216	58
7.2.1	Общие сведения	58
7.2.2	Работа с программой конфигурации.....	58
7.2.3	Ограничения.....	58
7.2.4	Доступ к данным.....	59
7.3	Рекомендации по настройке DDC AD6636.....	60
7.3.1	Общие сведения	60
7.3.2	Работа с программой конфигурации.....	60
7.3.3	Ограничения.....	60
7.3.4	Доступ к данным.....	61
8	ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ	62
	ПРИЛОЖЕНИЕ I. АДРЕСНОЕ ПРОСТРАНСТВО DSP.....	63
	ПРИЛОЖЕНИЕ II. АДРЕСНОЕ ПРОСТРАНСТВО МЕЗОНИНОВ.....	66
	ПРИЛОЖЕНИЕ III. ИДЕНТИФИКАЦИОННЫЕ КОДЫ ФУНКЦИОНАЛЬНЫХ УЗЛОВ	69
	СПИСОК ЛИТЕРАТУРЫ.....	71
	СПИСОК ИЗМЕНЕНИЙ.....	72
	Список изменений в версии 1.2	72
	ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ.....	73
	ВАЖНЫЕ ЗАМЕЧАНИЯ.....	74

1 Общие сведения

Данное руководство описывает программное обеспечение, предназначенное для управления семейством устройств цифровой обработки сигналов RDMA (далее RDMA) и поясняет процесс создания пользовательского программного обеспечения (ПО). Поставляемое в комплекте с RDMA ПО состоит из драйверов для операционных систем Windows 2000 и Windows XP, динамически подключаемой библиотеки “DSP64162.DLL”, программы-монитора, примеров применения и документации.

Данное программное обеспечение позволяет в полной мере использовать возможности RDMA.

Разработчик, использующий данное программное обеспечение, должен обладать навыками работы в ОС Microsoft® Windows™, программирования в среде WIN32, базовыми знаниями о языке программирования C. Раздел 4 предполагает наличие навыков использования динамически подключаемых библиотек (DLL) и функций обратного вызова.

Проектирование программного обеспечения для сигнальных процессоров TMS320C6416, установленных в RDMA, требует знания конструкции и принципа работы RDMA [1] и сигнальных процессоров серии TMS320C6416 [2].

Рекомендуется так же ознакомиться и с другими материалами, приведенными в списке литературы.

2 Установка программного обеспечения

Установка программного обеспечения производится путем запуска на исполнение файла “setup.exe” из корневого каталога прилагаемого компакт-диска. В файле “install.txt” в корневом каталоге прилагаемого компакт-диска могут находиться дополнительные сведения о процессе установки, не вошедшие в данное руководство.

3 Краткое описание RDMA

Полное описание конструкции модулей семейства RDMA приведено в [1]. В этом разделе приведено краткое описание с точки зрения вопросов взаимодействия аппаратуры и пользовательского программного обеспечения RDMA.

3.1 Структурная схема RDMA

RDMA может поставляться в различной конфигурации, которая определяется типом установленных дополнительных субмодулей (мезонинов). Таким образом, наличие тех или иных функциональных блоков внутри RDMA (FIFO, прерывания) и распределение адресного пространства могут существенно меняться. Однако основные принципы управления мезонинами, передачи потоков данных и обработки прерываний остаются неизменными.

3.1.1 Схема управления

Упрощенная структурная схема управления основными функциональными узлами RDMA приведена на рисунке 3.1. RDMA может поставляться в различной конфигурации, которая определяется типом установленных дополнительных мезонинов. Таким образом, наличие тех или иных функциональных блоков внутри RDMA (FIFO, прерывания) и распределение адресного пространства могут существенно меняться. Однако основные принципы управления мезонинами, передачи потоков данных и обработки прерываний остаются неизменными.

Управление всеми устройствами осуществляется через контроллер PCI, который обеспечивает работу RDMA на шине PCI версии 2.1 с частотой 33 МГц.

Каждый из двух сигнальных процессоров TMS320C6416 подключен к контроллеру PCI через свой HOST-порт. Контроллер PCI формирует для каждого процессора два окна в адресном пространстве PCI. Протяженность каждого окна составляет 8 Мбайт. На это окно может быть отображен любой участок внутреннего адресного пространства, соответствующего TMS320C6416, благодаря чему для доступа во внутреннее адресное пространство TMS320C6416 достаточно выполнить чтение или запись по соответствующему адресу в памяти компьютера.

Такой подход позволяет обойтись без вызовов драйвера RDMA при доступе к TMS320C6416 и тем самым устранить задержку, вызванную переходом между 3 и 0 кольцами защиты центрального процессора компьютера.

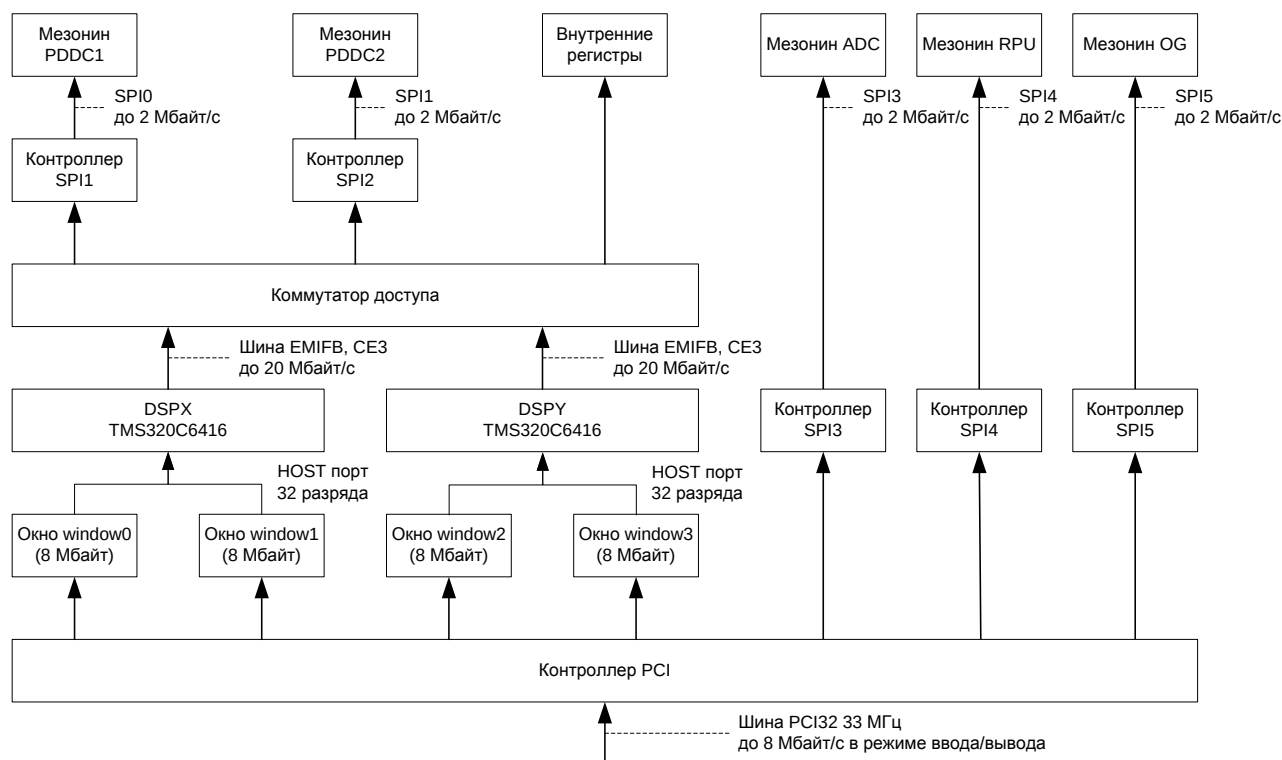


Рисунок 3.1. Структурная схема управления функциональными узлами модуля RDMA

Все мезонины управляются по последовательному синхронному интерфейсу через контроллеры SPI0 .. SPI4, которые идентичны по формату данных и принципу работы. Все контроллеры SPI доступны через шину PCI, контроллеры SPI0 и SPI1 доступны так же и из программы для TMS320C6416, что позволяет, например, подстраивать частоту и фазу DDC в процессе демодуляции.

Доступ к управляющим регистрам мезонинов PDDC1 и PDDC2, а так же к регистрам конфигурации RDMA производится по 16 разрядной параллельной асинхронной шине через интерфейс EMIFB процессоров DSPX или DSPY. Чтобы исключить конфликт при обращении к этим регистрам, используется коммутатор доступа, который позволяет подключить регистры управления PDDC1, PDDC2, канальных FIFO и некоторые другие регистры к процессору DSPX или DSPY.

Внимание. В устройстве, укомплектованном двумя мезонинами PDDC-24 коммутатор настроен таким образом, что DSPX работает только с PDDC1, DSPY работает только с PDDC2.

3.1.2 Схема передачи данных

Упрощенная структурная схема передачи данных между функциональными узлами модуля RDMA приведена на рисунке 3.2. Имейте в виду, что в используемом вами исполнении модуля RDMA могут присутствовать не все показанные на схеме функциональные узлы.

Входные аналоговые сигналы (от 1 до 8) поступают на мезонин ADC, на котором соответственно может быть установлено от 1 до 8 независимых АЦП. Отсчеты от АЦП через полнодоступный коммутатор каналов АЦП поступают на разъемы мезонинов PDDC1, PDDC2 (по 4 канала на каждый) и на входы 4-х FIFO АЦП. Таким образом, коммутатор имеет 8 входных 16-ти разрядных шин и 12 выходных, производительность каждой из которых составляет до 150 MS/s. К каждому каналу PDDC1, PDDC2 и FIFO АЦП может быть подключен любой из 8 АЦП.

В RDMA присутствуют два вида доступных пользователю потоков данных. Первый вид – необработанные отсчеты от АЦП (до 4 потоков). Второй вид – обработанные в мезонинах PDDC данные (до 64 потоков).

Данные с мезонинов PDDC, обработанные установленными на этих мезонинах цифровыми приемниками, поступают на канальные FIFO DDC. Канальные FIFO DDC и FIFO АЦП подключены через коммутатор доступа FIFO к DSPX и DSPY. Каждый из этих процессоров подключен через 64 разрядную параллельную синхронную шину EMIFA с производительностью около 800 Мбайт/с. Коммутатор позволяет подключать отдельные FIFO к DSPX или DSPY.

Внимание. В устройстве, укомплектованном двумя мезонинами PDDC-24 коммутатор настроен таким образом, что DSPX работает только с PDDC1, DSPY работает только с PDDC2.

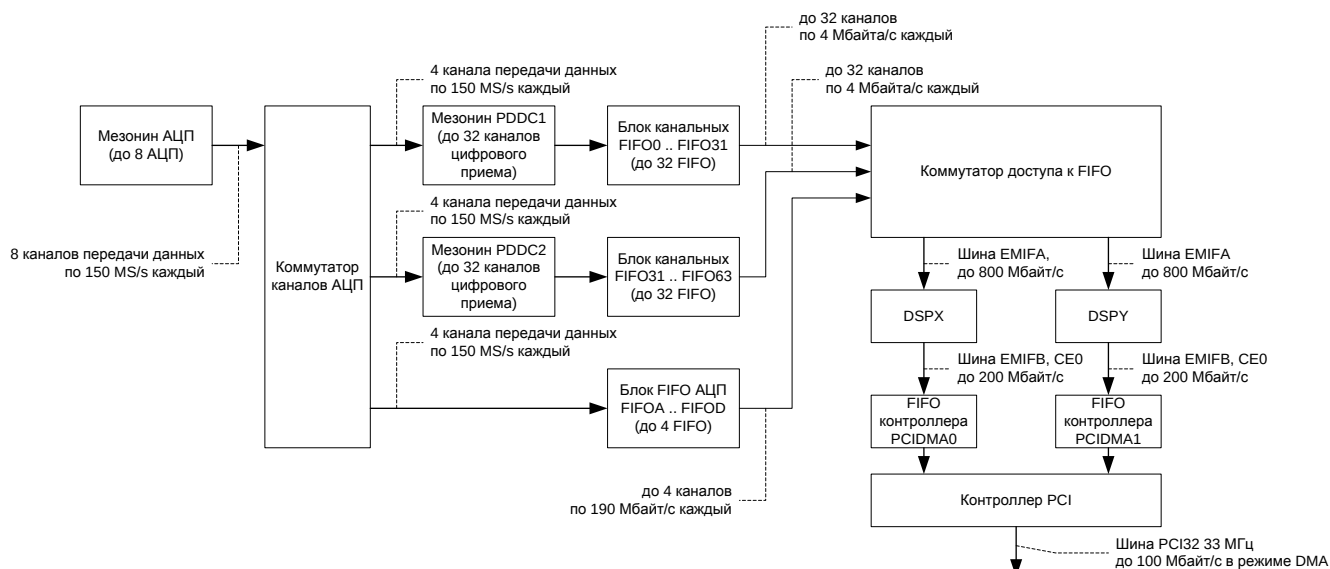


Рисунок 3.2 Структурная схема передачи потоков данных в RDMA

Скоростная передача данных через PCI осуществляется следующим образом. RDMA содержит два независимых контроллера DMA, по одному на каждый TMS320C6416. Процессоры DSPX и DSPY подключены к FIFO контроллера DMA 16 разрядной синхронной шиной с пропускной способностью до 200 Мбайт в секунду. Средняя пропускная способность шины PCI в таком режиме работы составляет около 100 Мбайт в секунду, при условии, что передается всего один поток, а активность других устройств в компьютере минимальна.

3.1.3 Схема обработки прерываний

Использование прерываний не обязательно при применении RDMA, однако, при передаче данных с высокой скоростью и при использовании встроенного в TMS320C6416 контроллера EDMA, без использования прерываний не обойтись.

Структурная схема системы прерываний приведена на рисунке 3.3.

Каждый процессор TMS320C6416 имеет 16 линий ввода/вывода (GPIO0 .. GPIO15), изменение уровня сигнала на которых может вырабатывать прерывание для TMS320C6416 или запускать контроллер EDMA. В устройстве RDMA эти линии используются следующим образом.

Линия GPIO0 и GPIO1 каждого процессора всегда подключены к FIFO соответствующего контроллера PCIDMA. На линии GPIO0 формируется импульс при освобождении FIFO ниже нижней границы. На линии GPIO1 формируется импульс при переполнении FIFO. В нормальных условиях линия GPIO1 всегда должна оставаться в “0”.

Линии GPIO4 .. GPIO15 подключены к выходу коммутатора сигналов прерываний. Ко входу этого коммутатора подключены сигналы наполнения и переполнения от каждого FIFO, на которые поступают потоки входных данных (FIFO0 .. FIFO63, FIFOA .. FIFOD). Коммутатор позволяет подключить к любой линии GPIO4 .. GPIO15 каждого из процессоров DSPX, DSPY любые сигналы от любых FIFO. Если к одной линии подключены несколько сигналов прерывания, они складываются по схеме “ИЛИ”. Такая схема обработки прерываний позволяет реализовать практически любую схему записи входных потоков в память TMS320C6416 с помощью встроенного контроллера EDMA.

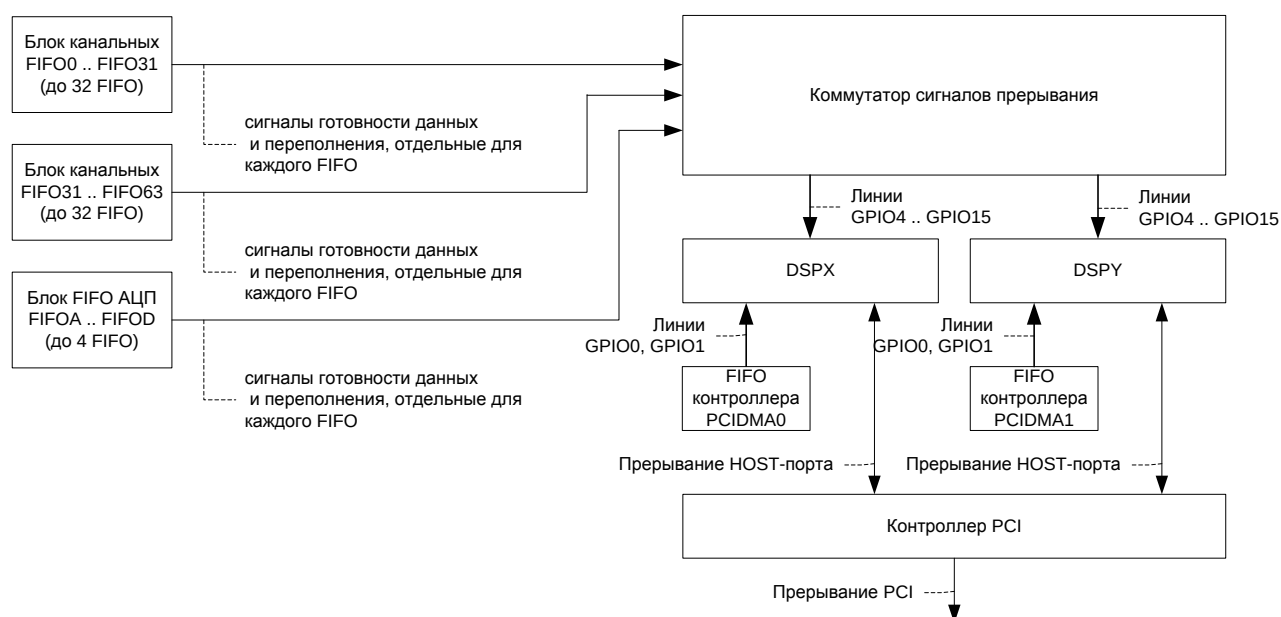


Рисунок 3.3 Структурная схема сигналов прерывания в RDMA

3.2 Устройство FIFO

Все потоки данных, которые, в конечном счете, поступают на процессоры ЦОС, предварительно накапливаются в специальных буферах с организацией типа FIFO. Все FIFO, находящиеся в RDMA имеют одинаковое устройство, показанное на рисунке 3.4.

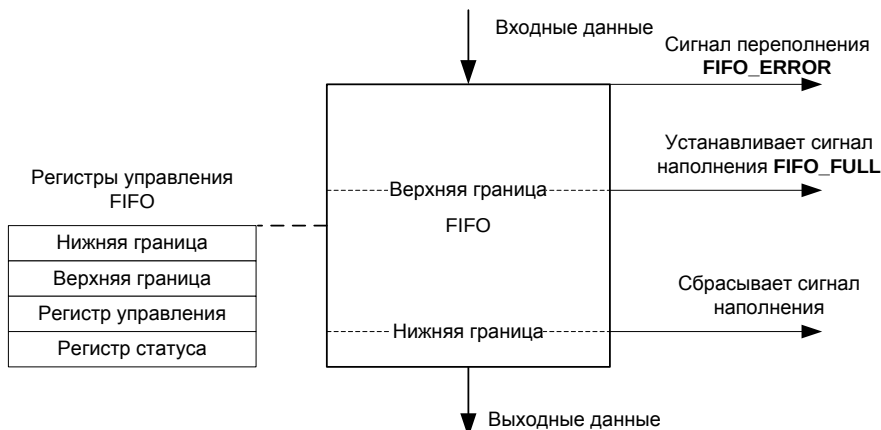


Рисунок 3.4

При поступлении данных текущее положение указателя FIFO перемещается вверх. Когда указатель достигает верхней границы, формируется фронт сигнала наполнения (FIFO_FULL). По этому сигналу должно начаться считывание блока данных из FIFO. Когда указатель достигнет нижней границы, сигнал наполнения FIFO сбросится. Если считывание происходит медленно, или не происходит вообще, FIFO наполнится до максимальной емкости. В этом случае будет сформирован сигнал переполнения (FIFO_ERROR), свидетельствующий о том, что в потоке данных произошел разрыв.

Каждое FIFO имеет определенную максимальную емкость и настройки по умолчанию. Параметры различных FIFO приведены в таблице.

Таблица 3.1

Название	Максимальная емкость	Нижняя граница	Верхняя граница	Примечание
FIFO DDC				
FIFO0	128 x 32	16	64	считывается через EMIFA
.....				
FIFO31	128 x 32	16	64	считывается через EMIFA
FIFO АЦП				
FIFOA	512 x 32	16	256	считывается через EMIFA
FIFO PCIDMA				
FIFODMA0	1024 x 16	512*	64*	Запись через EMIFB
FIFODMA1	1024 x 16	512*	64*	Запись через EMIFB

Примечание * - значение соответствует количеству свободного места в FIFO.

Регистр управления в данной версии не используется. В регистре статуса отдельные разряды соответствуют определенным состояниям FIFO:

- Разряды D11 .. D0 – количество слов в FIFO.
- Разряд D13 – сигнал ошибки, сбрасывается при чтении.
- Разряд D14 – сигнал опустошения до нижней границы.
- Разряд D15 – сигнал наполнения до верхней границы.

Внимание. Реально существующее количество FIFO зависит от исполнения устройства. Например, модуль RDMA, укомплектованный двумя мезонинами PDDC-24, имеет FIFO0 .. FIFO21 для первого мезонина PDDC1 и процессора DSPX и FIFO0 .. FIFO21 для второго мезонина PDDC2 и процессора DSPY.

4 Библиотека управления DSP64162

4.1 Общие сведения

Библиотека DSP64162.dll (далее библиотека управления) предназначена для управления модулем RDMA из программ пользователя, функционирующих в операционной системе семейства WIN32 в режиме пользователя (в 3 кольце защиты процессора). В тесном взаимодействии с драйвером устройства библиотека управления позволяет пользователю использовать все ресурсы RDMA, абстрагируясь от особенностей реализации конкретных узлов RDMA.

Библиотека предоставляет функции для доступа в адресное пространство всех узлов и модулей, входящих в состав RDMA, функции работы с контроллерами PCIDMA и функции поддержки прерываний PCI.

Для сложных операций управления существуют отдельные функции, например, загрузка ПЛИС, инициализация процессора TMS320C6416. Для некоторых операций управления мезонинами так же имеются отдельные функции, например, установка частоты модуля RPU-L. Остальные операции могут быть легко произведены доступом в адресное пространство соответствующего блока.

Вместе с библиотекой поставляется справочный файл DSP64162.hlp, в котором приведена наиболее полная информация о библиотеке.

Все используемые при работе с библиотекой символические константы и описания типов содержатся в файле “DSP64162.h”, находящемся на компакт-диске в каталоге “SUPPORT\INCLUDE”. Для статической линковки на компакт-диске в каталоге “SUPPORT\LIB” имеется файл DSP64162.lib в формате Microsoft Visual Studio 6.0. Если этот формат не поддерживается вашей средой разработки, следует использовать явную загрузку библиотеки.

Библиотеку можно использовать не только на C и C++, но и на других языках программирования, например на Borland Delphi, но в этом случае прототипы функций придется описывать самостоятельно, пользуясь образцами в файле “DSP64162.h”.

4.2 Работа с библиотекой

4.2.1 Подключение библиотеки к проекту

4.2.1.1 Неявное связывание

Неявное связывание DLL – наиболее быстрый способ начать ее использование. Этот способ подходит в том случае, если вы используете линковщик, понимающий файлы библиотек в формате Microsoft Visual Studio 6.0. Недостатком этого способа является то, что в случае, если библиотека будет отсутствовать во время запуска вашей программы, она не будет запущена, а на экране появится стандартное и малоинформативное сообщение.

Чтобы подключить библиотеку в проект, включите заголовочный файл “DSP64162.h” в нужное место с помощью директивы #include. Затем выберите в меню “Project/Settings” и на вкладке “Project/Settings” введите имя файла “DSP64162.lib” в список подключаемых библиотек.

Такой способ загрузки применяется во всех примерах, поставляемых в комплекте программного обеспечения RDMA.

4.2.1.2 Явное связывание

Если ваша среда разработки не поддерживает библиотечные файлы в формате Microsoft Visual Studio 6.0, или необходимо изменить поведение программы в случае отсутствия библиотеки управления, следует применять явное связывание и загрузку DLL с помощью функции LoadLibrary. Недостатком такого способа является необходимость дополнительного описания прототипов всех используемых функций и получение адресов этих функций вручную.

При явном связывании необходимо помнить, что все функции экспортируются под не декорированными именами и используют стандартное соглашение о вызове.

4.2.1.3 Отложенная загрузка DLL

Отложенная загрузка DLL, реализованная в Microsoft Visual Studio 6.0, позволяет использовать DLL как при неявном связывании, но выполнять программы в отсутствие библиотеки [3].

4.2.2 Подключение к устройству

Чтобы начать работать с RDMA, необходимо произвести первоначальное подключение к устройству. Для этого предназначена функция RDSP64162_Init. В качестве параметра этой функции следует передать порядковый номер устройства RDMA, установленного в компьютере. Все функции этой библиотеки в качестве первого параметра принимают этот номер.

Обратите внимание, что если в компьютере установлено несколько RDMA, не существует надежного способа определить, какой физической плате соответствует какой порядковый номер. С помощью функции RDSP64162_GetStatistics можно определить номер слота PCI, куда установлена та или иная плата, однако на многих персональных компьютерах этот номер не соответствует маркировке на печатной плате или на корпусе. Определить соответствие между платой и ее порядковым номером можно так же по заводскому номеру, который возвращает функция RDSP64162_GetStatistics. Этот заводской номер уникален для каждого модуля RDMA.

Проверить наличие подключения к устройству можно в любой момент вызовом функции RDSP64162_IsPresent.

Использование этих функций демонстрируется в примере Sample1, расположенном в каталоге “Samples” на прилагаемом компакт-диске.

4.2.3 Инициализация устройства

Процессоры TMS320C6416 и ПЛИС, входящие в состав RDMA, требуют первоначальной инициализации. Такая инициализация должна выполняться после включения питания до начала любых других операций с модулем. Кроме того, различные мезонины так же могут требовать инициализации после включения питания. Инициализацию всех устройств в модуле можно произвести вызовом функции RDSP64162_InitDevice с параметром RDSP64162_DEVICES.

Сброс процессоров TMS320C6416 выполняется функцией RDSP64162_Reset. После сброса автоматически выполняется настройка всех конфигурационных регистров процессора на значения, рекомендованные для RDMA.

4.2.4 Загрузка программ TMS320C6416

Для загрузки программ в процессоры TMS320C6416 предназначены функции RDSP64162_ProgramLoad, RDSP64162_ProgramLoadFile, RDSP64162_ProgramLoadResource.

Функция RDSP64162_ProgramLoad выполняет операцию копирования блока памяти, начиная с переданного ей адреса и заданной длины в память TMS320C6416.

Функция RDSP64162_ProgramLoadFile загружает программу из файла на диске. Этот файл должен быть в формате, который генерирует программа “hex6x.exe” из комплекта поставки Code Composer Studio фирмы Texas Instruments.

Применение этой функции демонстрируется в примере Sample1, расположенном в каталоге “Samples” на прилагаемом компакт-диске.

Функция RDSP64162_ProgramLoadResource загружает программу из ресурса. Этот ресурс должен иметь пользовательский тип “PROGRAM” и содержать такие же данные, что и файл в предыдущем случае.

В качестве одного из параметров этим функциям передается номер DSP, в который нужно загрузить программу. При одновременном указании флагов RDSP64162_DSP0 и RDSP64162_DSP1 программа загружается сразу в два процессора.

Применение этой функции демонстрируется в примере Sample2.

Внимание. Функции выполняют загрузку программы только в собственную встроенную память процессоров. Загрузка программы в дополнительную оперативную память SDRAM, которая может присутствовать в модуле, с помощью этих функций невозможна.

4.2.5 Загрузка прошивок ПЛИС

Для загрузки прошивок в ПЛИС, входящих в состав RDMA, предназначены функции RDSP64162_CPLDLoad, RDSP64162_CPLDLoadFile, RDSP64162_CPLDLoadResource. Они работают аналогично функциям загрузки программ в TMS320C6416. Формат входных файлов – Altera row binary format.

В использовании этих функций нет необходимости, поскольку загрузка ПЛИС производится автоматически при инициализации устройства.

4.2.6 Доступ в адресное пространство TMS320C6416

Доступ в адресное пространство процессоров TMS320C6416 осуществляется через “окна” в адресном пространстве PCI. Контроллер PCI формирует по 2 независимых “окна” для каждого процессора. Базовый адрес каждого “окна” во внутреннем адресном пространстве TMS320C6416 может независимо изменяться в диапазоне 0x00000000 .. 0xFF800000 с шагом 8 Мбайт.

Чтобы получить доступ в адресное пространство TMS320C6416, следует отобразить “окно” в диапазон виртуальных адресов процесса с помощью функции RDSP64162_MapWindow, произвести необходимые операции, используя в качестве базового адреса полученный от этой функции, отменить отображение с помощью функции RDSP64162_UnmapWindow.

Хотя пользователь может использовать оба окна для каждого процессора, использование окон 1 и 3 не рекомендуется, поскольку они используются при работе функций библиотеки. В этом случае потоковая безопасность как этих функций, так и программы пользователя может быть нарушена.

В адресном пространстве TMS320C6416 выше первых 8 Мбайт нет ресурсов, которые бы требовали постоянного доступа со стороны пользователя через шину PCI. Поэтому рекомендуется использовать окно 0 для DSPX, окно 2 для DSPY, настраивая их на базовый адрес 0x00000000.

Использование этих функций демонстрируется в примере Sample1, расположенном в каталоге “Samples” на прилагаемом компакт-диске.

Распределение адресного пространства TMS320C6416 приведено в Приложении I.

4.2.7 Обработка прерываний

Запросы прерываний для процессора компьютера формируются сигнальными процессорами TMS320C6416 в процессе выполнения ими своих программ, а так же при работе контроллера PCIDMA. Дополнительной поддержки от программы пользователя для корректной обработки этих прерываний не требуется. Однако программа может получать уведомление о приходе прерывания с помощью функции обратного вызова.

Чтобы воспользоваться этой возможностью, вы должны определить функцию типа RDSP64162_FUNCTION_HANDLER и зарегистрировать ее в обработчике прерываний с помощью функции RDSP64162_InstallInterruptHandler. Если происходит прерывание, эта функция будет вызвана с набором соответствующих флагов, которые позволяют установить, что послужило причиной прерывания.

Если получать уведомления о приходе прерывания больше нет необходимости, следует вызвать RDSP64162_InstallInterruptHandler, передав в качестве адреса функции NULL.

Использование этих функций демонстрируется в примере Sample2, расположенном в каталоге “Samples” на прилагаемом компакт-диске.

4.2.8 Генерация прерывания для TMS320C6416

программы, выполняющейся на TMS320C6416, следует вызвать функцию RDSP64162_Interrupt. В качестве одного из параметров передается номер процессора, для которого необходимо вызвать прерывание (RDSP64162_DSP0, RDSP64162_DSP1).

Использование этих функций демонстрируется в примере Sample2, расположенном в каталоге “Samples” на прилагаемом компакт-диске.

4.2.9 Передача данных через PCI с использованием DMA

Использование режима DMA при передаче данных через PCI позволяет не только добиться высокой скорости передачи, но и разгрузить процессор персонального компьютера для выполнения другой работы. Использование этого режима настоятельно рекомендуется, если поток данных, передаваемых от устройства RDMA, превышает 2 Мбайта в секунду. В устройстве RDMA реализована передача данных только от устройства.

Внимание. Для работы этого режима необходима поддержка со стороны пользовательского программного обеспечения, выполняющегося на процессоре TMS320C6416.

Для передачи данных от TMS320C6416 с помощью DMA служит функция RDSP64162_DMARead. Этой функции указывается номер контроллера DMA, который будет выполнять операцию, начальный адрес области памяти для записи данных, длина блока данных и набор флагов. Система функционирует таким образом, что позволяет избежать разрывов в данных между вызовами функции. Для этого функцию RDSP64162_DMARead следует вызывать в первый раз с флагом RDSP64162_MASTER_START, а последующие вызовы – с флагом RDSP64162_MASTER_CONTINUE.

Функция может быть вызвана как синхронно, так и асинхронно. Если возникает необходимость прервать выполнение функции в процессе передачи блока данных, следует вызвать функцию RDSP64162_DMABreak.

Использование этих функций демонстрируется в примере Sample3, расположенном в каталоге “Samples” на прилагаемом компакт-диске.

4.2.10 Получение справочной информации

С помощью функции RDSP64162_GetStatistics можно получать самую разнообразную дополнительную информацию об устройстве. В частности номер PCI слота, заводской номер, номер модификации устройства, наработку, температуру платы. К тому же эта функция позволяет определить состав основных функциональных узлов устройства. В частности тип основной платы, наличие и тип процессоров ЦОС, мезонинов ADC, RPU, RF, PDDC. Список доступных вариантов приведен в Приложении III.

4.2.11 Работа с субмодулем РПУ

Некоторые исполнения модуля RDMA содержат субмодуль РПУ, типа RRL-1 или RRL-2. Для управления субмодулем РПУ служат функции RDSP64162_RPU, которые позволяют устанавливать центральную частоту преобразователя, уровень вносимого затухания аттенюатора, включать и выключать АРУ, определять наличие сигнала захвата синтезатора и уровень выходного сигнала.

Мезонины РПУ могут иметь и дополнительные настройки и регулировки, изменить которые можно с помощью доступа к управляющим регистрам во внутреннем адресном пространстве субмодуля. Распределение адресного пространства мезонинов приведено в Приложении II.

4.2.12 Работа с субмодулем опорного генератора

Большинство исполнений RDMA содержат субмодуль RF, типа FFD-1, представляющий собой встроенный высококачественный перестраиваемый формирователь тактовой частоты для АЦП. Для управления субмодулем RF служат функции RDSP64162_OG, которые позволяют устанавливать тактовую частоту генератора, определять наличие сигнала захвата синтезатора, включать и выключать использование внешней тактовой частоты.

Субмодуль RF может иметь и дополнительные настройки и регулировки, изменить которые можно с помощью доступа к управляющим регистрам во внутреннем адресном пространстве субмодуля RF. Распределение адресного пространства мезонинов приведено в Приложении II.

4.2.13 Работа с мезонином ADC

Для управления мезонином ADC служат функции RDSP64162_ADC, которые позволяют включать и выключать режим энергосбережения АЦП, подключать АЦП к тем или иным каналам передачи данных (рисунок 3.2) и определять степень переполнения АЦП.

Мезонины ADC могут иметь и дополнительные настройки и регулировки, изменить которые можно с помощью доступа к управляющим регистрам во внутреннем адресном пространстве мезонина АЦП. Распределение адресного пространства мезонинов приведено в приложении II.

4.2.14 Работа с цифровыми приемниками

Варианты исполнений устройства RDMA, предназначенные для реализации функций многоканального цифрового радиоприёма, содержат одно или два посадочных места для мезонинов канальных цифровых приёмников (DDC), каждый из которых может содержать до 32 независимых каналов (всего до 64 каналов). Цифровые приемники могут быть разных типов, но все они имеют схожие функции управления.

Функции группы RDSP64162_PDDC позволяют загружать файл конфигурации, сбрасывать и запускать указанные DDC, записывать и читать отдельные регистры управления.

Для загрузки конфигурации в DDC предназначены функции RDSP64162_PDDC_Load, RDSP64162_PDDCLoadFile, RDSP64162_PDDCLoadResource.

Функция RDSP64162_PDDC_Load выполняет загрузку программы из памяти. Загружаемые значения должны располагаться парами 32 разрядных чисел, первое из которых представляет собой адрес во внутреннем адресном пространстве DDC, второе – загружаемые данные.

Функция RDSP64162_PDDCLoadFile загружает программу из файла на диске. Этот файл - bat файл в формате, который генерирует программа HSP50216.exe для DDC ISL5216 фирмы Intersil.

Функция RDSP64162_PDDCLoadResource загружает программу из ресурса. Этот ресурс должен иметь пользовательский тип "PDDC16" и содержать такие же данные, что и файл в предыдущем случае.

4.2.15 Доступ в адресное пространство мезонинов

Если приведенные выше функции не позволяют вам выполнить нужные настройки, вы можете выполнить необходимые действия, записывая значения непосредственно в регистры управления мезонинов. Функции RDSP64162_SerialRead, RDSP64162_SerialReadBlock, RDSP64162_SerialWrite, RDSP64162_SerialWriteBlock предоставляют такую возможность.

Распределение адресного пространства мезонинов приведено в Приложении II.

4.3 Перечень функций

В этом разделе приведен перечень функций библиотеки. Полное описание этих функций приведено в файле справочной системы DSP64162.hlp.

4.3.1 Функции инициализации

Функции инициализации предназначены для подключения к драйверу устройства и определения конфигурации устройства.

4.3.1.1 Функция RDSP64162_Init

Функция RDSP64162_Init предназначена для подключения библиотеки к драйверу устройства серии RDMA. Вызов этой функции должен предварять вызов любой другой функции библиотеки для соответствующего устройства.

```
HRESULT __stdcall RDSP64162_Init(  
    ULONG board  
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

После завершения работы с устройством обязательно вызывайте функцию RDSP64162_Done.

Если вызов функции закончился неудачно, вызовы остальных функций библиотеки для этого устройства так же будут заканчиваться неудачно. Проверить наличие подключения к драйверу можно в любой момент с помощью вызова функции RDSP64162_IsPresent.

4.3.1.2 Функция RDSP64162_Done

Функция RDSP64162_Done предназначена для отключения библиотеки от драйвера устройства серии RDMA. Вызов этой функции должен завершать работу программы пользователя.

```
HRESULT __stdcall RDSP64162_Done(  
    ULONG board  
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае, если подключение не произведено, возвращает S_FALSE.

В случае ошибки функция возвращает E_FAIL.

Примечание

Если подключение не было произведено, функция возвращает S_FALSE. Никакие действия при этом не выполняются.

4.3.1.3 Функция RDSP64162_GetConfig

Функция RDSP64162_GetConfig предназначена для получения информации о номере версии устройства серии RDMA и его текущей конфигурации.

```
HRESULT __stdcall RDSP64162_GetConfig(  
    ULONG board,  
    RDSP64162_CONFIG_STRUCT* pconfig  
);
```

Внимание

Это устаревшая функция, оставленная для совместимости. Вместо нее рекомендуется использовать функцию GetStatistics.

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
pconfig [out] Указатель на структуру RDSP64162_CONFIG_STRUCT, в которой возвращается запрошенная информация.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Перед вызовом функции следует правильно проинициализировать структуру RDSP64162_CONFIG_STRUCT, записав в поле length размер структуры.

4.3.1.4 Функция RDSP64162_IsPresent

Функция RDSP64162_IsPresent в любой момент позволяет узнать, произведено ли подключение к драйверу устройства серии RDMA с соответствующим порядковым номером.

```
HRESULT __stdcall RDSP64162_IsPresent(  
    ULONG board  
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

Возвращаемое значение

В случае наличия подключения функция возвращает S_OK.

В случае отсутствия подключения функция возвращает S_FALSE.

4.3.1.5 Функция RDSP64162_GetStatistics

Функция RDSP64162_GetStatistics предназначена для получения различной справочной информации об устройстве.

```
HRESULT __stdcall RDSP64162_GetStatistics(
    ULONG board,
    ULONG index,
    __int64* pdata
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
index [in] Индекс запрашиваемого параметра.

Название	Описание
DSP64162_STAT_PCISLOT	Номер PCI слота, в который установлено устройство.
DSP64162_STAT_REVISION	Номер ревизии.
DSP64162_STAT_VENDORID	Идентификатор производителя.
DSP64162_STAT_DEVICEID	Идентификатор устройства.
DSP64162_STAT_TIME	Наработка устройства, секунд.
DSP64162_STAT_POWERUP	Количество включений устройства.
DSP64162_STAT_TEMPERATURE1	Температура на плате, °C.
DSP64162_STAT_TEMPERATURE2	Температура кристалла ПЛИС, °C.
DSP64162_STAT_RPM1	Частота вращения вентилятора, RPM.
DSP64162_STAT_TYPEBOARD	Идентификатор основной платы.
DSP64162_STAT_TYPEDSP0	Идентификатор процессора DSPX.
DSP64162_STAT_TYPEDSP1	Идентификатор процессора DSPY.
DSP64162_STAT_TYPEADC	Идентификатор мезонина ADC.
DSP64162_STAT_TYPEPDDC0	Идентификатор мезонина PDDC-1.
DSP64162_STAT_TYPEPDDC1	Идентификатор мезонина PDDC-2.
DSP64162_STAT_TYPEOG	Идентификатор субмодуля RF
DSP64162_STAT_TYPERPU	Идентификатор субмодуля РПУ.
pdata [out]	Указатель на переменную, в которой будет сохранен результат запроса.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

4.3.1.6 Функция RDSP64162_InitDevice

Функция RDSP64162_InitDevice предназначена для инициализации регистров DSP TMS320C6416 и связанной с ним периферии.

```
HRESULT __stdcall RDSP64162_InitDevice(
    ULONG board,
    ULONG flags
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

flags [in] Показывает, какие устройства подлежат инициализации. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_DSP0	Настроить DSPX.
RDSP64162_DSP1	Настроить DSPY.
RDSP64162_CPLD	Настроить загружаемую ПЛИС.
RDSP64162_ADC	Настроить мезонин ADC
RDSP64162_PDDC0	Настроить мезонин PDDC-1
RDSP64162_PDDC1	Настроить мезонин PDDC-2
RDSP64162_RPU	Настроить мезонин РПУ
RDSP64162_OG	Настроить мезонин RF
RDSP64162_DEVICES	Настроить все устройства. Комбинация перечисленных выше флагов.

Перечисленные флаги можно комбинировать.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

После подключения к драйверу обязательно вызывайте функцию RDSP64162_InitDevice с параметром INIT_DEVICES.

Функция RDSP64162_Reset после снятия сброса с процессора вызывает для него функцию RDSP64162_InitDevice.

4.3.2 Функции доступа к DSP через окна

Функции работы с адресным пространством DSP предназначены для обеспечения доступа в адресное пространство DSP TMS320C6416, установленных на плате через окна в виртуальном адресном пространстве пользовательского процесса.

4.3.2.1 Функция RDSP64162_MapWindow

Функция RDSP64162_MapWindow предназначена для установки параметров определенного окна и отображения его в виртуальное адресное пространство вызывающего процесса.

```
HRESULT __stdcall RDSP64162_MapWindow(
    ULONG board,
    ULONG number,
    ULONG address,
    void** pwindow,
    ULONG* windowBase
);
```

Параметры функции

- board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
- number [in] Номер окна. Номер окна начинается с 0.
- address [in] Требуемый базовый адрес окна в адресном пространстве DSP TMS320C6416.
- pWindow [out] Указатель на переменную типа void*, в которой возвращается виртуальный адрес окна в адресном пространстве вызывающего процесса.
- WindowBase [out] Указатель на переменную типа ULONG, в которой возвращается действительный адрес начала окна в адресном пространстве DSP TMS320C6416.

Возвращаемое значение

- В случае успешного выполнения функция возвращает S_OK.
- В случае ошибки функция возвращает E_FAIL.

Примечание

Обращайте внимание на возвращаемое значение в переменной WindowBase. Действительное значение базового адреса окна не изменится, если окно уже отображено с помощью предшествующего вызова RDSP64162_MapWindow.

4.3.2.2 Функция RDSP64162_UnmapWindow

Функция RDSP64162_UnmapWindow отменяет отображение окна в адресное пространство процесса.

```
HRESULT __stdcall RDSP64162_Unmapwindow(
    ULONG board,
    ULONG number
);
```

Параметры функции

- board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
- number [in] Номер окна. Номер окна начинается с 0.

Возвращаемое значение

- В случае успешного выполнения функция возвращает S_OK.
- В случае ошибки функция возвращает E_FAIL.

4.3.2.3 Функция RDSP64162_GetWindows

Функция RDSP64162_GetWindows предназначена для получения виртуальных адресов окон, используемых для доступа в адресное пространство DSP TMS320C6416.

```
HRESULT __stdcall RDSP64162_GetWindows(
    ULONG board,
    void **pwindow0,
    void **pwindow1,
    void **pwindow2,
    void **pwindow3
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

pWindow0 [out] Указатель на переменную типа void*, в которой возвращается виртуальный адрес начала окна Window0.

pWindow1 [out] Указатель на переменную типа void*, в которой возвращается виртуальный адрес начала окна Window1.

pWindow2 [out] Указатель на переменную типа void*, в которой возвращается виртуальный адрес начала окна Window2.

pWindow3 [out] Указатель на переменную типа void*, в которой возвращается виртуальный адрес начала окна Window3.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Обратите внимание, что функция возвращает текущее состояние окон. После вызова функций RDSP64162_MapWindow и RDSP64162_UnmapWindow состояние окон и их виртуальные адреса могут измениться.

4.3.2.4 Функция RDSP64162_GetWindowsBase

Функция RDSP64162_GetWindowsBase предназначена для получения базовых адресов окон, используемых для доступа в адресное пространство DSP TMS320C6416.

```
HRESULT __stdcall RDSP64162_GetWindowsBase(
    ULONG board,
    ULONG* windowBase0,
    ULONG* windowBase1,
    ULONG* windowBase2,
    ULONG* windowBase3
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

WindowBase0 [out] Указатель на переменную типа ULONG, в которой возвращается базовый адрес начала окна Window0.

WindowBase1 [out] Указатель на переменную типа ULONG, в которой возвращается базовый адрес начала окна Window1.

WindowBase2 [out] Указатель на переменную типа ULONG, в которой возвращается базовый адрес начала окна Window2.

WindowBase3 [out] Указатель на переменную типа ULONG, в которой возвращается базовый адрес начала окна Window3.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Обратите внимание, что функция возвращает текущее состояние окон. После вызова функций RDSP64162_MapWindow и RDSP64162_UnmapWindow состояние окон и их базовые адреса могут измениться.

4.3.3 Функции потокобезопасного доступа к DSP

Функции потокобезопасного доступа к DSP позволяют получать доступ в адресное пространство DSP. Скорость доступа с помощью этих функций в общем случае ниже, чем при использовании окон напрямую, однако при этом гарантируется потоковая безопасность на уровне системы.

4.3.3.1 Функция RDSP64162_AddressRead

Функция RDSP64162_AddressRead предназначена для потокобезопасного доступа в адресное пространство сигнальных процессоров DSPX и DSPY.

```
HRESULT __stdcall RDSP64162_AddressRead(
    ULONG board,
    ULONG number,
    ULONG address,
    ULONG* pdata
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 number [in] Номер окна. Номер окна начинается с 0.
 address [in] Адрес в адресном пространстве процессора ЦОС.
 pdata [out] Указатель на переменную, в которой возвращается прочитанное значение.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
 В случае ошибки функция возвращает E_FAIL.

4.3.3.2 Функция RDSP64162_AddressReadBlock

Функция RDSP64162_AddressReadBlock предназначена для потокобезопасного доступа в адресное пространство сигнальных процессоров DSPX и DSPY.

```
HRESULT __stdcall RDSP64162_AddressReadBlock(
    ULONG board,
    ULONG number,
    ULONG address,
    ULONG* pdata,
    ULONG length,
    ULONG flags
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 number [in] Номер окна. Номер окна начинается с 0.
 address [in] Адрес в адресном пространстве процессора ЦОС.
 pdata [out] Указатель на область памяти, в которой возвращаются прочитанные значения.
 length [in] Количество слов, которые необходимо прочитать.
 flags [in] Набор флагов. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_INCREMENT	Автоматическое увеличение адреса после каждого обращения.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
 В случае ошибки функция возвращает E_FAIL.

4.3.3.3 Функция RDSP64162_AddressWrite

Функция RDSP64162_AddressRead предназначена для потокобезопасного доступа в адресное пространство сигнальных процессоров DSPX и DSPY.

```
HRESULT __stdcall RDSP64162_Addresswrite(
    ULONG board,
    ULONG number,
    ULONG address,
    USHORT data
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 number [in] Номер окна. Номер окна начинается с 0.
 address [in] Адрес в адресном пространстве процессора ЦОС.
 data [in] Записываемые данные.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
 В случае ошибки функция возвращает E_FAIL.

4.3.3.4 Функция RDSP64162_AddressWriteBlock

Функция RDSP64162_AddressRead предназначена для потокобезопасного доступа в адресное пространство сигнальных процессоров DSPX и DSPY.

```
HRESULT __stdcall RDSP64162_AddresswriteBlock(
    ULONG board,
    ULONG number,
    ULONG address,
    ULONG* pdata,
    ULONG length,
    ULONG flags
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 number [in] Номер окна. Номер окна начинается с 0.
 address [in] Адрес в адресном пространстве процессора ЦОС.
 pdata [in] Указатель на область памяти, в которой размещены записываемые данные.
 length [in] Количество слов, которые необходимо прочитать.
 flags [in] Набор флагов. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_INCREMENT	Автоматическое увеличение адреса после каждого обращения.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
 В случае ошибки функция возвращает E_FAIL.

4.3.4 Функции управления DSP

Функции управления DSP предназначены для управления DSP и загрузки программы.

4.3.4.1 Функция RDSP64162_Reset

Функция RDSP64162_Reset предназначена для сброса DSP, установленных на плате устройства RDMA.

```
HRESULT __stdcall RDSP64162_Reset(
    ULONG board,
    ULONG flags,
    ULONG reset_state = RESET_PULSE
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 flags [in] Показывает, какие устройства следует сбросить. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_DSP0	Сбросить DSPX.
RDSP64162_DSP1	Сбросить DSPY.

Перечисленные флаги можно комбинировать.

reset_state [in] Состояние линии сброса. Должно записываться одно из значений, приведенных в таблице.

Название	Описание
RESET_SETUP	Установить сигнал сброс в активное состояние.
RESET_RELEASE	Установить сигнал сброс в неактивное состояние.
RESET_PULSE	Сформировать импульс сброса и восстановить работоспособность устройств.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

4.3.4.2 Функция RDSP64162_Interrupt

Функция RDSP64162_Interrupt предназначена для выдачи сигнала прерывания для DSP, установленных на плате устройства серии RDMA.

```
HRESULT __stdcall RDSP64162_Interrupt(
    ULONG board,
    ULONG flags,
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 flags [in] Показывает, каким устройствам следует передать сигнал прерывания. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_DSP0	Передать сигнал прерывания DSPX.
RDSP64162_DSP1	Передать сигнал прерывания DSPY.

Перечисленные флаги можно комбинировать.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

4.3.4.3 Функция RDSP64162_ProgramLoad

Функция RDSP64162_ProgramLoad предназначена для загрузки и запуска программы для DSP TMS320C6416 из готового образа в памяти компьютера.

```
HRESULT __stdcall RDSP64162_ProgramLoad(
    ULONG board,
    ULONG flags,
    void* pdata,
    ULONG length,
    BOOL bStart = TRUE
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

flags [in] Показывает, какие процессоры следует загрузить. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_DSP0	Загрузить программу в DSPX.
RDSP64162_DSP1	Загрузить программу в DSPY.

Перечисленные флаги можно комбинировать.

pdata [in] Указатель на область памяти, в которой содержится битовый образ программы.

length [in] Длина битового образа программы в байтах.

bStart [in] Показывает, нужно ли запускать процессор после загрузки программы.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Данная функция выполняет копирование блока памяти, расположенной начиная с адреса pdata, длиной length байт в адресное пространство процессора TMS320C6416 начиная с адреса 0x00000000. Для копирования используется второе окно соответствующего процессора, которое настраивается на адрес 0x00000000. Если параметр bStart установлен в TRUE, выполняется запуск процессора, в который загружалась программа.

В случае, если программа загружалась в оба процессора, выполняется их синхронный запуск.

4.3.4.4 Функция RDSP64162_ProgramLoadFile

Функция RDSP64162_ProgramLoadFile предназначена для загрузки и запуска программы для DSP TMS320C6416 из файла.

```
HRESULT __stdcall RDSP64162_ProgramLoadFile(
    ULONG board,
    ULONG flags,
    LPCWSTR filename,
    BOOL bStart = TRUE
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

flags [in] Показывает, какие процессоры следует загрузить. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_DSP0	Загрузить программу в DSPX.
RDSP64162_DSP1	Загрузить программу в DSPY.

Перечисленные флаги можно комбинировать.

filename [in] Указатель на строку символов, содержащую имя файла.

bStart [in] Показывает, нужно ли запускать процессор после загрузки программы.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Данная функция выполняет загрузку программы в указанные процессоры из файла на диске. Формат файла должен соответствовать текстовому формату, используемому программой hex6x, поставляемой в комплекте TI Code Composer Studio.

Чтобы получить подходящий файл, используйте команду

```
hex6x <входной файл> -a -memwidth 32 -romwidth 32
```

В результате Вы получите файл с расширением *.a0, который и следует использовать для загрузки.

В остальном действие функции аналогично действию RDSP64162_ProgramLoad.

4.3.4.5 Функция RDSP64162_ProgramLoadResource

Функция RDSP64162_ProgramLoadResource предназначена для загрузки и запуска программы для DSP TMS320C6416 из ресурса.

```
HRESULT __stdcall RDSP64162_ProgramLoadResource(
    ULONG board,
    ULONG flags,
    HMODULE hModule,
    ULONG id,
    BOOL bStart = TRUE
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

flags [in] Показывает, какие процессоры следует загрузить. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_DSP0	Загрузить программу в DSPX.
RDSP64162_DSP1	Загрузить программу в DSPY.

Перечисленные флаги можно комбинировать.

hModule [in] Ссылка модуля, из которого требуется загрузить ресурс.

id [in] Идентификатор ресурса.

bStart [in] Показывает, нужно ли запускать процессор после загрузки программы.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Данная функция загружает указанные процессоры из файла ресурсов.

Указанный ресурс должен иметь пользовательский тип "PROGRAM" и иметь структуру, аналогичную структуре файла, используемого функцией RDSP64162_ProgramLoadFile.

В остальном действие функции аналогично действию RDSP64162_ProgramLoad.

4.3.5 Функции загрузки ПЛИС

Функции загрузки ПЛИС предназначены для загрузки конфигурации ПЛИС, установленных на плате.

4.3.5.1 Функция RDSP64162_CPLDLoad

Функция RDSP64162_CPLDLoad предназначена для загрузки и запуска программы для ПЛИС из готового образа в памяти компьютера.

```
HRESULT __stdcall RDSP64162_CPLDLoad(
    ULONG board,
    ULONG flags,
    void* pdata,
    ULONG length
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 flags [in] Показывает, какие ПЛИС следует загрузить. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_CPLD	Загрузить программу в основную ПЛИС EP1S20.
RDSP64162_ADC	Загрузить программу в ПЛИС мезонина ADC.
RDSP64162_PDDC0	Загрузить программу в ПЛИС мезонина PDDC1.
RDSP64162_PDDC1	Загрузить программу в ПЛИС мезонина PDDC2.

pdata [in] Указатель на область памяти, в которой содержится битовый образ программы.
 length [in] Длина битового образа программы в байтах.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
 В случае ошибки функция возвращает E_FAIL.

Примечание

Данная функция выполняет загрузку блока памяти, расположенной начиная с адреса pdata, длиной length байт в ПЛИС через специальный последовательный порт.

4.3.5.2 Функция RDSP64162_CPLDLoadFile

Функция RDSP64162_CPLDLoadFile предназначена для загрузки конфигурации для ПЛИС из файла.

```
HRESULT __stdcall RDSP64162_CPLDLoadFile(
    ULONG board,
    ULONG flags,
    LPCWSTR filename
);
```

Параметры функции

- board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
- flags [in] Показывает, какие ПЛИС следует загрузить. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_CPLD	Загрузить программу в основную ПЛИС EP1S20.
RDSP64162_ADC	Загрузить программу в ПЛИС мезонина ADC.
RDSP64162_PDDC0	Загрузить программу в ПЛИС мезонина PDDC1.
RDSP64162_PDDC1	Загрузить программу в ПЛИС мезонина PDDC2.

filename [in] Указатель на строку символов, содержащую имя файла.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Данная функция выполняет загрузку конфигурации в указанные ПЛИС из файла на диске. Формат файла должен соответствовать формату RBF, используемому программой Quartus фирмы Altera.

В остальном действие функции аналогично действию RDSP64162_CPLDLoad.

4.3.5.3 Функция RDSP64162_CPLDLoadResource

Функция RDSP64162_CPLDLoadResource предназначена для загрузки конфигурации для ПЛИС из ресурса.

```
HRESULT __stdcall RDSP64162_CPLDLoadResource(
    ULONG board,
    HMODULE hModule,
    ULONG flags,
    ULONG id
);
```

Параметры функции

- board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
- flags [in] Показывает, какие ПЛИС следует загрузить. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_CPLD	Загрузить программу в основную ПЛИС EP1S20.
RDSP64162_ADC	Загрузить программу в ПЛИС мезонина ADC.
RDSP64162_PDDC0	Загрузить программу в ПЛИС мезонина PDDC1.
RDSP64162_PDDC1	Загрузить программу в ПЛИС мезонина PDDC2.

- hModule [in] Ссылка модуля, из которого требуется загрузить ресурс.
- id [in] Идентификатор ресурса.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Данная функция загружает указанные ПЛИС из файла ресурсов.

Указанный ресурс должен иметь пользовательский тип "CPLD" и иметь структуру, аналогичную структуре файла, используемого функцией RDSP64162_CPLDLoadFile.

В остальном действие функции аналогично действию RDSP64162_CPLDLoad.

4.3.6 Функции сигнализации

Функции сигнализации предназначены для регистрации пользовательских обработчиков сигналов прерывания.

4.3.6.1 Функция RDSP64162_InstallInterruptHandler

Функция RDSP64162_InstallInterruptHandler предназначена для подключения обработчика прерываний от устройства серии RDMA.

```
HRESULT __stdcall RDSP64162_InstallInterruptHandler(
    ULONG board
    RDSP64162_FUNCTION_HANDLER* pHandler
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

pHandler [in] Адрес функции типа RDSP64162_FUNCTION_HANDLER, которая будет вызываться при приходе прерывания от устройства серии RDMA. Передайте NULL в качестве параметра, чтобы отключить обработчик прерывания.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

К каждому устройству DSP6416-2 может быть подключен только один обработчик прерывания. Если подключается новый обработчик прерывания, старый обработчик больше не будет вызываться.

4.3.6.2 RDSP64162_FUNCTION_HANDLER

Функция типа RDSP64162_FUNCTION_HANDLER, определяемая пользователем, вызывается при поступлении прерывания от устройства серии RDMA и при поступлении событий от контроллера DMA при передаче блоков данных.

```
typedef HRESULT __stdcall RDSP64162_FUNCTION_HANDLER(
    ULONG board,
    ULONG flags
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

flags [in] Набор флагов. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_INTERRUPT_DSPX	Прерывание от DSPX.
RDSP64162_INTERRUPT_DSPY	Прерывание от DSPY.
RDSP64162_INTERRUPT_DMA0	Прерывание от канала DMA0.
RDSP64162_INTERRUPT_DMA1	Прерывание от канала DMA1.
RDSP64162_INTERRUPT_ERRORDMA0	Прерывание по ошибке от канала DMA0.
RDSP64162_INTERRUPT_ERRORDMA1	Прерывание по ошибке от канала DMA1.
RDSP64162_INTERRUPT_ERRORCOOLER	Не работает вентилятор.

Перечисленные флаги могут быть комбинированы.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

4.3.7 Функции работы с контроллером DMA

Функции работы с контроллером DMA предназначены для передачи информации в режиме DMA.

4.3.7.1 Функция RDSP64162_DMARead

Функция RDSP64162_DMARead предназначена для передачи блока данных от устройства серии RDMA с помощью встроенного в устройство контроллера DMA.

```
HRESULT __stdcall RDSP64162_DMARead(
    ULONG board,
    ULONG number,
    ULONG length,
    ULONG flags,
    void* pBuffer,
    OVERLAPPED* overlapped = NULL
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

number [in] Порядковый номер канала. Номер 0 соответствует DSPX, номер 1 соответствует DSPY.

length [in] Длина в байтах принимаемого блока данных.

flags [in] Набор флагов. Допустимые значения приведены в таблице.

Название	Описание
DSP64162_MASTER_CHANNEL0	Выбрать для передачи канал DMA0.
DSP64162_MASTER_CHANNEL1	Выбрать для передачи канал DMA1.
DSP64162_MASTER_START	В начале передачи блока FIFO будет сброшено.
DSP64162_MASTER_CONTINUE	В начале передачи блока FIFO не будет сброшено. Таким образом, следующий блок будет передан без разрыва с предыдущим.
DSP64162_MASTER_STOP	В конце передачи блока FIFO будет сброшено.
DSP64162_MASTER_TEST	Передача данных от тестового генератора.
Перечисленные флаги можно комбинировать.	
pBuffer [in]	Указатель на область памяти, в которую будут записаны данные.
overlapped [in]	Адрес структуры типа OVERLAPPED, которая должна быть правильно инициализирована пользователем.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Функция RDSP64162_DMARead может вызываться синхронно и асинхронно. В случае асинхронного вызова в качестве параметра overlapped следует передать указатель на структуру типа OVERLAPPED. В этом случае функция возвращает управление немедленно. Пользователь может дожидаться окончания приема блока данных, используя событие overlapped->hEvent. Функция обратного вызова будет вызвана после окончания приема блока.

Если значение параметра overlapped равно NULL, функция будет вызвана синхронно. Управление будет возвращено после окончания приема блока данных.

4.3.7.2 Функция RDSP64162_DMAWrite

Функция RDSP64162_DMAWrite предназначена для передачи блока данных с помощью встроенного в устройство контроллера DMA.

```
HRESULT __stdcall RDSP64162_DMAWrite(
    ULONG board,
    ULONG number,
    ULONG length,
    ULONG flags,
    void* pBuffer,
    OVERLAPPED* overlapped = NULL
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

number [in] Порядковый номер канала DMA устройства серии RDMA. Номер 0 соответствует DSPX, номер 1 соответствует DSPY.

length [in] Длина в байтах передаваемого блока данных.

flags [in] Набор флагов аналогичен предыдущей функции.

pbuffer [in] Указатель на область памяти, из которой будут считаны данные.

overlapped [in] Адрес структуры типа OVERLAPPED, которая должна быть правильно инициализирована пользователем.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

В настоящее время на устройствах серии RDMA функция не поддерживается.

Функция RDSP64162_DMAWrite может вызываться синхронно и асинхронно. В случае асинхронного вызова в качестве параметра overlapped следует передать указатель на структуру типа OVERLAPPED. В этом случае функция возвращает управление немедленно. Пользователь может дождаться окончания передачи блока данных, используя событие overlapped->hEvent. Функция обратного вызова будет вызвана после окончания передачи блока.

Если значение параметра overlapped равно NULL, функция будет вызвана синхронно. Управление будет возвращено после окончания передачи блока данных.

4.3.7.3 Функция RDSP64162_DMAStatus

Функция RDSP64162_DMAStatus предназначена для определения состояния передачи блока данных с помощью встроенного в устройство контроллера DMA.

```
HRESULT __stdcall RDSP64162_DMAStatus(
    ULONG board,
    ULONG number,
    ULONG* length,
    ULONG* flags
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

number [in] Порядковый номер канала DMA.

length [out] Указатель на переменную, в которой возвращается текущее количество переданных байт.

flags [out] Указатель на переменную, в которой возвращается текущее состояние контроллера.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

4.3.7.4 Функция RDSP64162_DMABreak

Функция RDSP64162_DMABreak предназначена для отмены передачи блока данных с помощью встроенного в устройство контроллера DMA.

```
HRESULT __stdcall RDSP64162_DMABreak(
    ULONG board,
    ULONG number
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
number [in] Порядковый номер канала DMA. Номер 0 соответствует DSPX, номер 1 соответствует DSPY.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

4.3.8 Функции работы с мезонином ADC

Функции работы с мезонином ADC предназначены для управления АЦП.

4.3.8.1 Функция RDSP64162_ADCControl

Функция RDSP64162_ADCControl предназначена для управления состоянием АЦП.

```
HRESULT __stdcall RDSP64162_ADCControl(
    ULONG board,
    ULONG number,
    ULONG flags
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
number [in] Порядковый номер АЦП. Порядковый номер начинается с 0.
flags [in] Набор флагов. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_ADC_NORMAL	Нормальный режим работы АЦП.
RDSP64162_ADC_POWERDOWN	Переключить АЦП в режим пониженного энергопотребления.
RDSP64162_ADC_PLLON	Включить встроенный PLL.
RDSP64162_ADC_PLLOFF	Отключить встроенный PLL.
RDSP64162_ADC_TEST1	Включить АЦП в тестовый режим 1.
RDSP64162_ADC_TEST2	Включить АЦП в тестовый режим 2.
RDSP64162_ADC_TEST3	Включить АЦП в тестовый режим 3.

Флаги RDSP64162_ADC_PLLON и RDSP64162_ADC_PLLOFF можно комбинировать с другими.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

Примечание

Некоторые опции работают не на всех видах мезонинов ADC.

4.3.8.2 Функция RDSP64162_ADCSwitch

Функция RDSP64162_ADCSwitch предназначена для коммутации выходов АЦП на разные каналы передачи данных внутри устройства.

```
HRESULT __stdcall RDSP64162_Switch(
    ULONG board,
    ULONG channel,
    ULONG number
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 channel [in] Порядковый номер канала, начинается с 0.
 number [in] Порядковый номер АЦП, начинается с 0.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
 В случае ошибки функция возвращает E_FAIL.

Примечание

Количество АЦП и количество цифровых каналов передачи данных зависит от версии устройства.

4.3.8.3 Функция RDSP64162_ADCGetOverload

Функция RDSP64162_ADCGetOverload предназначена для получения информации о перегрузке входов АЦП. Вызов этой функции должен предварять вызов любой другой функции библиотеки для соответствующего устройства.

```
HRESULT __stdcall RDSP64162_ADCGetOverload(
    ULONG board,
    ULONG number,
    ULONG* pdata
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 number [in] Порядковый номер АЦП. Порядковый номер начинается с 0.
 pdata [out] Указатель на переменную, в которой будет возвращено запрашиваемое значение.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
 В случае ошибки функция возвращает E_FAIL.

Примечание

Функция работает не для всех типов АЦП, которые могут быть установлены на плате. Записываемое по адресу pdata значение показывает количество переполнений на 255 отсчетов входного сигнала для соответствующего АЦП.

4.3.9 Функции работы с submodule РПУ

Функции работы с преобразователем L-диапазона предназначены для управления submodule РПУ.

4.3.9.1 Функция RDSP64162_RPUSetFrequency

Функция RDSP64162_RPUSetFrequency предназначена для установки центральной частоты встроенного преобразователя частоты RPU-L.

```
HRESULT __stdcall RDSP64162_RPUSetFrequency(  
    ULONG board,  
    ULONG channel,  
    double frequency  
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
channel [in] Порядковый номер канала преобразователя. Порядковый номер начинается с 0.
frequency [in] Центральная частота настройки преобразователя частоты, Гц.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

Примечание

Количество каналов преобразователя частоты и диапазон допустимых значений для управляемых параметров зависят от вида устройства.

4.3.9.2 Функция RDSP64162_RPUSetGain

Функция RDSP64162_SetGain предназначена для установки коэффициента передачи встроенного submodule РПУ.

```
HRESULT __stdcall RDSP64162_RPUSetGain(  
    ULONG board,  
    ULONG channel,  
    double gain  
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
channel [in] Порядковый номер канала преобразователя. Порядковый номер начинается с 0.
gain [in] Дополнительное вносимое усиление в тракте преобразователя частоты, дБ.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

Примечание

Количество каналов преобразователя частоты и диапазон допустимых значений для управляемых параметров зависят от вида устройства.

4.3.9.3 Функция RDSP64162_RPUSetGainControl

Функция RDSP64162_Init предназначена для подключения библиотеки к драйверу устройства серии RDMA. Вызов этой функции должен предварять вызов любой другой функции библиотеки для соответствующего устройства.

```
HRESULT __stdcall RDSP64162_RPUSetGainControl(
    ULONG board,
    ULONG channel,
    ULONG flags
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
channel [in] Порядковый номер канала преобразователя. Порядковый номер начинается с 0.
flags [in] Набор флагов, управляющих работой системы АРУ.
DSP64162_RPU_AGCON - система АРУ включена.
DSP64162_RPU_AGCOFF - система АРУ выключена.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

Примечание

Количество каналов преобразователя частоты и диапазон допустимых значений для управляемых параметров зависят от вида устройства.

Система АРУ по умолчанию включена.

4.3.9.4 Функция RDSP64162_RPUGetStatus

Функция RDSP64162_RPUGetStatus предназначена для получения информации о состоянии встроенного субмодуля РПУ.

```
HRESULT __stdcall RDSP64162_RPUGetStatus(
    ULONG board,
    ULONG channel,
    ULONG* pdata
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
channel [in] Порядковый номер канала преобразователя. Порядковый номер начинается с 0.
pdata [in] Указатель на переменную, в которой возвращается запрашиваемое значение. Каждый разряд регистра статуса свидетельствует о наличии ошибки или включении соответствующего режима.
DSP64162_RPU_PLLFAIL - синтезатор неисправен.
DSP64162_RPU_AGC - включена система АРУ.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

Примечание

Количество каналов преобразователя частоты и диапазон допустимых значений для управляемых параметров зависят от вида устройства.

4.3.9.5 Функция RDSP64162_RPUGetSignalLevel

Функция RDSP64162_RPUGetSignalLevel предназначена для запроса среднеквадратического значения мощности сигнала на выходе встроенного субмодуля РПУ.

```
HRESULT __stdcall RDSP64162_RPUGetSignalLevel(
    ULONG board,
    ULONG channel,
    double* pdata
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
channel [in] Порядковый номер канала преобразователя. Порядковый номер начинается с 0.
pdata [out] Указатель на переменную, в которой возвращается запрашиваемое значение выходной мощности в дБмВт.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

Примечание

Количество каналов преобразователя частоты и диапазон допустимых значений для управляемых параметров зависят от вида устройства.

4.3.10 Функции работы с субмодулем RF

Функции работы с субмодулем RF предназначены для управления субмодулем RF – генератором частоты дискретизации.

4.3.10.1 Функция RDSP64162_OGSetFrequency

Функция RDSP64162_OGSetFrequency предназначена для установки частоты дискретизации, генерируемой субмодулем RF.

```
HRESULT __stdcall RDSP64162_OGSetFrequency(
    ULONG board,
    double frequency
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
frequency [in] Частота дискретизации, Гц.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

Примечание

Диапазон допустимых значений для управляемых параметров зависят от вида устройства.

Конструкция модулей RF не позволяет установить любое произвольное значение частоты дискретизации, вместо этого устанавливается допустимое значение, наиболее близкое к заданному. Определить значение частоты дискретизации можно с помощью функции RDSP64162_OGGetFrequency.

4.3.10.2 Функция RDSP64162_OGGetFrequency

Функция RDSP64162_OGGetFrequency возвращает текущее значение частоты дискретизации, генерируемой submodule RF.

```
HRESULT __stdcall RDSP64162_RPUGetFrequency(
    ULONG board,
    double* frequency
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
frequency [out] Указатель на переменную, в которой возвращается текущее значение частоты дискретизации.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

Примечание

Количество каналов преобразователя частоты и диапазон допустимых значений для управляемых параметров зависят от вида устройства.

Конструкция модулей RF не позволяет установить любое произвольное значение частоты дискретизации, вместо этого устанавливается допустимое значение, наиболее близкое к заданному. Поэтому после изменения частоты дискретизации рекомендуется определить ее действительное значение с помощью функции RDSP64162_OGGetFrequency.

4.3.10.3 Функция RDSP64162_OGSetInput

Функция RDSP64162_OGSetInput предназначена для выбора источника частоты дискретизации.

```
HRESULT __stdcall RDSP64162_OGSetInput (
    ULONG board,
    ULONG input
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
input [in] Набор флагов, позволяющий выбрать источник частоты дискретизации.
DSP64162_OG_EXTERNAL – вход внешней частоты.
DSP64162_OG_INTERNAL – submodule RF.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
В случае ошибки функция возвращает E_FAIL.

4.3.10.4 Функция RDSP64162_OGGetInput

Функция RDSP64162_GetInput возвращает номер источника частоты дискретизации.

```
HRESULT __stdcall RDSP64162_OGSetInput (
    ULONG board,
    ULONG* input
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

input [out] Указатель на переменную, в которой возвращается набор флагов, позволяющий установить источник частоты дискретизации.

DSP64162_OG_EXTERNAL – вход внешней частоты.

DSP64162_OG_INTERNAL – submodule RF.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

4.3.10.5 Функция RDSP64162_OGGetStatus

Функция RDSP64162_RPUGetStatus предназначена для получения информации о состоянии встроенного submodule RPU.

```
HRESULT __stdcall RDSP64162_OGGetStatus(
    ULONG board,
    ULONG* pdata
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

pdata [in] Указатель на переменную, в которой возвращается запрашиваемое значение. Каждый разряд регистра статуса свидетельствует о наличии ошибки или включении соответствующего режима.

DSP64162_OG_PLLFAIL - синтезатор неисправен.

DSP64162_OG_EXTERNAL - отсутствует внешняя тактовая частота.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

4.3.11 Функции работы с DDC

Функции работы с DDC предназначены для управления мезонинами PDDC и DDC, установленными на других мезонинах.

4.3.11.1 Функция RDSP64162_PDDCLoad

Функция RDSP64162_PDDCLoad предназначена для загрузки конфигурации DDC из готового образа в памяти компьютера.

```
HRESULT __stdcall RDSP64162_ProgramLoad(
    ULONG board,
    ULONG number,
    PDDC_CONFIG_RECORD* pdata,
    ULONG length
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

number [in] Порядковый номер канала DDC, установленного на плате. Порядковый номер начинается с 0. Если одна микросхема DDC содержит несколько каналов, следует указывать номер наименьшего канала.

pdata [in] Указатель на область памяти, в которой содержится конфигурация.

length [in] Количество записей.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Данная функция выполняет копирование битового образа конфигурации в соответствующие регистры указанного DDC.

Для загрузки используются функции RDSP64162_SerialRead, RDSP64162_SerialWrite.

4.3.11.2 Функция RDSP64162_PDDCLoadFile

Функция RDSP64162_PDDCLoadFile предназначена для загрузки конфигурации DDC из файла.

```
HRESULT __stdcall RDSP64162_PDDCLoadFile(
    ULONG board,
    ULONG number,
    LPCWSTR filename
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

number [in] Порядковый номер DDC, установленного на плате. Порядковый номер начинается с 0. Если одна микросхема DDC содержит несколько каналов, следует указывать номер наименьшего канала.

filename [in] Указатель на строку символов, содержащую имя файла.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Данная функция выполняет загрузку конфигурации в соответствующие регистры указанного DDC из файла на диске.

Формат файла должен соответствовать текстовому формату, используемому фирменными программами настройки, предназначенными для конкретного DDC.

Для загрузки используются функции RDSP64162_SerialRead, RDSP64162_SerialWrite.

4.3.11.3 Функция RDSP64162_PDDCLoadResource

Функция RDSP64162_PDDCLoadResource предназначена для загрузки конфигурации DDC из ресурса.

```
HRESULT __stdcall RDSP64162_ProgramLoadResource(
    ULONG board,
    ULONG number,
    HMODULE hModule,
    ULONG id
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

number [in] Порядковый номер DDC, установленного на плате. Порядковый номер начинается с 0. Если одна микросхема DDC содержит несколько каналов, следует указывать номер наименьшего канала.

hModule [in] Ссылка модуля, из которого требуется загрузить ресурс.

id [in] Идентификатор ресурса.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Данная функция выполняет загрузку конфигурации в соответствующие регистры указанного DDC из файла ресурсов.

Указанный ресурс должен иметь пользовательский тип "PDDC" и иметь структуру, аналогичную структуре файла, используемого функцией RDSP64162_PDDCLoadFile.

Для загрузки используются функции RDSP64162_SerialRead, RDSP64162_SerialWrite.

4.3.11.4 Функция RDSP64162_PDDCReset

Функция RDSP64162_PDDCReset предназначена для сброса DDC, установленных на мезонинах PDDC.

```
HRESULT __stdcall RDSP64162_Reset(
    ULONG board,
    __int64 flags,
    ULONG reset_state = RESET_PULSE
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

flags [in] Каждый разряд соответствует каналу DDC, который требуется сбросить.

reset_state [in] Состояние линии сброса. Должно записываться одно из значений, приведенных в таблице.

Название	Описание
RESET_SETUP	Установить сигнал сброс в активное состояние.
RESET_RELEASE	Установить сигнал сброс в неактивное состояние.
RESET_PULSE	Сформировать импульс сброса и восстановить работоспособность устройств.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

4.3.11.5 Функция RDSP64162_PDDCStart

Функция RDSP64162_PDDCStart предназначена для запуска DDC, установленных на мезонинах PDDC.

```
HRESULT __stdcall RDSP64162_PDDCStart(
    ULONG board,
    __int64 flags
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 flags [in] Каждый разряд соответствует каналу DDC, который требуется запустить.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
 В случае ошибки функция возвращает E_FAIL.

4.3.11.6 Функция RDSP64162_PDDCIsPresent

Функция RDSP64162_PDDCIsPresent предназначена для определения присутствия DDC, установленных на мезонинах PDDC.

```
HRESULT __stdcall RDSP64162_PDDCIsPresent(
    ULONG board,
    ULONG number
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 number [in] Порядковый номер канала DDC, установленного на плате. Порядковый номер начинается с 0.

Возвращаемое значение

В случае наличия соответствующего DDC возвращает S_OK.
 В случае отсутствия соответствующего DDC возвращает S_FALSE.
 В случае ошибки функция возвращает E_FAIL.

4.3.11.7 Функция RDSP64162_PDDCRead

Функция RDSP64162_PDDCRead предназначена для доступа в адресное пространство DDC через синхронные последовательные порты (SPI).

```
HRESULT __stdcall RDSP64162_PDDCRead(
    ULONG board,
    ULONG number,
    ULONG address,
    ULONG* pdata
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 number [in] Порядковый номер канала DDC, установленного на плате. Порядковый номер начинается с 0. Если одна микросхема DDC содержит несколько каналов, следует указывать номер наименьшего канала.
 address [in] Адрес в адресном пространстве DDC.
 pdata [out] Указатель на переменную, в которой возвращается прочитанное значение.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.
 В случае ошибки функция возвращает E_FAIL.

Примечание

Для доступа используются функции RDSP64162_SerialRead, RDSP64162_SerialWrite.

4.3.11.8 Функция RDSP64162_PDDCWrite

Функция RDSP64162_PDDCWrite предназначена для доступа в адресное пространство DDC через синхронные последовательные порты (SPI).

```
HRESULT __stdcall RDSP64162_PDDCWrite(
    ULONG board,
    ULONG number,
    ULONG address,
    ULONG data
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

number [in] Порядковый номер канала DDC, установленного на плате. Порядковый номер начинается с 0. Если одна микросхема DDC содержит несколько каналов, следует указывать номер наименьшего канала.

address [in] Адрес в адресном пространстве DDC.

data [in] Данные для записи.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Для доступа используются функции RDSP64162_SerialRead, RDSP64162_SerialWrite.

4.3.12 Функции доступа к мезонинам

Функции доступа к мезонинам предназначены для доступа в адресное пространство мезонинов.

4.3.12.1 Функция RDSP64162_SerialRead

Функция RDSP64162_SerialRead предназначена для доступа в адресное пространство мезонинов через синхронные последовательные порты (SPI).

```
HRESULT __stdcall RDSP64162_SerialRead(
    ULONG board,
    ULONG address,
    USHORT* pdata
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

address [in] Адрес в адресном пространстве последовательных портов.

pdata [out] Указатель на переменную, в которой возвращается прочитанное значение.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Доступ к последовательным портам мезонинов PDDC-1 и PDDC-2 осуществляется через шину DSPX:EMIFB, поэтому используется окно 1 для доступа во внутреннее адресное пространство DSPX. Базовый адрес окна устанавливается 0x60000000.

4.3.12.2 Функция RDSP64162_SerialReadBlock

Функция RDSP64162_SerialReadBlock предназначена для доступа в адресное пространство мезонинов через синхронные последовательные порты (SPI).

```
HRESULT __stdcall RDSP64162_SerialReadBlock(
    ULONG board,
    ULONG address,
    USHORT* pdata,
    ULONG length,
    ULONG flags
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 address [in] Адрес в адресном пространстве последовательных портов.
 pdata [out] Указатель на область памяти, в которой возвращаются прочитанные значения.
 length [in] Количество слов, которые необходимо прочитать.
 flags [in] Набор флагов. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_INCREMENT	Автоматическое увеличение адреса после каждого обращения.

Перечисленные флаги можно комбинировать.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Доступ к последовательным портам мезонина PDDC-1 и PDDC-2 осуществляется через шину DSPX:EMIFB, поэтому используется окно 1 для доступа во внутреннее адресное пространство DSPY. Базовый адрес окна устанавливается 0x60000000.

4.3.12.3 Функция RDSP64162_SerialWrite

Функция RDSP64162_SerialWrite предназначена для доступа в адресное пространство мезонинов через синхронные последовательные порты (SPI).

```
HRESULT __stdcall RDSP64162_Serialwrite(
    ULONG board,
    ULONG address,
    USHORT data
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.
 address [in] Адрес в адресном пространстве последовательных портов.
 data [in] Записываемые данные.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Доступ к последовательным портам мезонина PDDC-1 и PDDC-2 осуществляется через шину DSPX:EMIFB, поэтому используется окно 1 для доступа во внутреннее адресное пространство DSPY. Базовый адрес окна устанавливается 0x60000000.

4.3.12.4 Функция RDSP64162_SerialWriteBlock

Функция RDSP64162_SerialWriteBlock предназначена для доступа в адресное пространство мезонинов через синхронные последовательные порты (SPI).

```
HRESULT __stdcall RDSP64162_SerialwriteBlock(
    ULONG board,
    ULONG address,
    USHORT* pdata,
    ULONG length,
    ULONG flags
);
```

Параметры функции

board [in] Порядковый номер устройства серии RDMA, установленного в системе. Порядковый номер начинается с 0.

address [in] Адрес в адресном пространстве последовательных портов.

pdata [in] Указатель на область памяти, в которой размещены записываемые данные.

length [in] Количество слов, которые необходимо прочитать.

flags [in] Набор флагов. Допустимые значения приведены в таблице.

Название	Описание
RDSP64162_INCREMENT	Автоматическое увеличение адреса после каждого обращения.

Перечисленные флаги можно комбинировать.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK.

В случае ошибки функция возвращает E_FAIL.

Примечание

Доступ к последовательным портам мезонина PDDC-1 и PDDC-2 осуществляется через шину DSPX:EMIFB, поэтому используется окно 1 для доступа во внутреннее адресное пространство DSPY. Базовый адрес окна устанавливается 0x60000000.

5 Программа монитор DSP6416-2

5.1 Назначение программы

Программа монитор предназначена для демонстрации функциональности и проверки работоспособности устройства RDMA. Эта программа так же может оказать некоторую помощь при отладке пользовательских программ для TMS320C6416, поскольку позволяет выполнять основные операции по управлению установленными в RDMA процессорами TMS320C6416.

В программе используются в основном стандартные элементы управления Windows, которые не должны вызвать затруднений у пользователя, знакомого с этой операционной системой. В случае возникновения каких либо вопросов следует обратиться к встроенной справочной системе, которая выдаст контекстно-зависимую подсказку при нажатии клавиши “F1”.

5.2 Управление модулем RDMA

5.2.1 Общие функции

При выборе в списке, расположенном в главном окне программы слева, пункта “Управление”, пользователь получает возможность управлять параметрами, касающимися всего устройства RDMA. В частности, в поле “Статистика” отображается номер слота PCI, в который установлено устройство, наработка и количество включений, температурный режим. Ниже расположено поле ввода, в котором можно установить тактовую частоту для АЦП. Эта возможность доступна не во всех исполнениях RDMA.

Получить более подробную информацию можно из справочной системы программы монитора.

5.2.2 Управление сигнальными процессорами

При выборе в списке, расположенном в главном окне программы справа, пункта “DSP №1” или “DSP №2”, пользователь получает возможность управлять соответствующим процессором TMS320C6416, установленным в RDMA. В частности, можно производить чтение и запись, сбрасывать процессор, загружать в него программу из файла, вызывать прерывание, контролировать поступление прерываний от TMS320C6416.

5.2.2.1 Доступ в адресное пространство TMS320C6416

Чтение и запись внутренней памяти процессора производится на вкладке “DSP – Чтение/запись”. В верхней части окна в поле “Однократные операции” можно выполнить однократные операции записи или чтения. В поле “Блочные операции” можно прочитать или записать сразу большой участок памяти, или многократно прочитать или записать один адрес. Доступ осуществляется 32 разрядными словами.

5.2.2.2 Управление процессором

Чтобы загрузить программу в TMS320C6416, в поле “Загрузка программы” на вкладке “Управление процессором” нажмите кнопку “Выбрать”. В появившемся диалоговом окне выберите программу, подготовленную в соответствии с разделом 4.2.4, нажмите кнопку “Загрузить”.

Чтобы запустить загруженную таким образом программу, нажмите кнопку “Запустить”. Если программа уже запущена, появляется кнопка “Прерывание”, нажатие на которую вызывает прерывание TMS320C6416.

Нажатие кнопки “Сбросить” вызывает сброс и повторную инициализацию TMS320C6416. После этого программу нужно повторно запустить.

Индикатор “Прерывание от DSP” окрашивается ярким зеленым цветом, когда приходит прерывание от TMS320C6416. Индикатор автоматически гаснет через 200 мс, поэтому если прерывания поступают с меньшим интервалом, индикатор будет светиться постоянно.

Ниже расположены органы управления, позволяющие получить доступ в адресное пространство TMS320C6416, чтобы не переключаться между вкладками.

Получить более подробную информацию можно из справочной системы программы монитора.

5.2.3 Управление DDC

Вкладки “PDDC №1” и “PDDC №2” позволяют получить доступ к функциям управления DDC, установленными на этих мезонинах. Для управления каждым DDC в отдельности имеется несколько вложенных вкладок, по числу установленных на мезонине DDC.

5.2.3.1 Управление всеми DDC

На этой вкладке можно загрузить файл конфигурации во все DDC сразу, сбросить и запустить все каналы DDC, а так же получить доступ в адресное пространство мезонина.

Чтобы загрузить во все DDC одинаковую конфигурацию, в поле “Загрузка программы во все DDC” нажмите кнопку “Выбрать”. В появившемся диалоговом окне выберите файл конфигурации, подготовленный в соответствии с разделом 7, нажмите кнопку “Загрузить”.

Чтобы запустить все каналы DDC, нажмите кнопку “Запустить”. Нажатие кнопки “Сбросить” переводит DDC в состояние сброса. После этого потребуется повторная загрузка файла конфигурации.

5.2.3.2 Индивидуальное управление

Загрузка файла конфигурации, сброс и запуск каждого DDC производится так же, как и всех DDC вместе. Однако расположенные внизу вкладки элементы управления позволяют получить доступ не в адресное пространство мезонина, а к регистрам управления конкретного DDC.

5.2.4 Сигнал с АЦП

На вкладке “Сигнал с АЦП” пользователь может посмотреть спектр и осциллограмму аналоговых сигналов на входах мезонина ADC, установленного в модуле RDMA.

5.2.4.1 Переключение каналов

С помощью кнопок “1” .. “8” можно включать или выключать отображение сигнала с соответствующих входов мезонина АЦП. Переключать каналы так же можно с помощью клавиш “1” .. “8” на клавиатуре.

Индикаторы, расположенные над этими кнопками, отображают переполнение соответствующего АЦП. Если индикатор зеленого цвета – вход не перегружен, если желтого – не более 10% отсчетов превышают максимальное значение, если красного – максимальное значение превышают более 10% отсчетов.

5.2.4.2 Осциллограмма

Чтобы просмотреть осциллограмму сигналов на входах мезонина ADC, нажмите кнопку “Осциллограф” и нажмите кнопку “Запустить”. Переключайте отображение нужных каналов с помощью клавиш “1” .. “8”. Остановите обновление графика с помощью кнопки “Остановить”.

Отсчеты сигнала с каждого АЦП записываются последовательно. На экран одновременно выводится 1000 отсчетов. Графики обновляются каждые 100 мс.

5.2.4.3 Спектрограмма

Чтобы просмотреть спектрограмму сигналов на входах мезонина ADC, нажмите кнопку “Спектр” и нажмите кнопку “Запустить”. Переключайте отображение нужных каналов с помощью клавиш “1” .. “8”. С помощью поля ввода “Усреднение” можно задавать усреднение в диапазоне 1 .. 16, применяемое к спектрограмме.

Остановите обновление графика с помощью кнопки “Остановить”.

Отсчеты сигнала с каждого АЦП записываются последовательно. На экран одновременно выводится 1024 частотных точки. Графики обновляются каждые 100 мс.

5.2.4.4 Масштабирование графика

График, на котором отображаются осциллограммы и спектрограммы имеет возможность масштабирования.

Чтобы изменить масштаб графика по вертикали, нажимайте клавиши “Ctrl - ↑” и “Ctrl - ↓” на клавиатуре. Чтобы изменить масштаб графика по горизонтали, нажимайте клавиши “Ctrl - ←” и “Ctrl - →” на клавиатуре. Этого же можно добиться с помощью “мыши”. Наведите указатель на соответствующую полосу прокрутки и вращайте колесико.

В крупном масштабе на экране виден не весь график. Чтобы переместить отображаемую часть графика по вертикали, нажимайте клавиши “Alt - ↑” и “Alt - ↓” на клавиатуре. Чтобы переместить график по горизонтали, нажимайте клавиши “Alt - ←” и “Alt - →” на клавиатуре. Этого же можно добиться с помощью “мыши”. Наведите указатель на соответствующую полосу прокрутки и нажмите левую кнопку “мыши”. Перемещайте “мышь” в нужную сторону, затем отпустите кнопку.

5.2.4.5 Измерение точных значений

На графике можно включить два маркера, которые перемещаются по точкам графика. Точные значения амплитуды сигнала на осциллограмме или спектре сигнала в этих точках отображаются в верхней части графика. В правом верхнем углу отображается разность между двумя маркерами.

Цвет маркера соответствует цвету графика, с которым он связан в данный момент. С помощью клавиш “F5” для первого маркера и “F6” для второго можно переключать маркеры между графиками или вообще выключать их. Нажмите клавишу “F5” или “F6” несколько раз для получения нужного результата.

5.2.5 Сигнал с DDC

На вкладке “Сигнал с DDC” пользователь может посмотреть спектр и осциллограмму сигналов на выходе цифровых РПУ, установленных на мезонинах PDDC-1 и PDDC-2. Обратите внимание, что для нормальной работы этой функции все цифровые РПУ должны иметь на выходе одинаковую тактовую частоту.

5.2.5.1 Переключение каналов

С помощью кнопок “График 1” .. “График 4” можно выбирать номер цифрового РПУ, сигнал которого будет отображаться на соответствующем графике. Переключать каналы так же можно с помощью клавиш “1” .. “4” на клавиатуре.

5.2.5.2 Осциллограмма

Чтобы просмотреть осциллограмму сигналов на выходах цифровых РПУ, нажмите кнопку “Осциллограф” и нажмите кнопку “Запустить”. Переключайте отображение нужных каналов с помощью клавиш “1” .. “4”. Остановите обновление графика с помощью кнопки “Остановить”.

На экран одновременно выводится 1000 отсчетов. Графики обновляются каждые 100 мс.

5.2.5.3 Спектрограмма

Чтобы просмотреть спектрограмму сигналов на выходах цифровых РПУ, нажмите кнопку “Спектр” и нажмите кнопку “Запустить”. Переключайте отображение нужных каналов с помощью клавиш “1” .. “4”. С помощью поля ввода “Усреднение” можно задавать усреднение в диапазоне 1 .. 16, применяемое к спектрограмме.

Остановите обновление графика с помощью кнопки “Остановить”.

На экран одновременно выводится 1024 частотных точки. Графики обновляются каждые 100 мс.

5.2.5.4 Масштабирование графика

График, на котором отображаются осциллограммы и спектрограммы имеет возможность масштабирования.

Чтобы изменить масштаб графика по вертикали, нажимайте клавиши “Ctrl - ↑” и “Ctrl - ↓” на клавиатуре. Чтобы изменить масштаб графика по горизонтали, нажимайте клавиши “Ctrl - ←” и “Ctrl - →” на клавиатуре. Этого же можно добиться с помощью “мыши”. Наведите указатель на соответствующую полосу прокрутки и вращайте колесико.

В крупном масштабе на экране виден не весь график. Чтобы переместить отображаемую часть графика по вертикали, нажимайте клавиши “Alt - ↑” и “Alt - ↓” на клавиатуре. Чтобы переместить график по горизонтали, нажимайте клавиши “Alt - ←” и “Alt - →” на клавиатуре. Этого же можно добиться с помощью “мыши”. Наведите указатель на соответствующую полосу прокрутки и нажмите левую кнопку “мыши”. Перемещайте “мышь” в нужную сторону, затем отпустите кнопку.

5.2.5.5 Измерение точных значений

На графике можно включить два маркера, которые перемещаются по точкам графика. Точные значения амплитуды сигнала на осциллограмме или спектре сигнала в этих точках отображаются в верхней части графика. В правом верхнем углу отображается разность между двумя маркерами.

Цвет маркера соответствует цвету графика, с которым он связан в данный момент. С помощью клавиш “F5” для первого маркера и “F6” для второго можно переключать маркеры между графиками или вообще выключать их. Нажмите клавишу “F5” или “F6” несколько раз для получения нужного результата.

5.2.6 Управление субмодулем РПУ

На вкладке “Управление РПУ” пользователь может настроить входящий в состав устройства преобразователь частоты на тот или иной участок диапазона входных частот, регулировать коэффициент усиления преобразователя и контролировать его работу.

5.2.6.1 Установка центральной частоты

Чтобы установить центральную частоту настройки преобразователя частоты, введите в поле “Частота настройки” необходимое значение в МГц и нажмите клавишу “Enter”. Допустимое значение центральной частоты настройки изменяется в диапазоне 900 .. 2500 МГц с шагом 1 МГц для субмодулей RRL-1 и RRL-2.

5.2.6.2 Установка усиления

Чтобы отрегулировать усиление преобразователя частоты, введите в поле “Вносимое усиление” необходимое значение в дБ. Усиление, дополнительно вносимое регулируемым аттенуатором, может изменяться в диапазоне -50 до 0 дБ с шагом 0.5 дБ. Контролируйте уровень сигнала на выходе приемника по значению в поле “Уровень выхода”. Номинальный уровень выходного сигнала составляет -20 дБмВт.

5.2.6.3 Включение системы АРУ

Преобразователь снабжен системой АРУ, позволяющей поддерживать уровень сигнала на выходе автоматически. Чтобы включить систему АРУ, установите в поле “Система АРУ” необходимое значение уровня сигнала на выходе преобразователя частоты и поставьте “галочку” в поле “Система АРУ”.

В этом режиме ручная регулировка усиления не доступна. Чтобы отключить систему АРУ, снимите “галочку” в поле “Система АРУ”.

5.2.6.4 Контроль работоспособности

Рядом с полем ввода частоты расположен индикатор, показывающий состояние системы ФАПЧ преобразователя частоты. Если индикатор окрашивается ярким зеленым цветом, преобразователь частоты работает нормально. Красный цвет индикатора свидетельствует об отсутствии опорного сигнала или неисправности преобразователя частоты.

5.3 Проверка функционирования RDMA

5.3.1 Экспресс-проверка

Экспресс-проверка устройства может быть запущена пользователем на вкладке “Проверка устройства”. С помощью этой проверки можно обнаружить неисправности отдельных блоков устройства. При нажатии кнопки “Запустить” начнется последовательное тестирование, индикатор в верхнем левом углу будет показывать ход выполнения тестов, а в нижней части экрана будут появляться сообщения, поясняющие происходящие процессы. После окончания проверки нажмите кнопку “Отчет”, чтобы просмотреть отчет и список возможных ошибок.

Проверка занимает несколько минут, в зависимости от производительности компьютера. Прервать процесс можно в любой момент нажатием кнопки “Остановить”.

5.3.2 Проверка HOST интерфейса TMS320C6416

На этой вкладке проверяется исправность соединений между процессорами TMS320C6416 и шиной PCI, исправность контроллера PCI и самих процессоров.

Тесты “HOST0” и “HOST1” проверяют доступность регистров HOST – порта соответствующих процессоров. Возникновение ошибок в этих тестах свидетельствует о неисправности контроллера PCI либо нарушении соединения на плате.

Тесты “DSP1 окно 0”, “DSP1 окно 1”, “DSP2 окно 2”, “DSP2 окно 3” выполняют проверку внутренней памяти TMS320C6416 путем чтения / записи отдельных адресов в памяти процессоров через “окна”. Возникновение ошибок в этих тестах свидетельствует о неисправности соответствующего процессора.

5.3.3 Проверка встроенной памяти SDRAM

Тесты “DSP1 CS0” и “DSP1 CS1” проверяют исправность SDRAM, подключенной к процессору DSPX, тесты “DSP2 CS0”, “DSP2 CS1” – ко второму. Проверка выполняется записью и последующим считыванием блоков данных в SDRAM из программы, работающей на соответствующем процессоре.

Возникновение ошибок в этих тестах свидетельствует о неисправности микросхем SDRAM либо неисправностях соединений.

5.3.4 Проверка контроллеров PCIDMA

Тест “Проверка ПЛИС” осуществляет многократную загрузку ПЛИС Stratix, расположенной на плате устройства. Возникновение ошибок в этом тесте свидетельствует о неисправности ПЛИС Stratix либо о неисправностях соединений.

Тест “Проверка EEPROM” проверяет установленную на плате микросхему энергонезависимой памяти путем многократного чтения/записи отдельных адресов. Возникновение ошибок в этом тесте свидетельствует о неисправности EEPROM либо о неисправностях соединений.

Тест “Проверка GPIO” проверяет линии GPIO соответствующих процессоров. Возникновение ошибок в этом тесте свидетельствует о неисправностях соединений на плате.

Тесты “Канал DMA0” и “Канал DMA1” позволяют проверить функционирование контроллера PCIDMA. Возникновение ошибок в этом тесте свидетельствует о неисправностях соединений на плате между контроллером PCI и соответствующим процессором.

5.3.5 Проверка интерфейсов EMIFA, EMIFB TMS320C6416

Тесты “EMIFA CS2”, “EMIFA CS3”, “EMIFB CS0”, “EMIFB CS1”, “EMIFB CS2”, “EMIFB CS3” позволяют проверить внешние шины соответствующих процессоров. Возникновение ошибок в этих тестах свидетельствует о неисправностях соединений на плате между ПЛИС Stratix и соответствующим процессором.

5.3.6 Проверка последовательных портов TMS320C6416

Тесты “Порт 1” и “Порт 2” позволяют проверить исправность последовательных портов, расположенных на плате устройства. Для проведения этих тестов необходимо подключить к соответствующему разъему на плате специальную заглушку, входящую в комплект поставки устройства. Возникновение ошибок в этих тестах свидетельствует о неисправностях выходных буферов или соединений на плате.

Тесты “DSP1 – DSP2” и “DSP2 – DSP1” позволяют проверить исправность последовательных портов, которыми соединены DSPX и DSPY между собой. Возникновение ошибок в этих тестах свидетельствует о неисправностях соединений на плате.

5.3.7 Проверка мезонина ADC-8-14

Тест “Последовательный порт” позволяет проверить исправность интерфейса управления мезонином ADC-8-14. Возникновение ошибок в этом тесте свидетельствует о неисправности мезонина либо нарушении контакта в разъеме.

Тесты “Каналы А .. D”, “Каналы Е .. F”, “Каналы G .. H” позволяют проверить соединения на плате в каналах передачи данных с АЦП. Возникновение ошибок в этом тесте свидетельствует о неисправности мезонина либо нарушении контакта в разъеме.

Тест “Выходы АЦП” позволяет проверить исправность выходных линий АЦП. Возникновение ошибок в этом тесте свидетельствует о неисправности мезонина.

5.3.8 Проверка субмодуля РПУ

Тест “Последовательный порт” позволяет проверить исправность интерфейса управления субмодулем РПУ. Возникновение ошибок в этом тесте свидетельствует о неисправности мезонина либо нарушении контакта в разъеме.

Тесты “Канал 1” и “Канал 2” позволяют проверить исправность синтезаторов частоты. В тестах выполняется последовательная настройка на все допустимые частоты и проверяется наличие сигнала захвата ФАП. Возникновение ошибок в этом тесте свидетельствует о неисправности мезонина.

5.3.9 Проверка мезонинов PDDC

Тест “Последовательный порт” позволяет проверить исправность интерфейса управления соответствующим мезонином PDDC. Возникновение ошибок в этом тесте свидетельствует о неисправности мезонина либо нарушении контакта в разъеме.

6 Работа с процессорами TMS320C6416

6.1 Общие сведения

Используемый в устройстве RDMA процессор TMS320C6416¹ фирмы Texas Instruments представляет собой процессор для цифровой обработки сигналов с фиксированной точкой и архитектурой VLIW. На сегодняшний день это один из самых быстродействующих процессоров ЦОС с фиксированной точкой. Фирмой изготовителем этот процессор позиционируется для использования в системах связи, в том числе для применения в сотовых сетях 3G.

В RDMA могут устанавливаться один или два процессора TMS320C6416 или TMS320C6416T с тактовыми частотами 600, 720, 840 и 1000 МГц. Тактовая частота процессора влияет только на производительность вычислений. В остальном эти процессоры идентичны.

Разработку программного обеспечения для TMS320C6416 рекомендуется вести с помощью пакета Code Composer Studio версии 2.0 или старше фирмы Texas Instruments. Все прилагаемые примеры рассчитаны на использование именно этой среды разработки. Настоятельно рекомендуется ознакомиться с документацией [2], [4], [5], [6], а так же с другими указаниями по применению (с учётом исправлений, представленных на сайте компании Texas Instruments в Интернете).

6.2 Примеры программ

6.2.1 Простейшая программа

В каталоге с примерами “Sample1\TMS320C6416” находится простейшая программа, которая, тем не менее, позволяет пояснить основные приемы работы и с библиотекой DSP64162.dll, и с процессорами TMS320C6416.

6.2.1.1 Загрузка и компиляция проекта

Чтобы начать работу с проектом, запустите Code Composer Studio, выберите в меню пункт “Project/Open” и откройте файл “project1.pjt”. Весь текст программы содержится в файле “Sample1.c”. В файле “Sample1.cmd” задано распределение внутренней памяти TMS320C6416 для линковщика. Выберите в меню пункт “Project/Configuration” и в диалоговом окне установите текущую конфигурацию “Release”. Выберите пункт меню “Project/Build”. После окончания процесса компиляции программы в каталоге “Sample1\TMS320C6416\Release” появится готовый файл программы “Sample1.out” в формате COFF.

Обратите внимание, что и в этом, и в последующих примерах, DSP BIOS не используется. Вместо этого используется библиотека “startup.a64”, выполняющая необходимую в программах на языке C инициализацию.

6.2.1.2 Преобразование выходного файла

Чтобы преобразовать этот файл в формат, пригодный для загрузки с помощью библиотеки DSP64162.dll, необходимо запустить на исполнение командный файл “Sample1.bat”. Этот файл запускает на выполнение программу “hexbx.exe”, которая позволяет преобразовать формат COFF в формат Motorola HEX с нужной шириной слова. В нашем случае требуется ширина 32 бита. Полученный файл “Sample1.a0” можно загружать в процессор с помощью программы монитора или из собственной программы с помощью функции RDSP64162_LoadProgramFile.

6.2.1.3 Запуск программы

Рассматриваемая простейшая программа выполняет всего одно действие – она постоянно увеличивает значение 32 разрядной переменной, расположенной во внутренней памяти процессора TMS320C6416 по адресу 0x00010000. Запуск этой программы на исполнение после выполнения компиляции можно выполнить двумя способами.

Запустите находящуюся в каталоге “Sample1\Windows\Release” программу “Sample1.exe”. В числе прочих действий эта программа дважды с определенным интервалом считывает несколько значений, начиная с этого адреса.

Другой способ – использование программы монитора “DSP6416 monitor” в соответствии с разделом 5.2.2 настоящего руководства. Загрузите программу, запустите ее, и считывая периодически адрес 0x00010000, вы увидите, что содержимое этого адреса все время увеличивается.

6.2.2 Использование системы прерываний PCI

В каталоге с примерами “Sample2\TMS320C6416” находится программа, которая позволяет пояснить генерацию и обработку прерываний на шине PCI в процессорах TMS320C6416.

¹ В данном документе рассматривается работа с процессором типа TMS320C6416(T), который имеет встроенный декодер Витерби (VCP) и турбодекодер (TCP). Если потребность в таких функциях отсутствует, то рекомендуется применять модификации RDMA с более дешёвой версией процессора TMS320C6414(T). При этом правила работы с процессором остаются теми же, что и для версии TMS320C6416(T).

6.2.2.1 Загрузка и компиляция проекта

Загрузите и скомпилируйте проект, как описано выше, или воспользуйтесь уже готовой программой “Sample2\TMS320C6416\Release\Sample2.a0”. Обратите внимание, что в этом примере используется TMS320C6000 Chip Support Library [4]. Для этого к проекту подключается библиотека “csl6416.lib”.

6.2.2.2 Проверка работы программы

Загрузите эту программу с помощью монитора в процессор TMS320C6416 и запустите ее.

Вы увидите, что индикатор “Прерывание DSP” вспыхивает примерно один раз в секунду. Нажимая кнопку “Прерывание”, вы можете генерировать прерывание процессора TMS320C6416. Считывайте несколько адресов, начиная с 0x00010000. По адресу 0x00010000 расположен постоянно увеличивающийся счетчик. По адресу 0x00010004 расположен счетчик сформированных TMS320C6416 прерываний на PCI с момента запуска программы. По адресу 0x00010008 расположен счетчик прерываний TMS320C6416, пришедших от персонального компьютера, с момента запуска программы.

6.2.2.3 Внутреннее устройство программы

В файле “Sample2.c” находится весь исходный текст программы. По сравнению с простейшим примером, рассмотренным нами ранее, в программу добавлены обработчик прерывания от PCI, регистрация этого обработчика, формирование прерывания на PCI.

Сформировать прерывание на PCI очень просто. Для этого достаточно сделать два последовательных вызова функции HPI_setHint.

```
HPI_setHint(1); // Clear interrupt signal.  
HPI_setHint(0); // Setup interrupt signal.
```

Обратите внимание, что сначала снимается предыдущее прерывание, а потом устанавливается следующее. Снимать прерывание сразу же нельзя, поскольку в этом случае система обработки прерываний персонального компьютера не успеет среагировать на это прерывание. Лучше всего снять прерывание перед установкой следующего.

Обработчик прерывания от PCI – функция IRQ_DSP.

```
void IRQ_DSP(Uint32 argument, Uint32 event)  
{  
    HPI_setDspint(1);  
    dspcount++;  
}
```

Передаваемые этой функции параметры не используются, однако функция должна иметь именно такой вид для ее регистрации с помощью библиотеки. Первым делом функция должна снять флаг прерывания с помощью HPI_setDspint(1).

Этот обработчик подключается к системе прерываний следующим образом.

```
IRQ_configArgs(IRQ_EVT_DSPINT, IRQ_DSP, 0, IRQ_CCMASK_DEFAULT,  
               IRQ_IEMASK_DEFAULT);  
IRQ_clear(IRQ_EVT_DSPINT);  
IRQ_enable(IRQ_EVT_DSPINT);  
HPI_setDspint(1);  
IRQ_globalEnable();
```

Обратите внимание, что после регистрации обработчика должен быть снят флаг прерывания.

6.2.3 Использование контроллера DMA на шине PCI

В каталоге с примерами “Sample3\TMS320C6416” находится программа, которая позволяет пояснить приемы работы с контроллером PCIDMA. Этот контроллер позволяет передавать данные от TMS320C6416 в ОЗУ персонального компьютера со средней скоростью около 100 Мбайт в секунду. Устройство контроллера приведено в [1], краткое описание – в разделе 3.1.2 и разделе 3.2.

6.2.3.1 Загрузка и компиляция проекта

Загрузите и скомпилируйте проект, как описано выше, или воспользуйтесь уже готовой программой “Sample3\TMS320C6416\Release\Sample3.a0”. Обратите внимание, что в этом примере используется TMS320C6000 Chip Support Library [4]. Для этого к проекту должна быть подключена библиотека “csl6416.lib”.

6.2.3.2 Проверка работы программы

Проверить эту программу с помощью монитора не удастся. Вместо этого запустите программу пример для Windows “Sample3.exe” из каталога “Sample3\Windows\Release”.

Программа “Sample3.exe” загружает процессор TMS320C6416, запускает программу и считывает через DMA 1 Мбайт данных блоками по 64 кбайта. Эти данные записываются в файл “result.bin”. Программа “Sample3.a0” передает данные через контроллер DMA, причем передаваемые данные представляют собой 16 разрядные нарастающие числа. С помощью какой-либо программы просмотра файлов (например, Cool Edit 2000) можно убедиться, что именно эта информация содержится в выходном файле “result.bin”.

6.2.3.3 Внутреннее устройство программы

В файле “Sample3.c” находится весь исходный текст программы. По сравнению с простейшим примером, рассмотренным нами ранее, в программу добавлены обработчик прерывания от линии GPIO0, регистрация этого обработчика и настройка линии GPIO0 на генерацию прерывания.

Обработчик прерывания от линии GPIO0 – функция IRQ_PCIDMA.

```
void IRQ_DSP(Uint32 argument, Uint32 event)
{
    for (i=0; i<PCIDMA_FIFO_LENGTH/2; i++) PCIDMA_FIFO = current++;
}
```

Передаваемые этой функции параметры не используются, однако функция должна иметь именно такой вид для ее регистрации с помощью библиотеки. Эта функция записывает в FIFO контроллера PCIDMA блок нарастающего кода, длина которого равна половине емкости FIFO.

Этот обработчик подключается к линии GPIO0 следующим образом.

```
GPIO_pinEnable(GPIO_PIN0);
GPIO_pinDirection(GPIO_PIN0, GPIO_INPUT);
GPIO_intPolarity(GPIO_PIN0, GPIO_RISING);

IRQ_configArgs(IRQ_EVT_GPINT0, IRQ_PCIDMA, 0,
               IRQ_CCMASK_DEFAULT, IRQ_IEMASK_DEFAULT);
IRQ_clear(IRQ_EVT_GPINT0);
IRQ_enable(IRQ_EVT_GPINT0);
```

Обратите внимание, что линия прерывания GPIO0 должна быть предварительно настроена на использование в качестве источника прерывания.

После запуска программа заполняет FIFO контроллера PCIDMA. Контроллер начинает передавать информацию из этого FIFO по запросу программы “Sample3.exe” и FIFO начинает опустошаться. Когда в FIFO будет свободно 512 16 разрядных слов, будет сформирован сигнал на линии GPIO0. Процедура обработки прерывания получит управление и пополнит FIFO.

Внимание. При получении сигнала прерывания необходимо записать в FIFO не менее 512 16 разрядных слов. Пока такое количество данных не будет записано в FIFO, новый сигнал прерывания не будет выработан.

6.3 Работа с устройствами на плате RDMA

Основными блоками RDMA, доступными из TMS320C6416, являются канальные FIFO (FIFO0 .. FIFO63) и FIFO АЦП (FIFOA .. FIFOD).

Внимание. Реальное количество FIFO зависит от комплектации устройства RDMA. Например, устройство RDMA, укомплектованное двумя мезонинами PDDC-24 содержит FIFO0 .. FIFO22 для мезонина PDDC-1 и процессора DSPX и аналогичные FIFO0 .. FIFO22 для мезонина PDDC-2 и процессора DSPY.

6.3.1 Управление FIFO

6.3.1.1 Размещение в адресном пространстве

Доступ к FIFO АЦП и к FIFO DDC со стороны процессора TMS320C6416 осуществляется по 32 разрядной синхронной шине EMIFA. Каждое FIFO занимает в адресном пространстве TMS320C6416 4 соседних байта. Распределение адресного пространства приведено в разделе Приложение I.

Таким образом, за одно обращение к FIFO TMS320C6416 всегда считывает два отсчета АЦП или DDC, при этом более ранний отсчет находится в разрядах D15 .. D0, более поздний – в разрядах D31 .. D16.

6.3.1.2 Разделение FIFO между процессорами

Для предотвращения конфликтов между процессорами при использовании общих для нескольких FIFO регистров, используется специальный коммутатор, который позволяет подключать FIFO и все связанные с ними ресурсы в адресное пространство одного или другого процессора TMS320C6416.

Этот 16 разрядный регистр DSPCTRL расположен по адресу 0x600000FE и доступен только из DSPY. Разряд D0 в этом регистре управляет группой FIFO32 .. FIFO63, разряд D8 – группой FIFOA, разряд D9 – группой FIFOB, разряд D10 – группой FIFOC, разряд D11 – группой FIFOD. Если соответствующий разряд равен “0”, то группа регистров подключена к DSPX, если равен “1” – к DSPY. По умолчанию все группы подключены к DSPY.

В группу FIFO32 .. FIFO63 входят следующие регистры DSPSPI1, DSPCFST0:D63 .. DSPCFST0:D32, DSPCFST5:D63 .. DSPCFST5:D32, DSPCFSTE:D63 .. DSPCFSTE:D32, регистры управления канальными FIFO FIFO32 .. FIFO63, DSPCFD32 .. DSPCFD63.

В группу FIFOA входят следующие регистры DSPAFS:D0, DSPAFSTS:D0, DSPAFSTS:D8, регистры управления канальным FIFOA, DSPAFDA.

В группу FIFOB входят следующие регистры DSPAFS:D1, DSPAFSTS:D1, DSPAFSTS:D9, регистры управления канальным FIFOB, DSPAFDB.

В группу FIFOC входят следующие регистры DSPAFS:D2, DSPAFSTS:D2, DSPAFSTS:D10, регистры управления канальным FIFOC, DSPAFDC.

В группу FIFOD входят следующие регистры DSPAFS:D3, DSPAFSTS:D3, DSPAFSTS:D11, регистры управления канальным FIFOD, DSPAFDD.

6.3.1.3 Управление FIFO

Каждое FIFO АЦП имеет несколько собственных регистров управления: регистр для записи минимального порогового значения, максимального порогового значения, выбора режима работы и регистр статуса. Общее устройство FIFO описано в разделе 3.2. Кроме этого, FIFOA .. FIFOD имеют общие регистры запуска и статуса.

Чтобы начать работу с FIFO, необходимо остановить FIFO, записав в соответствующий разряд регистра DSPAFS “0”. Затем настроить верхнюю и нижнюю границы FIFO, если это необходимо. Далее следует подключить сигналы FIFO_FULL и FIFO_ERROR к линиям GPIO4 .. GPIO15, записав “1” в соответствующий разряд нужного регистра. После этого необходимо запустить FIFO, записав в соответствующий разряд регистра DSPAFS “1”. После этого, тем или иным способом определяя состояние сигнала FIFO_FULL, следует периодически считывать из него данные, читая соответствующий регистр DSPAFDA .. DSPAFDD.

Управление FIFO DDC отличается тем, что после записи “0” или “1” в соответствующий разряд DSPCFST0 для запуска или остановки FIFO, необходимо произвести запись любого числа в регистр DSPCFST1. Таким способом обеспечивается синхронный запуск всех FIFO0 .. FIFO63 в случае необходимости.

6.3.2 Считывание данных

6.3.2.1 Классификация способов считывания

Способы, которыми TMS320C6416 может считывать данные из FIFO, делятся на две группы. Данные можно считывать либо с помощью непосредственного обращения к регистрам DSPAFDA .. DSPAFDD в адресном пространстве, либо с помощью встроенного в TMS320C6416 контроллера EDMA. Оба этих способа обеспечивают одинаковую скорость считывания данных, однако применение контроллера EDMA позволяет процессору во время считывания информации из FIFO заниматься другой работой, например, обрабатывать считанный ранее блок данных.

При считывании из FIFO необходимо определить момент, когда FIFO наполнится. Это так же может быть сделано двумя способами – по опросу регистров статуса и по прерыванию процессора.

6.3.2.2 Считывание с помощью опроса регистров

Это самый простой способ, не требующий ни применения EDMA, ни обработки прерываний. Возможные алгоритмы считывания приведены на рисунке.

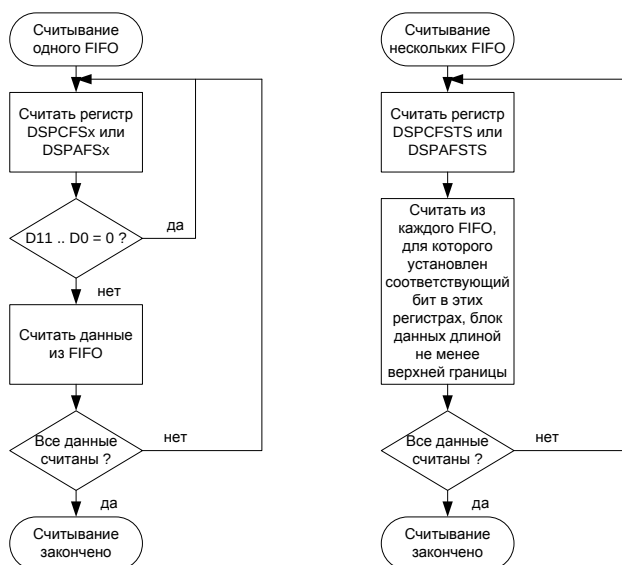


Рисунок 6.1

При считывании данных из одного FIFO достаточно периодически считывать соответствующий регистр DSPCFS для FIFO DDC или DSPAFS для FIFO АЦП. В разрядах D11 .. D0 этого регистра содержится количество слов, находящихся в данный момент в FIFO. Необходимо считать это количество слов из FIFO. Этот процесс следует повторять, пока не будет считан необходимый объем данных.

Считывание из нескольких FIFO можно вести несколько другим способом. Вместо считывания регистров статуса FIFO, можно считывать регистр DSPCFS для FIFO DDC или DSPAFS для FIFO АЦП. В этих регистрах каждый разряд соответствует сигналу наполнения FIFO_FULL соответствующего FIFO. Таким образом, разряды этих регистров установлены в “1”, если соответствующее FIFO заполнено до верхней границы.

6.3.2.3 Считывание с помощью прерывания процессора

Все 12 линий GPIO, подключенных к сигналам FIFO_FULL и FIFO_ERROR через коммутатор (раздел 3.1.3) могут генерировать прерывание для TMS320C6416. Подключив одну или несколько из этих линий с помощью коммутатора к этим сигналам, можно получить уведомление о сигналах наполненности FIFO (FIFO_FULL) или возникшем переполнении.

Вообще, если все FIFO запущены синхронно и данные поступают в них с одной и той же скоростью, достаточно всего двух обработчиков прерывания. Один (ISR_ERROR) будет вызываться при получении сигнала ошибки, другой (ISR_FULL) – при наполнении одного из FIFO. В этом обработчике выполняется считывание количества данных, соответствующих верхней границе FIFO из всех FIFO.

Для реализации такого сценария необходимо следующее.

- К одной линии, например GPIO4, подключить сигнал FIFO_FULL от FIFO0 (DSPGPS4 = 0x00000001).
- Настроить эту линию для генерации прерывания, как показано в примере Sample3 для линии GPIO0.
- Подключить обработчик прерывания ISR_FULL к прерыванию EXT_INT4, как показано в примере Sample3 для обработчика IRQ_PCIDMA.

- К другой линии, например GPIO5, подключить сигнал FIFO_ERROR от всех используемых FIFO (для FIFO0 .. FIFO15, например, DSPGPS5 = 0x0000FFFF).
- Настроить эту линию для генерации прерывания, как показано в примере Sample3 для линии GPIO0.
- Подключить обработчик прерывания ISR_ERROR к прерыванию EXT_INT5, как показано в примере Sample3 для обработчика IRQ_PCIDMA.

Внимание. Чтобы обеспечить минимальную вероятность возникновения переполнения, следует установить верхнюю границу FIFO на половину максимальной емкости. Такое значение установлено по умолчанию.

6.3.2.4 Считывание с помощью EDMA

Использование контроллера EDMA похоже на использование прерываний, но есть и некоторые отличия. Сигнал FIFO_FULL через один из выводов GPIO4 .. GPIO15 поступает непосредственно на контроллер EDMA, который настроен таким образом, что при поступлении этого сигнала считывает определенное количество данных из FIFO. Когда все данные будут считаны, контроллер EDMA генерирует прерывание EDMA_INT. Процессор, получив это прерывание, может приступить к обработке блока данных.

Контроллер EDMA отличается большой гибкостью и может выполнять многие операции автоматически. На рисунке приведена возможная схема работы EDMA контроллера для считывания данных из 4 FIFO, запущенных синхронно и имеющих одинаковую скорость.

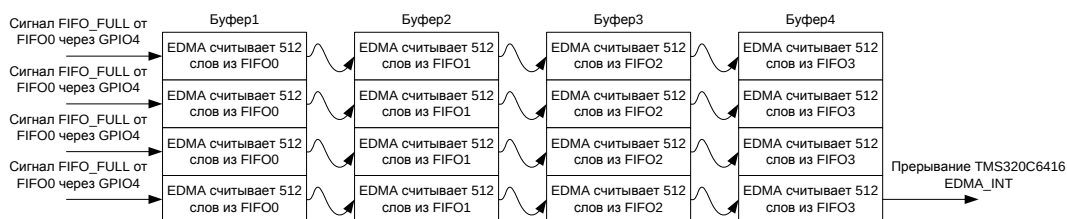


Рисунок 6.2

Таким образом, когда TMS320C6416 получит прерывание EDMA_INT, в его распоряжении окажется 4 готовых к обработке блока данных по 2048 слов в каждом. Настроив в процедуре обработки прерывания EDMA_INT контроллер EDMA на заполнение следующих 4 буферов, процессор может приступить к выполнению других действий.

Указания по настройке контроллера EDMA приведены в [2] в разделе 6.

6.3.3 Доступ в адресное пространство мезонинов PDDC

Хотя доступ в адресное пространство мезонинов и некоторые функции управления реализованы в библиотеке DSP64162.dll, в некоторых случаях может потребоваться доступ к DDC и из программы TMS320C6416. В адресном пространстве TMS320C6416 имеются два регистра. Регистр DSPSPI0 используется для доступа к мезонину PDDC-1, регистр DSPSPI1 – к мезонину PDDC-2.

Доступ на запись осуществляется следующим образом.

- Убедитесь, что разряд D31 регистра DSPSPI установлен в “1”. Это свидетельствует о готовности контроллера SPI принять данные.
- Запишите в регистр DSPSPI 32 разрядное слово, в разрядах D15 .. D0 которого содержатся данные для передачи в мезонин PDDC, в разрядах D29 .. D16 – адрес в адресном пространстве мезонина, в разряде D30 – “0”.
- Запись будет завершена, когда разряд D31 регистра DSPSPI снова будет установлен в “1”.

Доступ на чтение осуществляется следующим образом.

- Убедитесь, что разряд D31 регистра DSPSPI установлен в “1”. Это свидетельствует о готовности контроллера SPI принять данные.
- Запишите в регистр DSPSPI 32 разрядное слово, в разрядах D29 .. D16 которого содержится адрес в адресном пространстве мезонина, в разряде D30 – “1”.
- Дождитесь, когда разряд D31 регистра DSPSPI снова будет установлен в “1” – считанные по SPI данные будут в разрядах D15 .. D0.

Распределение адресного пространства мезонинов приведено в приложении Приложение II.

6.3.4 Использование последовательных портов

Последовательные порты MBSP0 процессоров DSPX и DSPY выведены на специальные разъемы “SERIAL_PORT X” и “SERIAL_PORT Y” соответственно. Последовательные порты MBSP1 соединяют процессоры между собой на плате устройства.

Конструкция устройства не накладывает никаких ограничений на использование всех возможностей этих портов, заложенных в TMS320C6416(T). Пользователь может использовать любой из вариантов работы портов, описанных в [2] в разделе 12. Следует только следить за тем, чтобы соединенные друг с другом порты были настроены согласованно – использовали одну и ту же скорость и формат данных, а так же способ формирования тактовой частоты. Например, если передатчик порта MBSP1 процессора DSPX настроен на формирование тактовой частоты, приемник порта MBSP1 процессора DSPY должен быть настроен на использование внешней тактовой частоты.

Внимание. При подключении к разъемам “SERIAL_PORT X” и “SERIAL_PORT Y” каких либо устройств, в том числе и других RDMA, строго следуйте рекомендациям, приведенным в [1]. Неправильное соединение способно вывести из строя RDMA или подключенные к нему устройства.

6.3.5 Использование внутрисхемных эмуляторов

На плате устройства RDMA, предусмотрен разъем “EMULATOR” для подключения внутрисхемного JTAG-эмулятора процессоров TMS320C6416(T), например, XDS560 фирмы Texas Instruments.

При подключении эмулятора и работе с ним следует руководствоваться документацией на используемый эмулятор и среду разработки. При настройке эмулятора имейте в виду, что процессоры DSPX и DSPY соединены цепочкой.

При невозможности подключения разъема используемого вами эмулятора непосредственно в разъем “EMULATOR”, используйте переходник – удлинитель, входящий в комплект поставки RDMA.

7 Работа с DDC

7.1 Общие сведения

Микросхемы DDC представляют собой специализированные процессоры цифровой обработки сигналов, ориентированные на выполнение функций, свойственных многоканальным радиоприёмным устройствам, таких как: селекция сигнала из группового спектра, квадратурный перенос спектра сигнала на ПЧ 0 МГц, цифровая фильтрация, децимация, автоматическая регулировка усиления, ресемплирование.

На рынке представлено достаточно большое количество микросхем DDC от разных фирм-производителей. Конструкция RDMA позволяет применение практически любых из этих микросхем, при выполнении на их основе мезонинов многоканальных цифровых РПУ.

Ниже даны рекомендации по работе с двумя разновидностями микросхем DDC, лежащими в основе двух типов мезонинов:

- Модуль PDDC – 24 содержащий до шести микросхем ISL5216, предназначенных для одновременной обработки четырёх узкополосных сигналов (с полосами от десятков герц до 1 МГц) каждая.
- Модуль ADC-2-12-26 – мезонин, на котором совмещены функции АЦП и DDC. В качестве микросхемы DDC в нём применена AD6636, предназначенная для одновременного приёма и обработки до шести широкополосных сигналов (с полосой от нескольких килогерц до 30 МГц).

7.2 Рекомендации по настройке DDC ISL5216

7.2.1 Общие сведения

Всего на мезонинах, установленных в устройстве RDMA, может содержаться до 12 ISL5216, что обеспечивает 48 каналов цифрового преобразования частоты и фильтрации. Микросхемы DDC ISL5216 имеют множество возможностей по настройке, однако в составе RDMA на них налагается ряд ограничений, в основном связанных с настройкой ввода и вывода данных. Далее описаны эти ограничения и рекомендации по настройке ISL5216. Более подробная информация приведена в [7].

7.2.2 Работа с программой конфигурации

Фирмой Intersil свободно распространяется программное обеспечение для лабораторной платы собственного производства “HSP50216 / ISL5216 EVAL Software”. Это программное обеспечение позволяет настраивать практически все режимы работы ISL5216. Программа позволяет формировать файл конфигурации ISL5216, который можно загружать в устройство с помощью функции RDSP64162_PDDC_LoadFile библиотеки “DSP64162.dll”.

Программное обеспечение “HSP50216 / ISL5216 EVAL Software” устанавливается в подкаталог DDC_ISL5216. Прилагаемое к RDMA ПО имеет версию 1.21. Более новая версия может быть получена через Интернет с сайта компании Intersil.

Внимание. Устанавливать имеющиеся в комплекте ПО “HSP50216 / ISL5216 EVAL Software” драйверы нет необходимости. Они предназначены только для работы с лабораторной платой. Функции, доступ к которым осуществляется через пункты 11, 12, 13, 15, 16, 17 главного меню, не работают без лабораторной платы фирмы Intersil.

Чтобы начать работу с программой, запустите исполняемый файл “ISL5216.exe” из каталога DDC_ISL5216 и следуйте указаниям [8]. В последующих разделах приведены ограничения и рекомендации, которых следует придерживаться при настройке ISL5216.

После окончания настройки выберите пункты меню (9), (10) и (14), чтобы сформировать файл, пригодный для загрузки с помощью функции RDSP64162_PDDC_LoadFile.

7.2.3 Ограничения

Входы ISL5216 подключены к каналам передачи данных с мезонина ADC как указано в таблице 7.1.

Таблица 7.1

Входы ISL5216	Мезонин PDDC1	Мезонин PDDC2
INA[15:0]	Канал А	Канал Е
INB[15:0]	Канал В	Канал F
INC[15:0]	Канал С	Канал G
IND[15:0]	Канал D	Канал H

В пункте (1) главного меню необходимо установить следующие значения.

```
(2) Input Clock Rate..... тактовая частота АЦП в Гц
(4) Serial Clock Rate..... CLK/4
(5) Serial Clock Polarity... L->H @ Midbit
(8) Target device: ISL5216
```

В пункте (2) главного меню необходимо установить следующие значения.

```
(2) Input Format..... Twos Complement
(3) ..... Fxd Point
(4) Demux Delay..... 0 clocks
```

В пункте (1) меню настройки каналов необходимо установить следующие значения.

```
(1) Sample Rate.... тактовая частота АЦП в Гц
(2) Input Source... AIN, BIN, CIN, DIN
(3) Input Format... Twos Comp.,real
(4) ... Fxd Point
(7) Demux Delay.... 0
```

В пункте (3) меню настройки каналов необходимо установить следующие значения.

```
(1) Sync Active..... High
(2) Sync Position... Early
(3) Output Delay.... 0
Output 1:
Slot Type Bits Sync
( 8) 1 любое 16 fixed Y
( 9) 2 zeros 0 fixed N
(10) 3 zeros 0 fixed N
(11) 4 zeros 0 fixed N
(12) 5 zeros 0 fixed N
(13) 6 zeros 0 fixed N
(14) 7 zeros 0 fixed N
Output 2:
Slot Type Bits
(15) 1 любое 16 fixed
(16) 2 zeros 0 fixed
(17) 3 zeros 0 fixed
(18) 4 zeros 0 fixed
(19) 5 zeros 0 fixed
(20) 6 zeros 0 fixed
(21) 7 zeros 0 fixed
```

Выходы Output1 и Output2 к выводам ISL5216 для разных каналов следует подключать в соответствии с таблицей.

Таблица 7.2

	Channel 0	Channel 1	Channel 2	Channel 3
Output1	SD1A	SD1B	SD1C	SD1D
Output2	SD2A	SD2B	SD2C	SD2D

В качестве выходных данных можно выбирать любые параметры, например амплитуду и фазу, главное, чтобы первый параметр передавался через выход Output1, второй – через выход Output2. Все параметры должны иметь длину 16 битов.

Остальные управляющие регистры ISL5216 могут устанавливаться свободно, в соответствии с необходимостью.

В каталоге Examples\Sample5\ISL5216 приведен пример файла конфигурации, который учитывает описанные выше ограничения и обеспечивает фильтрацию в полосе 50 кГц при частоте дискретизации 80 МГц с помощью стандартных фильтров HBF1 ISL5216.

7.2.4 Доступ к данным

Выходные данные из ISL5216 доступны через каналные FIFO0 .. FIFO31 (PDDC1) или FIFO32 .. FIFO63 (PDDC2). Сформированные на выходе ISL5216 данные записываются последовательно. Значение, передаваемое через выход Output1 соответствующего канала ISL5216 записывается первым. За ним записывается значение, переданное через выход Output2.

7.3 Рекомендации по настройке DDC AD6636

7.3.1 Общие сведения

Использование DDC AD6636 в составе RDMA налагает на их конфигурацию ряд ограничений, в основном связанных с настройкой ввода и вывода данных. Далее описаны эти ограничения и рекомендации по настройке AD6636. Более подробная информация приведена в [9].

7.3.2 Работа с программой конфигурации

Фирмой Analog Devices свободно распространяется программное обеспечение для DDC “SoftCell™ Filter Design”. Это программное обеспечение позволяет рассчитать коэффициенты фильтров MRCF и DRCF, а так же оценить характеристики получившегося тракта цифровой фильтрации при разном распределении коэффициентов децимации по разным фильтрам. Рассчитанные характеристики можно сохранить в файл. К сожалению, этот файл не содержит всей необходимой информации для настройки AD6636. Чтобы получить файл конфигурации, пригодный для загрузки с помощью функции RDSP64162_PDDC_LoadFile библиотеки “DSP64162.dll”, воспользуйтесь программой AD6636_Convertor.

Программное обеспечение “SoftCell™ Filter Design” устанавливается в подкаталог DDC_AD6636. Прилагаемое к RDMA ПО имеет версию 2.5. Более новая версия может быть получена через Интернет с сайта компании Analog Devices.

Чтобы начать работу с программой, запустите исполняемый файл “FilterDesign.exe” из каталога DDC_AD6636 и следуйте указаниям встроенной справочной системы. После окончания настройки выберите пункт меню File/Save Filter, чтобы записать файл.

Запустите программу “AD6636.exe” из каталога DDC_AD6636. На экране появится основное окно программы, показанное на рисунке. Для каждого из каналов AD6636 установите номер входа, центральную частоту настройки и загрузите полученный ранее файл конфигурации с расширением “f36”. После этого нажмите кнопку “Сохранить”.

Полученный таким образом конфигурационный файл можно загружать функцией RDSP64162_PDDC_LoadFile. В каталоге примеров Sample6 находится файл 80000_50.bat, который был получен таким образом.

7.3.3 Ограничения

7.3.3.1 Жестко заданные регистры

Некоторые регистры конфигурации в адресном пространстве AD6636 должны записываться жестко заданными значениями. Эти регистры перечислены в таблице 7.3.

Таблица 7.3

Адрес	Название, согласно [9]	Значение	Примечание
0x01	Chip I/O access control	0x00	
0x04	Pin synchronisation configuration	0x5F	
0x06	LVDS control	0x06FC	
0x14	Port CD IQ correction control	0x0000	
0x16	Port AB IQ correction control	0x0000	
0xBC	Parallel port output control	0x00B08C83	
0xC0	Output port control	0x018F	

7.3.3.2 Регистр ADC Input Control

Этот регистр управляет подключением входов каналов Channel 0 .. 5 к различным входам AD6636. Особенность конструкции мезонина ADC-2 ограничивает возможные значения разрядов <23:20>, <19:16>, <15:12>, <11:8>, <7:4>, <3:0>. Допустимые значения этих разрядов приведены в таблице 7.4.

Таблица 7.4

Значение разрядов	Источник входных данных
0	Вход А мезонина ADC-2-12
1	Вход В мезонина ADC-2-12
4	Генератор случайных чисел

7.3.3.3 Регистр ADC CLK Control

Разряды <11:8> этого регистра всегда должны быть установлены в “0”.

Остальные управляющие регистры AD6636 могут устанавливаться свободно, в соответствии с необходимостью.

В каталоге Examples\Sample6\AD6636 приведен пример файла конфигурации, который учитывает описанные выше ограничения и обеспечивает фильтрацию в полосе 50 кГц.

7.3.4 Доступ к данным

Выходные данные из AD6636 доступны через FIFO0 .. FIFO5. Первым в FIFO записывается значение I, за ним записывается значение Q.

Обратите внимание, что данные с AD6636 подаются так же на каналы передачи данных A .. H, как и данные от АЦП (раздел 3.1.2).

Устройства серии RDMA. Руководство по программированию.

Модули ЦОС серии RDMA.

8 Дополнительная информация

Самая свежая информация и изменения данного руководства, относительно программных компонентов, поставляемых вместе с устройством RDMA, если таковые имеются, приведены в файле "readme.txt" в корневом каталоге на прилагаемом к устройству компакт-диске.

Приложение I. Адресное пространство DSP

В этом разделе приведен перечень управляющих регистров. Подробное описание управляющих регистров приведено в справочном файле DSP64162.hlp.

Таблица III.1

Адрес	Длина	Обозначение	Краткое описание
0x00000000	1М		Внутренняя память или кеш L2
0x01800000	1000М		Регистры конфигурации процессора TMS320C6416
0x60000000	4	DSPSPI0	Регистр управления мезонином PDDC по SPI
0x60000010	8	DSPCFST0	Регистр запуска канальных FIFO0 .. 31
0x60000018	2	DSPCFST1	Регистр изменения конфигурации FIFO0 .. 31
0x60000020	4	DSPCFSTS	Регистр признака наполненности FIFO0 .. 31
0x60000028	4	DSPCFSTE	Регистр признака переполнения FIFO0 .. 31
0x60000030	2	DSPAFS	Регистр запуска FIFOA .. D
0x60000032	2	DSPAFSTS	Регистр статуса FIFOA .. D
0x60000180	16	DSPGPS4	Регистр управления линией GPIO4 по наполнению FIFO
0x60000190	16	DSPGPE4	Регистр управления линией GPIO4 по переполнению FIFO
0x600001A0	16	DSPGPS5	Регистр управления линией GPIO5 по наполнению FIFO
0x600001B0	16	DSPGPE5	Регистр управления линией GPIO5 по переполнению FIFO
0x600001C0	16	DSPGPS6	Регистр управления линией GPIO6 по наполнению FIFO
0x600001D0	16	DSPGPE6	Регистр управления линией GPIO6 по переполнению FIFO
0x600001E0	16	DSPGPS7	Регистр управления линией GPIO7 по наполнению FIFO
0x600001F0	16	DSPGPE7	Регистр управления линией GPIO7 по переполнению FIFO
0x60000200	16	DSPGPS8	Регистр управления линией GPIO8 по наполнению FIFO
0x60000210	16	DSPGPE8	Регистр управления линией GPIO8 по переполнению FIFO
0x60000220	16	DSPGPS9	Регистр управления линией GPIO9 по наполнению FIFO
0x60000230	16	DSPGPE9	Регистр управления линией GPIO9 по переполнению FIFO
0x60000240	16	DSPGPS10	Регистр управления линией GPIO10 по наполнению FIFO
0x60000250	16	DSPGPE10	Регистр управления линией GPIO10 по переполнению FIFO
0x60000260	16	DSPGPS11	Регистр управления линией GPIO11 по наполнению FIFO
0x60000270	16	DSPGPE11	Регистр управления линией GPIO11 по переполнению FIFO
0x60000280	16	DSPGPS12	Регистр управления линией GPIO12 по наполнению FIFO
0x60000290	16	DSPGPE12	Регистр управления линией GPIO12 по переполнению FIFO
0x600002A0	16	DSPGPS13	Регистр управления линией GPIO13 по наполнению FIFO
0x600002B0	16	DSPGPE13	Регистр управления линией GPIO13 по переполнению FIFO
0x600002C0	16	DSPGPS14	Регистр управления линией GPIO14 по наполнению FIFO
0x600002D0	16	DSPGPE14	Регистр управления линией GPIO14 по переполнению FIFO
0x600002E0	16	DSPGPS15	Регистр управления линией GPIO15 по наполнению FIFO
0x600002F0	16	DSPGPE15	Регистр управления линией GPIO15 по переполнению FIFO
0x60000400	2	DSPCFL0	Регистр нижнего порогового значения канального FIFO0
0x60000402	2	DSPCFH0	Регистр верхнего порогового значения канального FIFO0
0x60000404	2	DSPCFC0	Регистр управления канального FIFO0
0x60000406	2	DSPCFS0	Регистр статуса канального FIFO0
0x60000408	8		Регистры управления канального FIFO1
0x60000410	8		Регистры управления канального FIFO2
0x60000418	8		Регистры управления канального FIFO3
0x60000420	8		Регистры управления канального FIFO4
0x60000428	8		Регистры управления канального FIFO5
0x60000430	8		Регистры управления канального FIFO6
0x60000438	8		Регистры управления канального FIFO7
0x60000440	8		Регистры управления канального FIFO8
0x60000448	8		Регистры управления канального FIFO9
0x60000450	8		Регистры управления канального FIFO10
0x60000458	8		Регистры управления канального FIFO11
0x60000460	8		Регистры управления канального FIFO12
0x60000468	8		Регистры управления канального FIFO13
0x60000470	8		Регистры управления канального FIFO14
0x60000478	8		Регистры управления канального FIFO15
0x60000480	8		Регистры управления канального FIFO16

Адрес	Длина	Обозначение	Краткое описание
0x60000488	8		Регистры управления канального FIFO17
0x60000490	8		Регистры управления канального FIFO18
0x60000498	8		Регистры управления канального FIFO19
0x600004A0	8		Регистры управления канального FIFO20
0x600004A8	8		Регистры управления канального FIFO21
0x600004B0	8		Регистры управления канального FIFO22
0x600004B8	8		Регистры управления канального FIFO23
0x600004C0	8		Регистры управления канального FIFO24
0x600004C8	8		Регистры управления канального FIFO25
0x600004D0	8		Регистры управления канального FIFO26
0x600004D8	8		Регистры управления канального FIFO27
0x600004E0	8		Регистры управления канального FIFO28
0x600004E8	8		Регистры управления канального FIFO29
0x600004F0	8		Регистры управления канального FIFO30
0x600004F8	8		Регистры управления канального FIFO31
0x60000600	8		Регистры управления FIFOA
0x60000608	8		Регистры управления FIFOB
0x60000610	8		Регистры управления FIFOC
0x60000618	8		Регистры управления FIFOD
0x6C001000	2		Идентификатор основной платы
0x6C001002	2		Идентификатор DSPX
0x6C001004	2		Идентификатор DSPY
0x6C001006	2		Идентификатор мезонина ADC
0x6C001008	2		Идентификатор мезонина PDDC0
0x6C00100A	2		Идентификатор мезонина PDDC1
0x6C00100C	2		Идентификатор субмодуля RPU
0x6C00100E	2		Идентификатор субмодуля OG
0x6C001010	4		Серийный номер устройства
0x6C001014	2		Номер процессора
0xA0000000	8	DSPCFD0	Регистр доступа к данным канального FIFO0
0xA0000010	8	DSPCFD1	Регистр доступа к данным канального FIFO1
0xA0000020	8	DSPCFD2	Регистр доступа к данным канального FIFO2
0xA0000030	8	DSPCFD3	Регистр доступа к данным канального FIFO3
0xA0000040	8	DSPCFD4	Регистр доступа к данным канального FIFO4
0xA0000050	8	DSPCFD5	Регистр доступа к данным канального FIFO5
0xA0000060	8	DSPCFD6	Регистр доступа к данным канального FIFO6
0xA0000070	8	DSPCFD7	Регистр доступа к данным канального FIFO7
0xA0000080	8	DSPCFD8	Регистр доступа к данным канального FIFO8
0xA0000090	8	DSPCFD9	Регистр доступа к данным канального FIFO9
0xA00000A0	8	DSPCFD10	Регистр доступа к данным канального FIFO10
0xA00000B0	8	DSPCFD11	Регистр доступа к данным канального FIFO11
0xA00000C0	8	DSPCFD12	Регистр доступа к данным канального FIFO12
0xA00000D0	8	DSPCFD13	Регистр доступа к данным канального FIFO13
0xA00000E0	8	DSPCFD14	Регистр доступа к данным канального FIFO14
0xA00000F0	8	DSPCFD15	Регистр доступа к данным канального FIFO15
0xA0000100	8	DSPCFD16	Регистр доступа к данным канального FIFO16
0xA0000110	8	DSPCFD17	Регистр доступа к данным канального FIFO17
0xA0000120	8	DSPCFD18	Регистр доступа к данным канального FIFO18
0xA0000130	8	DSPCFD19	Регистр доступа к данным канального FIFO19
0xA0000140	8	DSPCFD20	Регистр доступа к данным канального FIFO20
0xA0000150	8	DSPCFD21	Регистр доступа к данным канального FIFO21
0xA0000160	8	DSPCFD22	Регистр доступа к данным канального FIFO22
0xA0000170	8	DSPCFD23	Регистр доступа к данным канального FIFO23
0xA0000180	8	DSPCFD24	Регистр доступа к данным канального FIFO24
0xA0000190	8	DSPCFD25	Регистр доступа к данным канального FIFO25
0xA00001A0	8	DSPCFD26	Регистр доступа к данным канального FIFO26
0xA00001B0	8	DSPCFD27	Регистр доступа к данным канального FIFO27
0xA00001C0	8	DSPCFD28	Регистр доступа к данным канального FIFO28

Адрес	Длина	Обозначение	Краткое описание
0xA00001D0	8	DSPCFD29	Регистр доступа к данным канального FIFO29
0xA00001E0	8	DSPCFD30	Регистр доступа к данным канального FIFO30
0xA00001F0	8	DSPCFD31	Регистр доступа к данным канального FIFO31
0xB0000000	8	DSPAFDA	Регистр доступа к данным канального FIFOA
0xB0000010	8	DSPAFDB	Регистр доступа к данным канального FIFOB
0xB0000020	8	DSPAFDC	Регистр доступа к данным канального FIFO C
0xB0000030	8	DSPAFDD	Регистр доступа к данным канального FIFO D

Внимание. Адреса, не указанные в данной таблице, зарезервированы. Обращение по этим адресам может вызвать непредсказуемые последствия вплоть до зависания компьютера.

Приложение II. Адресное пространство мезонинов

В этом разделе приведен перечень управляющих регистров. Подробное описание управляющих регистров приведено в справочном файле DSP64162.hlp.

Таблица III.1 Адресное пространство мезонина ADC-8-14

Адрес на шине SPI	Обозначение	Краткое описание
0x00	ADC8C0	Регистр коммутации канала А
0x01	ADC8C1	Регистр коммутации канала В
0x02	ADC8C2	Регистр коммутации канала С
0x03	ADC8C3	Регистр коммутации канала D
0x04	ADC8C4	Регистр коммутации канала Е
0x05	ADC8C5	Регистр коммутации канала F
0x06	ADC8C6	Регистр коммутации канала G
0x07	ADC8C7	Регистр коммутации канала H
0x08	ADC8DL	Управление настройками АЦП (младшие данные)
0x09	ADC8DH	Управление настройками АЦП (старшие данные)
0x0A	ADC8DA	Управление настройками АЦП (номер)
0x0B	ADC8FC	Регистр управления FIFO
0x0C	ADC8FS	Регистр статуса FIFO
0x18	ADC8FSC	Регистр переключения входа тактовой частоты
0x7D	ADC8FN	Номер версии
0x7F	ADC8TEST	Тестовый регистр

Таблица III.2 Адресное пространство мезонина ADC-2-12

Адрес на шине SPI	Обозначение	Краткое описание
0x00	ADC2C0	Регистр коммутации канала А
0x01	ADC2C1	Регистр коммутации канала В
0x02	ADC2C2	Регистр коммутации канала С
0x03	ADC2C3	Регистр коммутации канала D
0x04	ADC2C4	Регистр коммутации канала Е
0x05	ADC2C5	Регистр коммутации канала F
0x06	ADC2C6	Регистр коммутации канала G
0x07	ADC2C7	Регистр коммутации канала H
0x08	ADC2DDCI	Регистр управления входами DDC AD6636
0x20	ADC2DDC0D	Регистр данных DDC AD6636
0x22	ADC2DDC0AW	Регистр адреса на запись DDC AD6636
0x23	ADC2DDC0AR	Регистр адреса на чтение DDC AD6636
0x60	ADC2DDCRST	Регистр управления сбросом DDC AD6636
0x61	ADC2DDCSYNC	Регистр управления запуском DDC AD6636
0x7D	ADC2FN	Номер версии
0x7F	ADC2TEST	Тестовый регистр

Таблица ПП.3 Адресное пространство мезонина PDDC-16

Адрес на шине SPI	Обозначение	Краткое описание
0x00	PDDC16C	Регистр управления
0x01	PDDC16S	Регистр состояния
0x02	PDDC16T	Регистр управления тестовым генератором
0x03	PDDC16O	Регистр коммутации выходов
0x20	PDDC16L0	Регистр данных (D15 .. D0) DDC0
0x21	PDDC16H0	Регистр данных (D31 .. D16) DDC0
0x22	PDDC16W0	Регистр адреса DDC0 для записи
0x23	PDDC16R0	Регистр адреса DDC0 для чтения
0x28 - 0x2F	-	Регистры DDC1
0x30 - 0x37	-	Регистры DDC2
0x38 - 0x3F	-	Регистры DDC3
0x7D	PDDC16FN	Номер версии
0x7E	PDDC16FI	Идентификатор мезонина, считывается 0x21
0x7F	PDDC16TEST	Тестовый регистр

Таблица ПП.4 Адресное пространство мезонина PDDC-24

Адрес на шине SPI	Обозначение	Краткое описание
0x00	PDDC24C	Регистр управления
0x01	PDDC24S	Регистр состояния
0x02	PDDC24T	Регистр управления тестовым генератором
0x03	PDDC24OL	Регистр коммутации выходов LSB
0x04	PDDC24OH	Регистр коммутации выходов MSB
0x05	PDDC24SL	Регистр маски запуска LSB
0x06	PDDC24SH	Регистр маски запуска MSB
0x07	PDDC24S	Регистр синхронного индивидуального запуска
0x20	PDDC24L0	Регистр данных (D15 .. D0) DDC0
0x21	PDDC24H0	Регистр данных (D31 .. D16) DDC0
0x22	PDDC24W0	Регистр адреса DDC0 для записи
0x23	PDDC24R0	Регистр адреса DDC0 для чтения
0x28 - 0x2F	-	Регистры DDC1
0x30 - 0x37	-	Регистры DDC2
0x38 - 0x3F	-	Регистры DDC3
0x40 - 0x48	-	Регистры DDC4
0x48 - 0x4F	-	Регистры DDC5
0x7D	PDDC24FN	Номер версии
0x7E	PDDC24FI	Идентификатор мезонина, считывается 0x22
0x7F	PDDC24TEST	Тестовый регистр

Таблица III.5 Адресное пространство мезонина РПУ RRL-1

Адрес на шине SPI	Обозначение	Краткое описание
0x00	RPULE	Регистр сигнала ошибки
0x01	RPULS	Регистр статуса
0x02	RPULF	Частота настройки, МГц (900 .. 2500)
0x03	RPULIF	Промежуточная частота, МГц (70/140 - только чтение)
0x04	RPULG	Усиление аттенуаторов, 0.01 дБ (0 .. -5950)
0x05	RPULA	Включение АРУ, 0/1
0x06	RPULAG	Требуемый уровень выхода, 0.01 дБмВт
0x07	RPULOL	Уровень выхода, 0.01 дБмВт (только чтение)
0x08	RPULFOG	Опорная частота, МГц (1 .. 20)
0x7D	RPULFN	Номер версии
0x7E	RPULFI	Идентификатор мезонина, 0x40
0x7F	RPULTEST	Тестовый регистр

Таблица III.6 Адресное пространство мезонина РПУ RRL-2

Адрес на шине SPI	Обозначение	Краткое описание
0x00	RPULE	Регистр сигнала ошибки
0x01	RPULS	Регистр статуса
0x02	RPULF0	Частота настройки канала 1, МГц (900 .. 2500)
0x04	RPULG0	Усиление аттенуаторов канала 1, 0.01 дБ (0 .. -5950)
0x08	RPULOG	Опорная частота, МГц (1 .. 20)
0x22	RPULF1	Частота настройки канала 2, МГц (900 .. 2500)
0x24	RPULG1	Усиление аттенуаторов канала 2, 0.01 дБ (0 .. -5950)
0x7D	RPULFN	Номер версии
0x7E	RPULFI	Идентификатор мезонина, 0x41
0x7F	RPULTEST	Тестовый регистр

Таблица III.7 Адресное пространство мезонина RF FFD-1

Адрес на шине SPI	Обозначение	Краткое описание
0x00	OGE	Регистр сигнала ошибки
0x01	OGS	Регистр статуса
0x02	OGFL	Частота настройки, Гц (младшие 16 битов)
0x03	OGFH	Частота настройки, Гц (старшие 16 битов)
0x06	OGOUTPUT	Комутатор частоты (0/1 – внешняя/внутренняя)
0x08	RPULFOG	Опорная частота, МГц (1 .. 20)
0x7D	OGFN	Номер версии
0x7E	OGFI	Идентификатор мезонина, 0x51
0x7F	OGTEST	Тестовый регистр

Приложение III. Идентификационные коды функциональных узлов

В этом разделе приведен перечень идентификационных кодов основных функциональных узлов, возвращаемых функцией DSP6416_GetStatistics (раздел 4.2.10). Подробное описание соответствующих функциональных узлов и их технические характеристики приведены в [1].

Таблица III.1 Варианты основной платы RDMA

Идентификатор	Краткое описание
0x0000	RDMA ревизия 1.0 с установленным на плате опорным генератором 10 МГц “МТТ”
0x0001	RDMA ревизия 1.0 с установленным на плате опорным генератором 20 МГц “ГК85-ТС-1”
0x0002	RDMA ревизия 1.0 без опорного генератора

Таблица III.2 Варианты сигнальных процессоров

Идентификатор	Краткое описание
0x0000	TMS320C6416-600 с 64 Мб SDRAM
0x0001	TMS320C6416-720 с 64 Мб SDRAM
0x0002	TMS320C6416-840 с 64 Мб SDRAM
0x0003	TMS320C6416-1000 с 64 Мб SDRAM
0x0010	TMS320C6416-600 с 32 Мб SDRAM
0x0011	TMS320C6416-720 с 32 Мб SDRAM
0x0012	TMS320C6416-840 с 32 Мб SDRAM
0x0013	TMS320C6416-1000 с 32 Мб SDRAM
0x0020	TMS320C6416-600 без SDRAM
0x0021	TMS320C6416-720 без SDRAM
0x0022	TMS320C6416-840 без SDRAM
0x0023	TMS320C6416-1000 без SDRAM
0xFFFF	Соответствующий сигнальный процессор не установлен

Таблица ПП.3 Варианты мезонинов ADC

Идентификатор	Краткое описание
0x0000	ADC-8 ревизии 1.0, содержащий 8 АЦП ADS5500
0x0001	ADC-8 ревизии 1.1, содержащий 8 АЦП ADS5500
0x0002	ADC-8 ревизии 1.1, содержащий 4 АЦП ADS5500
0x0003	ADC-8 ревизии 1.1, содержащий 2 АЦП ADS5500
0x0004	ADC-8 ревизии 1.1, содержащий 1 АЦП ADS5500
0x0010	ADC-2 ревизии 1.0, содержащий 2 АЦП AD9430
0x0011	ADC-2 ревизии 1.1, содержащий 2 АЦП AD9430
0x0012	ADC-2 ревизии 1.1, содержащий 1 АЦП AD9430
0xFFFF	Мезонин ADC не установлен

Таблица ПП.4 Варианты мезонинов PDDC

Идентификатор	Краткое описание
0x0021	PDDC-16 ревизии 1.1, содержащий 4 ISL5216
0x0022	PDDC-24 ревизии 1.0, содержащий 6 ISL5216
0x0023	PDDC-24 ревизии 1.0, содержащий 4 ISL5216
0xFFFF	Соответствующий мезонин PDDC не установлен

Таблица ПП.5 Варианты субмодулей RF

Идентификатор	Краткое описание
0x0051	Субмодуль FFD-1
0x0071	Субмодуль FFD-2 с неуправляемым генератором частоты 40.000 МГц
0x0072	Субмодуль FFD-2 с неуправляемым генератором частоты 80.000 МГц
0x0073	Субмодуль FFD-2 с неуправляемым генератором частоты 93.(3) МГц
0x0074	Субмодуль FFD-2 с неуправляемым генератором частоты 100.000 МГц
0x0075	Субмодуль FFD-2 с неуправляемым генератором частоты 112.000 МГц
0x0076	Субмодуль FFD-2 с неуправляемым генератором частоты 125.000 МГц
0x0077	Субмодуль FFD-2 с неуправляемым генератором частоты 186.(6) МГц
0xFFFF	Субмодуль RF не установлен

Таблица ПП.6 Варианты субмодулей РПУ

Идентификатор	Краткое описание
0x0040	Субмодуль RRL-1 – одноканальный преобразователь частоты L диапазона
0x0041	Субмодуль RRL-2 – двухканальный преобразователь частоты L диапазона
0xFFFF	Субмодуль РПУ не установлен

Список литературы

1. “RDMA. Руководство по эксплуатации”, ООО "Резонанс-РД", 2005.
2. TMS320C6000 Peripherals Reference Guide. Literature Number: SPRU190D, Texas Instruments, February 2001.
3. Рихтер Дж. Windows для профессионалов: создание эффективных Win32 приложений с учетом специфики 64-разрядной версии Windows/Пер, англ - 4-е изд. - СПб; Питер; М.: Издательско-торговый дом "Русская Редакция", 2001. - 752 с.; ил.
4. TMS320C6000 Chip Support Library API User's Guide. Literature Number SPRU401C. Texas Instruments, October 2001.
5. TMS320C6000 Optimizing Compiler. User's Guide. Literature Number: SPRU187I. Texas Instruments, April 2001.
6. TMS320C6000 Programmer's Guide. Literature Number: SPRU198F. Texas Instruments, February 2001.
7. ISL5216 Datasheet. Intersil corporation, July 8 2005.
8. HSP50216 / ISL5216 EVAL Software Users manual. Intersil corporation, June 2001.
9. 150 MSPS, Wideband, Digital Downconverter (DDC) AD6636. Revision A. Analog Devices, 2005.

Рекомендуется для ознакомления.

10. Jeffrey Richter, “Programming for Microsoft® Windows®”, fourth edition, Microsoft Press, 1999.
11. Бьерн Страуструп “Язык программирования C++”, Невский Диалект, СПб., 2001.

Список изменений

Список изменений в версии 1.2

Страница 56

Ошибочно указана максимальная частота ввода данных из ISL5216 в мезонине PDDC-24.

Было

```
(2) Input Clock Rate..... тактовая частота АЦП в Гц
(4) Serial Clock Rate..... CLK
(5) Serial Clock Polarity... L->H @ Midbit
(8) Target device: ISL5216
```

Стало

```
(2) Input Clock Rate..... тактовая частота АЦП в Гц
(4) Serial Clock Rate..... CLK/4
(5) Serial Clock Polarity... L->H @ Midbit
(8) Target device: ISL5216
```

Лицензионное соглашение

ВАЖНО — ПРОЧТИТЕ ВНИМАТЕЛЬНО! Настоящее лицензионное соглашение является юридическим соглашением, заключаемым между Вами (физическим или юридическим лицом, далее «Вы») и ООО "Резонанс-РД" относительно сопровождаемого данным лицензионным соглашением программного продукта ООО "Резонанс-РД", включая любые носители данных, любые печатные материалы, а также любую «встроенную» или «электронную» документацию, исходные коды, библиотеки и примеры применения (далее «программное обеспечение»). Программное обеспечение включает также любые обновления, дополнительные программные средства и/или дополнения, которые могут быть предоставлены или доступны вам со стороны ООО "Резонанс-РД" или его официальных представителей.

Устанавливая, копируя, загружая, осуществляя доступ или иным образом используя указанное программное обеспечение, Вы тем самым принимаете на себя условия настоящего лицензионного соглашения. Если Вы не согласны с условиями настоящего лицензионного соглашения, Вы не имеете права устанавливать, осуществлять доступ или использовать данное программное обеспечение.

1. ЛИЦЕНЗИЯ НА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.

Данное программное обеспечение поставляется бесплатно для программной поддержки модулей серий RDMA и RDMB, выпускающихся ООО "Резонанс-РД". Программное обеспечение защищено законами и международными соглашениями о правах на интеллектуальную собственность. Данное программное обеспечение лицензируется, а не продается. Вы являетесь владельцем физического носителя информации, на котором записано или иным образом зафиксировано программное обеспечение, но права собственности на само программное обеспечение остаются за ООО "Резонанс-РД".

Разрешается устанавливать и использовать неограниченное количество копий программного обеспечения на любом компьютере, рабочей станции или ином электронном устройстве (далее "устройство"). Разрешается создать неограниченное количество копий и вносить в них любые модификации, при условии, что они не будут публиковаться или распространяться любым способом.

Разрешается внедрять данное программное обеспечение в создаваемые Вами программы и распространять эти программы, без каких либо отчислений в пользу ООО "Резонанс-РД", только в случае выполнения следующих условий:

- Вы гарантируете ООО "Резонанс-РД" освобождение от ответственности и защиту в связи с любыми претензиями, или исками, возникающими в результате использования или распространения использованных или измененных вами компонентов программного обеспечения, включая оплату расходов на юридические услуги;
- Указанное программное обеспечение работает только на устройствах, содержащих в своем составе модули серий RDMA или RDMB, выпускаемые ООО "Резонанс-РД".

Все остальные права на данное программное обеспечение принадлежат ООО "Резонанс-РД".

2. АВТОРСКИЕ ПРАВА.

Авторские права на программное обеспечение и данное руководство принадлежат ООО "Резонанс-РД" и защищены законодательством Российской Федерации. Копирование, сканирование и воспроизведение в любом, в том числе и электронном виде этого руководства возможно только с официального согласия ООО "Резонанс-РД".

3. ОТВЕТСТВЕННОСТЬ

ООО "Резонанс-РД" предоставляет программное обеспечение и услуги по его технической поддержке на условиях «как есть», со всеми неисправностями, и отказываются от всех других явных, подразумеваемых или предусмотренных законодательством гарантий и условий, включая (но не ограничиваясь только ими) отказ от подразумеваемой гарантии, обязательств или условий пригодности для продажи и применимости для определенной цели, точности или полноты ответов или результатов работы, гарантии высокой квалификации, отсутствия вирусов, отсутствия небрежности при изготовлении программного обеспечения, а также предоставления или непредоставления услуг по технической поддержке.

В наибольшей степени, допускаемой действующим законодательством, ни при каких обстоятельствах ООО "Резонанс-РД" не несет ответственности за какой-либо особый, случайный, косвенный или опосредованный ущерб или убытки, возникающие в связи с использованием или невозможностью использования данного программного обеспечения, либо оказанием или неоказанием услуг по технической поддержке, а так же в связи с любыми другими положениями данного лицензионного соглашения, даже в случае, если ООО "Резонанс-РД" было заранее извещено о возможности таких убытков. В любом случае, максимальный размер ответственности ООО "Резонанс-РД" не может превысить суммы, фактически уплаченной при приобретении данного программного обеспечения.

Важные замечания

ООО "Резонанс РД" оставляет за собой право модификации своих продуктов, и прекращать выпуск и поддержку без уведомления пользователей этих продуктов и предоставления им какой либо информации о возможных заменах или применению продукции третьих фирм.

ООО "Резонанс РД" ведет постоянную работу по улучшению своих продуктов, в том числе и сопроводительной документации, однако это не значит, что предоставляемые материалы полностью свободны от ошибок и обладают исчерпывающей полнотой. ООО "Резонанс РД" предоставляет техническую поддержку своих продуктов по электронной почте (Email: support@resonance.ru), но не гарантирует предоставления полной и исчерпывающей информации по возникающим у пользователей вопросам.

ООО "Резонанс РД" не несет ответственности за неправильное применение своих продуктов в составе других изделий и не несет ответственности за работоспособность этих изделий.

ООО "Резонанс РД" не несет ответственности за работоспособность и безопасность своих продуктов при нарушении максимальных рабочих режимов или условий эксплуатации.

Все зарегистрированные торговые марки и товарные знаки являются собственностью их правообладателей.