

Устройства ЦОС серии RDMB. Руководство по программированию.

Модули ЦОС серии RDMB.

Устройства ЦОС RDMB.

Руководство по программированию.

Версия 1.2.



Аннотация

Устройства ЦОС серии RDMB оптимизированы для приема многоканальных систем связи. 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 RDMB-1.

Содержание

У	СТРОЙС	ГВО ЦОС RDMB	1
P	уководо	СТВО ПО ПРОГРАММИРОВАНИЮ	1
BI	ЕРСИЯ 1.	2	
A	ннотан	ИЯ	2
C			2
U	ОДЕРЖА	ние	<i>L</i>
1	ОБЩИ	Е СВЕДЕНИЯ	
2	УСТАН	ЮВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	4
3	КРАТК	ОЕ ОПИСАНИЕ RDMB	5
	31 Ст	NVKTVNHAG CYPMA RDMR	5
	311	Схема управления	5
	3.1.2	Схема уприоления Схема передачи данных	
	3.1.3	Схема обработки прерываний	6
	3.1.4	Схема внешней синхронизации.	
	3.2 Yc	тройство FIFO	8
4	гигш	I AOTERA VIIDAD HEIHIG DSD(416)	10
4	DIIDJII	ЮТЕКА У ПРАБЛЕНИЯ DSP04102	
	4.1 06	щие сведения	
	4.2 Pa	бота с библиотекой	
	4.2.1	Подключение виблиотеки к проекту	
	4.2.2	Подключение к устройству	
	4.2.3	Инициализация устройства	
	4.2.4	Загрузка программ ТМ\$320С6416	
	4.2.5	Загрузка прошивок ПЛИС	
	4.2.6	Доступ в адресное пространство TMS320C6416	
	4.2.7	Обработка прерываний	
	4.2.8	Генерация прерывания для TMS320C6416	
	4.2.9	Передача данных через РСІ с использованием DMA	
	4.2.10	Получение справочной информации	
	4.2.11	Работа с генератором частоты дискретизации	
	4.2.12	Работа с АЦП	
	4.2.13	Работа с цифровыми приемниками	
	4.3 Ile	речень функций	
	4.3.1	Функции инициализации	
	4.3.2	Функции доступа к DSP через окна	
	4.3.3	Функции потокобезопасного доступа к DSP	
	4.3.4	Функции управления DSP	
	4.3.5	Функции сигнализации	
	4.3.0	Функции работы с контроллером DMA	
	4.3.7	Функции работы с АЦП	
	4.3.8	Функции работы с генератором частоты дискретизации	
	4.3.9	Функции работы с DDC	



5	ПР	ОГРАММА МОНИТОР DSP6416-2	
	5.1	Назначение программы	
	5.2	Управление молулем RDMB	
	5.2.	гариания модуном за настрания 1 Общие функции	
	5.2.	2 Управление сигнальными проиессорами	
	5.2.	3 Управление DDC	
	5.2.	4 Сигнал с АЦП	40
	5.2.	5 Сигнал с DDC	41
	5.3	Проверка функционирования RDMB	41
	5.3.	1 Экспресс-проверка	41
6	PAI	5ОТА С ПРОЦЕССОРАМИ TMS320C6416	42
,	6.1	Общие сведения	
	6.2	Примеры программ	
	6.2.	1 Простейшая программа	
	6.2.	2 Использование системы прерываний PCI	
	6.2.	3 Использование контроллера DMA на шине PCI	
	6.3	Работа с устройствами на плате RDMB	44
	6.3.	1 Управление FIFO	44
	6.3.	2 Считывание данных	45
	6.3.	3 Доступ в адресное пространство DDC	46
	6.3.	.4 Использование последовательных портов	
7	PAI	50TA C DDC	
	7.1	Общие свеления	
	7.2	О одно сводения по настройке DDC ISL 5216	
	7.2.	I Общие сведения	
	7.2.	2 Работа с программой конфигурации	48
	7.2.	3 Ограничения	
	7.2.	4 Доступ к данным	49
8	ДО	ПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ	
П	рило	ЖЕНИЕ І. АДРЕСНОЕ ПРОСТРАНСТВО DSP	
П	рило	ЖЕНИЕ П. ИДЕНТИФИКАЦИОННЫЕ КОДЫ ФУНКЦИОНАЛЬНЫХ УЗЛОВ	
СТ	исс	и питературы	56
		лу изменений	
U	Cance	л полненения в вавали 1.2	
JH	ацен	ІЗИОННОЕ СОГЛАШЕНИЕ	
BA	ЖНІ	ЫЕ ЗАМЕЧАНИЯ	59



1 Общие сведения

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

Данное программное обеспечение позволяет в полной мере использовать возможности RDMB, а так же позволяет управлять всеми ресурсами устройства.

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

Проектирование программного обеспечения для сигнальных процессоров TMS320C6416, установленных в RDMB, требует знания устройства RDMB [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 в соответствии с указаниями, содержащимися в файле "install.txt" в корневом каталоге прилагаемого компакт-диска.



3 Краткое описание RDMB

Полное описание RDMB приведено в [1]. В этом разделе приведено описание с точки зрения вопросов взаимодействия аппаратуры и пользовательского программного обеспечения RDMB.

3.1 Структурная схема RDMB

RDMB может поставляться в различной конфигурации, которая определяется типом установленных дополнительных плат (мезонинов). Таким образом, наличие тех или иных функциональных блоков внутри RDMB (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 RDMB приведена на рисунке 3.1. RDMB может поставляться в различной конфигурации, которая определяется типом установленных дополнительных плат (мезонинов). Таким образом, наличие тех или иных функциональных блоков внутри RDMB (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 RDMB на шине PCI32 версии не ниже 2.1 с частотой 33 МГц.

Сигнальный процессор TMS320C6416 подключен к контроллеру PCI через HOST-порт. Контроллер PCI формирует два окна в адресном пространстве PCI. Протяженность каждого окна составляет 8 Мбайт. На это окно может быть отображен любой участок внутреннего адресного пространства 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 RDMB 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 RDMB

Доступ к управляющим регистрам ISL5216, а так же к регистрам конфигурации RDMB производится по 16 разрядной параллельной асинхронной шине через интерфейс EMIFB процессора.

3.1.2 Схема передачи данных

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

Входные аналоговые сигналы (от 1 до 4) поступают на входы АЦП. Отсчеты от АЦП через полнодоступный коммутатор каналов АЦП поступают на входы ISL5216 и на входы 4-х FIFO АЦП.

В RDMВ присутствуют два вида потоков данных. Первый вид – необработанные отсчеты от АЦП (до 4 потоков). Второй вид – обработанные в ISL5216 данные (до 16 потоков).



Данные с выходов ISL5216, обработанные входящими в их состав цифровыми канальными РПУ, поступают на канальные FIFO DDC. Канальные FIFO DDC и FIFO АЦП подключены через 32 разрядную параллельную синхронную шину EMIFA с производительностью около 600 Мбайт/с.



Рисунок 3.2 Структурная схема передачи потоков данных в RDMB

Скоростная передача данных через PCI осуществляется следующим образом. RDMB содержит контроллер DMA. Процессор подключен 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 RDMB, 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.

Процессор TMS320C6416 имеет 16 линий ввода/вывода (GPIO0 .. GPIO15), изменение уровня сигнала на которых может вырабатывать прерывание для TMS320C6416 или сигнал для контроллера EDMA. В устройстве RDMB эти линии используются следующим образом.

Линия GPIO0 процессора всегда подключена к FIFO контроллера PCIDMA. На линии GPIO0 формируется импульс при освобождении FIFO ниже нижней границы.

Линии GPIO0 .. 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 .. FIFO15, FIFOA .. FIFOD). Коммутатор позволяет подключить к любой линии GPIO4 .. GPIO15 каждого из процессоров DSP0, DSP1 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 RDMB

3.1.4 Схема внешней синхронизации

В модуле RDMB ревизии 2 введена схема внешней синхронизации. Эта схема позволяет использовать сигналы с любого из входов внешней синхронизации SYNC0 .. SYNC4 в качестве любого из внутренних сигналов запуска каналов DDC, запуска FIFO DDC, FIFO АЦП и сигналов на линиях GPIO0 .. GPIO15 процессора TMS320C6416. Схема формирования синхроимпульсов со входа SYNC0 представлена на рисунке 3.4.



Рисунок 3.4. Схема использования сигналов внешней синхронизации.

Сигнал со входа SYNC0 разветвляется на 52 приемника через переключатель, который управляется одним разрядом в регистре "SYNC0" (Приложение I). Сигналы от входов SYNC1 .. SYNC4 обрабатываются аналогично. Сигналы от нескольких входов SYNC, поступающие на один приемник и сигналы от регистров управления складываются по схеме "ИЛИ". Это приводит к тому, что если вход SYNC0 подключен к сигналу SYNC0 DDC0 и на этом входе все время присутствует "1", запустить канал 0 DDC0 с помощью доступа к регистру DSPCFST0 не удастся.



При использовании внешнего запуска имейте ввиду, что каналы DDC, FIFO DDC и FIFO АЦП запускаются передним фронтом импульса.

Линии синхронизации могут быть настроены на ввод и на вывод с помощью регистра SYNCCTRL. По умолчанию они настроены на ввод. Если в разряд D0 регистра SYNCCTRL записать "1", линия синхронизации SYNC0 будет настроена на вывод, и управлять ее состоянием можно будет с помощью записи в разряд D0 регистра SYNCDATA. Если линия настроена на вывод, то состояние соответствующего разряда в регистре SYNCDATA не только транслируется на разъем "SYNC", но и поступает на внутреннюю схему.

Таким образом, соединив разъемы "SYNC" нескольких модулей RDMB плоским кабелем и настроив в одном из них линии SYNC на вывод, а во всех остальных на ввод, можно одновременно во всех платах запустить каналы DDC, или накопление отсчетов АЦП, или сформировать прерывание процессора.

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 RDMB имеют одинаковое устройство, показанное на рисунке 3.5.



Рисунок 3.5

При поступлении данных текущее положение указателя FIFO перемещается вверх. Когда указатель достигает верхней границы, флаг наполнения FIFO устанавливается в "1" (FIFO_FULL). Когда указатель достигнет нижней границы, флаг 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.

Сигнал FIFO_FULL, поступающий на выводы GPIO формируется несколько иначе. При поступлении каждых N отсчетов (N – значение, установленные в качестве верхней границы) формируется короткий импульс, который может служить сигналом прерывания или событием для запуска контроллера EDMA.

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

Таблица 3.1

Максимальная	Нижняя	Верхняя	Примечание			
емкость	граница	граница				
512 x 32	16	256	считывается через EMIFA			
FIFO15 512 x 32 16 256 считывается через EMIFA						
512 x 32	16	256	считывается через EMIFA			
512 x 32	16	256	считывается через EMIFA			
512 x 32	16	256	считывается через EMIFA			
512 x 32	16	256	считывается через EMIFA			
FIFO PCIDMA						
1024 x 16	256*	64*	Запись через EMIFB			
	Максимальная емкость 512 x 32 512 x 32 512 x 32	Максимальная емкость Нижняя граница 512 x 32 16 512 x 32 16	Максимальная емкость Нижняя граница Верхняя граница 512 x 32 16 256 1024 x 16 256* 64*			

Примечание * - значение соответствует количеству свободного места в FIFO.



Регистр управления в настоящее время 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 наполнения до верхней границы.
- Разряд D15 сигнал опустошения до нижней границы.



4 Библиотека управления DSP64162

4.1 Общие сведения

Библиотека DSP64162.dll (далее библиотека управления) предназначена для управления модулем RDMB 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 RDMB, абстрагируясь от особенностей реализации конкретных узлов RDMB.

Библиотека предоставляет функции для доступа в адресное пространство всех узлов и модулей, входящих в состав RDMB, 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 и в проектах на других языках программирования, например на Borland Delphi, но в этом случае прототипы функций придется описывать самостоятельно, пользуясь образцами в файле "DSP64162.h".

Внимание. Эта библиотека предназначена для поддержки устройств семейства RDMB и MBDSP-64T-2. Поэтому некоторые функции, например, функции управления преобразователем частоты, с устройством RDMB работать не будут.



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 RDMB.

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.

Очень хорошо этот способ описан в книге "Microsoft Windows для профессионалов" Джеффри Рихтера [3].

4.2.2 Подключение к устройству

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

Обратите внимание, что если в компьютере установлено несколько RDMB, 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.

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

Использование этих функций демонстрируется в примере Sample1.

4.2.3 Инициализация устройства

На плате RDMB расположены процессоры TMS320C6416 и ПЛИС, которые требуют первоначальной инициализации. Такая инициализация должна выполняться после включения питания до начала любых других операций с устройством. Кроме того, различные мезонины так же могут требовать инициализации после включения питания. Такую инициализацию 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 RDMB.

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.

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.

4.2.5 Загрузка прошивок ПЛИС

Для загрузки прошивок в ПЛИС, установленные на плате, предназначены функции RDSP64162_CPLDLoad, RDSP64162_CPLDLoadFile, RDSP64162_CPLDLoadResource. Они работают аналогично функциям загрузки программ в TMS320C6416. Формат входных файлов – Altera raw binary file.

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 процессора TMS320C6416 осуществляется через "окна" в адресном пространстве PCI. Контроллер PCI формирует по 2 независимых "окна". 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, использование окна 1 не рекомендуется, поскольку оно 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 Мбайтов нет ресурсов, которые бы требовали постоянного доступа со стороны пользователя через шину PCI. Поэтому рекомендуется использовать окно 0, настраивая его на базовый адрес 0x00000000.

Использование этих функций демонстрируется в примере Sample1.

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.

4.2.8 Генерация прерывания для TMS320C6416

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

Модули ЦОС серии RDMB. Руководство по программированию.

Использование этих функций демонстрируется в примере Sample2.

12



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 RDMB, превышает 2 Мбайта в секунду. В настоящее время реализована передача данных только от RDMB.

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.

4.2.10 Получение справочной информации

С помощью функции RDSP64162_GetStatistics можно получать самую разнообразную дополнительную информацию об устройстве. В частности номер PCI слота, 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 генератором частоты дискретизации

RDMB содержат высококачественный перестраиваемый формирователь тактовой частоты для АЦП. Для управления им служат функции RDSP64162_OG, которые позволяют устанавливать тактовую частоту генератора, определять наличие сигнала захвата синтезатора, включать и выключать использование внешней тактовой частоты.

4.2.12 Работа с АЦП

Для управления АЦП 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.

4.2.13 Работа с цифровыми приемниками

RDMB содержат до 16 независимых каналов цифрового радиоприема на базе ISL5216.

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.



4.3 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 RDMB. Вызов этой функции должен предварять вызов любой другой функции библиотеки для соответствующего устройства.

Параметры функции

board [in] Порядковый номер устройства серии RDMB, 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.

Замечания

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 RDMB. Вызов этой функции должен завершать работу программы пользователя.

HRESULT __stdcall RDSP64162_Done(ULONG board);

Параметры функции

board [in] Порядковый номер устройства серии RDMB, 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.

Замечания

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 RDMB и его текущей конфигурации.

Внимание

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

Параметры функции

- board [in] Порядковый номер устройства серии RDMB, 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.

Замечания

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 RDMB с соответствующим порядковым номером.

> HRESULT __stdcall RDSP64162_IsPresent(ULONG board);

Параметры функции

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

index [in] Индекс запрашиваемого параметра.

Название

Описание

DSP64162_STAT_PCISLOT	Номер РСІ 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, °С.
DSP64162_STAT_TEMPERATURE2	Температура кристалла ПЛИС, °С.
DSP64162_STAT_RPM1	Частота вращения вентилятора, RPM.
DSP64162_STAT_TYPEBOARD	Идентификатор основной платы.
DSP64162_STAT_TYPEDSP0	Идентификатор процессора DSP0.
DSP64162_STAT_TYPEDSP1	Идентификатор процессора DSP1.
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 RPU.
pdata [out] Указатель на пере	менную, в которой будет сохранен результат запроса.

Возвращаемое значение



4.3.1.6 Функция RDSP64162_InitDevice

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

Параметры функции

- board [in] Порядковый номер устройства серии RDMB, 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 DSP0.
RDSP64162_DSP1	Настроить DSP1.
RDSP64162_CPLD	Настроить загружаемую ПЛИС.
RDSP64162_ADC	Настроить мезонин АДС
RDSP64162_PDDC0	Настроить мезонин PDDC-1
RDSP64162_PDDC1	Настроить мезонин PDDC-2
RDSP64162_RPU	Настроить мезонин RPU
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.

Замечания

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.

Параметры функции

board [in] Порядковый номер устройства серии RDMB, 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.

Замечания

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.

Параметры функции

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

number [in] Номер окна. Номер окна начинается с 0.

Возвращаемое значение



4.3.2.3 Функция RDSP64162_GetWindows

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

Параметры функции

- board [in] Порядковый номер устройства серии RDMB, 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.

Замечания

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 RDMB, 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.

Замечания



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 DSP0 и DSP1.

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

Параметры функции

 board [in]
 Порядковый номер устройства, 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_AddressRead предназначена для потокобезопасного доступа в адресное пространство сигнальных процессоров DSP0 и DSP1.

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

Параметры функции

board [in]	Порядковый номер устройства, 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



4.3.3.3 Функция RDSP64162_AddressWrite

Функция RDSP64162_AddressRead предназначена для потокобезопасного доступа в адресное пространство сигнальных процессоров DSP0 и DSP1.

HRESULT __stdcall RDSP64162_AddressWrite(ULONG board, ULONG number, ULONG address, USHORT data);

Параметры функции

board [in] Порядковый номер устройства, 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 Е FAIL.

4.3.3.4 Функция RDSP64162_AddressWriteBlock

Функция RDSP64162_AddressRead предназначена для потокобезопасного доступа в адресное пространство сигнальных процессоров DSP0 и DSP1.

Параметры функции

board [in]	Порядковый номер устройства, 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



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 DSP6416-2.

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

Параметры функции

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

flags	[in]	1	Іоказывает,	какие ус	стройства	следует	сбросить.	. Допустимы	е значения	приведены	в таблице.
-------	------	---	-------------	----------	-----------	---------	-----------	-------------	------------	-----------	------------

Название	Описание
RDSP64162_DSP0	Сбросить DSP0.

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 .

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

Параметры функции

- board [in] Порядковый номер устройства серии RDMB, 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 DSP0.

Возвращаемое значение



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	<pre>stdcall RDSP64162_ProgramLoad(</pre>
	ULONG board,
	ULONG flags,
	void* pdata,
	ULONG length,
	BOOL bStart = TRUE
):	

Параметры функции

- board [in] Порядковый номер устройства серии RDMB, 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 DSP0.

Перечисленные флаги можно комбинировать.

pdata [In] Указатель на область памяти, в которой содержится битовый образ програм	odata [in]	Указатель на область	памяти, в которой содержится	битовый образ программ
--	------------	----------------------	------------------------------	------------------------

- 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.

Замечания

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.

<pre>HRESULTstdcall RDSP64162_ProgramLoadFile(</pre>
ULONG board,
ULONG flags,
LPCWSTR filéname.
BOOL bStart = TRUE
):

Параметры функции

- board [in] Порядковый номер устройства серии RDMB, 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 DSP0.

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.

Замечания

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 RDMB, 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 DSP0.

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.

Замечания

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 сигнализации

Функции сигнализации предназначены для регистрации пользовательских обработчиков сигналов прерывания.

4.3.5.1 Функция RDSP64162_InstallInterruptHandler

Функция RDSP64162_InstallInterruptHandler предназначена для подключения обработчика прерываний от устройства.

Параметры функции

- board [in] Порядковый номер устройства серии RDMB, установленного в системе. Порядковый номер начинается с 0.
- pHandler [in] Адрес функции типа RDSP64162_FUNCTION_HANDLER, которая будет вызываться при приходе прерывания от устройства. 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.

4.3.5.2 RDSP64162_FUNCTION_HANDLER

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

Параметры функции

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

Название	Описание
RDSP64162_INTERRUPT_DSP0	Прерывание от DSP0.
RDSP64162_INTERRUPT_DMA0	Прерывание от канала DMA0.
RDSP64162_INTERRUPT_ERRORDMA0	Прерывание по ошибке от
	канала DMA0.

Перечисленные флаги могут быть комбинированы.

Возвращаемое значение



4.3.6 Функции работы с контроллером DMA

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

4.3.6.1 Функция RDSP64162_DMARead

Функция RDSP64162_DMARead предназначена для передачи блока данных от 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 numbe length flags [i	[in] I Her[in] I [in] J in] H	Іорядковый на ачинается с 0 Іорядковый на Ілина в байтах Іабор флагов.	дковый номер устройства серии RDMB, установленного в системе. Порядковый но нается с 0. дковый номер канала. Номер 0 соответствует DSP0. а в байтах принимаемого блока данных. р флагов. Допустимые значения приведены в таблице.			
Название DSP64162_N	IASTER	_START	Описание В начале передачи блока FIFO будет сброшено			
DSP64162_M	IASTER	_CONTINUE	В начале передачи блока FIFO не будет сброшено. Таким образом, следующий блок будет передан без разрыва с предыдущим.			
DSP64162_N	IASTER	_STOP	В конце передачи блока FIFO будет сброшено.			
DSP64162_M Переч	IASTER исленны	_TEST ie флаги можн	Передача данных от тестового генератора. по 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.

Замечания

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 дождаться 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.



4.3.6.2 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 RDMB, установленного в системе. Порядковый номер
	начинается с 0.
number [in]	Порядковый номер канала DMA. Номер 0 соответствует DSP0, номер 1 соответствует DSP1.
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.

Замечания

В настоящее время на устройствах серии RDMB 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.

4.3.6.3 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 RDMB, 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



4.3.6.4 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 RDMB, установленного в системе. Порядковый номер начинается с 0.

number [in] Порядковый номер канала DMA.

Возвращаемое значение

В случае успешного выполнения функция возвращает S_OK. В случае ошибки функция возвращает Е FAIL.

4.3.7 Функции работы с АЦП

Функции работы с АЦП предназначены для управления АЦП.

4.3.7.1 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 RDMB, установленного в системе. Порядковый номер начинается с 0.
number [in] flags [in]	Порядковый номер АЦП. Порядковый номер начинается с 0. 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_PI	LLON 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.

Замечания

Некоторые опции работают не на всех видах мезонинов ADC.



4.3.7.2 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 RDMB, 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.

Замечания

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

4.3.7.3 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 RDMB, 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.

Замечания

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.



4.3.8 Функции работы с генератором частоты дискретизации

4.3.8.1 Функция RDSP64162_OGSetFrequency

Функция RDSP64162 OGSetFrequency предназначена для установки частоты дискретизации.

ULONG board.	(
olonio source,	
double frequency	

Параметры функции

board [in] Порядковый номер устройства, 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.

Замечания

Диапазон допустимых значений для управляемых параметров зависят от вида устройства.

Конструкция опорного генератора 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.

4.3.8.2 Функция RDSP64162_OGGetFrequency

Функция RDSP64162_OGGetFrequency возвращает текущее значение частоты дискретизации.

Параметры функции

board [in] Порядковый номер устройства, 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 Е FAIL.

Замечания

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

Конструкция опорного генератора 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.



4.3.8.3 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, установленного в системе. Порядковый номер начинается с 0.
input [in]	Набор флагов, позволяющий выбрать источник частоты дискретизации.
	DSP64162_OGINPUT_EXTERNAL – вход внешней частоты.
	DSP64162_OGINPUT_INTERNAL – встроенный генератор.
	DSP64162_OGSYNTH_5MHZ – встроенный генератор с синхронизацией от 5 МГц.
	DSP64162_OGSYNTH_10MHZ – встроенный генератор с синхронизацией от 10 МГц.

Возвращаемое значение

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

4.3.8.4 Функция RDSP64162_OGGetInput

Функция RDSP64162_GetInput возвращает номер источника частоты дискретизации.

Параметры функции

board [in]	Порядковый номер устройства, установленного в системе. Порядковый номер начинается с 0.
input [out]	Указатель на переменную, в которой возвращается набор флагов, позволяющий установить
	источник частоты дискретизации.
	DSP64162_OGINPUT_EXTERNAL – вход внешней частоты.
	DSP64162_OGINPUT_INTERNAL – встроенный генератор.
	DSP64162_OGSYNTH_5MHZ – встроенный генератор с синхронизацией от 5 МГц.
	DSP64162_OGSYNTH_10MHZ – встроенный генератор с синхронизацией от 10 МГц.

Возвращаемое значение

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

4.3.8.5 Функция RDSP64162_OGGetStatus

Функция RDSP64162_OGGetStatus предназначена для получения информации о состоянии встроенного генератора частоты дискретизации.

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

Параметры функции

board [in] Порядковый номер устройства, 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.

Модули ЦОС серии RDMB. Руководство по программированию.





4.3.9 Функции работы с DDC

Функции работы с DDC предназначены для управления DDC ISL5216.

4.3.9.1 Функция RDSP64162_PDDCLoad

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

Параметры функции

board [in]	Порядковый номер устройства, установленного в системе. Порядковый номер начинается с 0.
number [m]	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.

Замечания

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

4.3.9.2 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, установленного в системе. Порядковый номер начинается с 0. питьег [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.

Замечания

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.



4.3.9.3 Функция RDSP64162_PDDCLoadResource

Функция RDSP64162_PDDCLoadResource предназначена для загрузки конфигурации DDC из ресурса.

Параметры функции

board [in]	Порядковый номер устройства, 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.

Замечания

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.

4.3.9.4 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, 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



4.3.9.5 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, 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 Е FAIL.

4.3.9.6 Функция RDSP64162_PDDClsPresent

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

Параметры функции

board [in] Порядковый номер устройства, 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 соотвествующего DDC возвращает S ОК.

В случае отсутствия соотвествующего DDC возвращает S FALSE.

В случае ошибки функция возвращает E_FAIL.



4.3.9.7 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]	Поряд	ковы й	і номер устрої	іства, у	установлени	ного в систе	ме. Поряді	ковый ног	мер начинае	тся с 0.
number [in]	Поряд	іковый	і номер канала	a DDC,	, установлен	ного на пла	те. Поряди	ковый ном	мер начинае	тся с 0.
	Если	одна	микросхема	DDC	содержит	несколько	каналов,	следует	указывать	номер
	наиме	еньшег	о канала.							
address [in]	Алрес	в алр	есном простра	анстве	DDC					

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.

Замечания

Для доступа используются функции RDSP64162_SerialRead, RDSP64162_SerialWrite.

4.3.9.8 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, 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.

Замечания

Для доступа используются функции RDSP64162_SerialRead, RDSP64162_SerialWrite.



5 Программа монитор DSP6416-2

5.1 Назначение программы

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

5.2.1 Общие функции

При выборе в списке, расположенном в главном окне программы справа, пункта "Управление", пользователь получает возможность управлять параметрами, касающимися всего устройства RDMB. 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 RDMB.

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", пользователь получает возможность управлять соответствующим процессором TMS320C6416, установленным в RDMB. 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 TMS320C6146

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 TMS320C6146, 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" загорается, когда приходит прерывание от TMS320C6416. Индикатор автоматически гаснет через 200 мс, поэтому если прерывания поступают с меньшим интервалом, индикатор будет гореть 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

Вкладка "DDC" позволяет 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 DCC

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 7.2, 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 АЦП, установленного в модуле RDMB.

5.2.4.1 Переключение каналов

С помощью кнопок "1".. "4" можно включать или выключать отображение сигнала с соответствующих входов мезонина АЦП. Переключать каналы так же можно с помощью клавиш "1".. "4" 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" .. "4". 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" .. "4". 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 4096 частотных точек. 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 - \uparrow " и "Ctrl - \downarrow " на клавиатуре. Чтобы изменить масштаб графика по горизонтали, нажимайте клавиши "Ctrl - \leftarrow " и "Ctrl - \rightarrow " 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 - \uparrow " и "Alt - \downarrow " на клавиатуре. Чтобы переместить график по горизонтали, нажимайте клавиши "Alt - \leftarrow " и "Alt - \rightarrow " 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 сигнала или спектра 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 плате устройства. 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, сигнал с какого цифрового РПУ 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 - \uparrow " и "Ctrl - \downarrow " на клавиатуре. Чтобы изменить масштаб графика по горизонтали, нажимайте клавиши "Ctrl - \leftarrow " и "Ctrl - \rightarrow " 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 - \uparrow " и "Alt - \downarrow " на клавиатуре. Чтобы переместить график по горизонтали, нажимайте клавиши "Alt - \leftarrow " и "Alt - \rightarrow " 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 сигнала или спектра 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.

5.3 Проверка функционирования RDMB

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".



6 Работа с процессорами TMS320C6416

6.1 Общие сведения

Используемый в устройстве RDMB процессор TMS320C6416 или TMS320C6414 фирмы Texas Instruments представляет собой процессор для цифровой обработки сигналов с 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 RDMB 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 С 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 "hex6x.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.



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 1 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 РСІ 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.

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 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_FIF0_LENGTH/2; i++) PCIDMA_FIF0 = current++;
}</pre>
```

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.

```
GPI0_pinEnable(GPI0_PIN0);
GPI0_pinDirection(GPI0_PIN0, GPI0_INPUT);
GPI0_intPolarity(GPI0_PIN0, GPI0_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 ровно 512 16 разрядных слов. Если будет записываться больше слов, FIFO переполнится, если меньшее – FIFO опустошится и прерывания перестанут поступать.

6.3 Работа с устройствами на плате RDMB

Основными блоками RDMB, доступными из TMS320C6416, являются канальные FIFO (FIFO0 .. FIFO15) и FIFO AЦП (FIFOA .. FIFOD).

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 приложении.

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 Управление FIFO

Каждое FIFO АЦП имеет несколько собственных регистров управления: регистр для записи минимального 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 .. FIFO15 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 16 линий GPIO, подключенных к сигналам FIFO_FULL и FIFO_ERROR через коммутатор (раздел 3.1.3) могут генерировать прерывание для TMS320C6416 или событие для запуска контроллера EDMA. 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 запущенны 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 приступать 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 DDC

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. Доступ осуществляется путем обращения к регистрам управления, расположенным в адресном пространстве TMS320C6416.

Распределение адресного пространства приведено в приложении.



6.3.4 Использование последовательных портов

Последовательные порты MBSP0 и MBSP1 сигнального процессора на специальные разъемы "SERIAL PORT 1" и "SERIAL PORT 2" соответственно.

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 настроен на формирование тактовой частоты, приемник соединенного с ним порта 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 RDMB, строго следуйте рекомендациям, приведенным в [1]. Неправильное соединение способно вывести из строя RDMB или подключенные к нему устройства.



Работа с DDC 7

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.

7.2 Рекомендации по настройке DDC ISL5216

7.2.1 Общие сведения

Всего в устройстве RDMB, может содержаться до 4 ISL5216, что обеспечивает 16 каналов цифрового преобразования частоты и фильтрации. Микросхемы DDC ISL5216 имеют множество возможностей по настройке, однако в составе RDMB 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 RDMB 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 следующим образом.

Таблица 7.1

Входы ISL5216	Входы устройства
INA[15:0]	Вход АЦП "1″
INB[15:0]	Вход АЦП "2″
INC[15:0]	Вход АЦП "З"
IND[15:0]	Вход АЦП "4″

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

Input Clock Rate..... тактовая частота АЦП в Гц Serial Clock Rate..... CLK Serial Clock Polarity... L->H @ Midbit 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.

Sample Rate.... тактовая частота АЦП в Гц
 Input Source... AIN, BIN, CIN, DIN
 Input Format... Twos Comp.,real
 ... Fxd Point
 Demux Delay.... 0

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

1) 2)	Sync Sync	Active Positi	on Ear	n Iy			
5)	outpu		ly 0				
υτρι	IT_I:				ουτρυτ 2:		
	Slot	Туре	Bits	Sync	S	lot Type	Bits
(8)	1	любое	16 fixed	Ý	(15)	1 любое	16 fixed
9)	2	zeros	0 fixed	Ν	(16)	2 zeros	0 fixed
10)	3	zeros	0 fixed	Ν	(17)	3 zeros	0 fixed
11)	4	zeros	0 fixed	Ν	(18)	4 zeros	0 fixed
12)	5	zeros	0 fixed	Ν	(19)	5 zeros	0 fixed
13)	6	zeros	0 fixed	Ν	(20)	6 zeros	0 fixed
14)	7	zeros	0 fixed	N	(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 O	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 с помощью стандартных фильтров HBF1 ISL5216.

7.2.4 Доступ к данным

C

Выходные данные из ISL5216 доступны через канальные FIFO0 .. FIFO15. 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.



8 Дополнительная информация

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

Copyright © 2006 ООО "Резонанс-РД".



Приложение І. Адресное пространство DSP

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

Таблица	ПІ.1
---------	------

Апрес	Лпина	Обознанение	Кратиое описание
	Длипа 1М	Обозначение	
0x00000000	1000M		Внутренняя намять или ксш L2 Вопистры конфигирации произосора TMS22006416
0x01000000	2		Регистры конфигурации процессора 11/15/2000410
0x6C000010	2	DSPCFS10	
0x6C000020	2	DSPCFS15	Регистр признака наполненности FIFO0 FIFO15
0x6C000028	2	DSPCFSTE	Регистр признака переполнения FIFO0 FIFO15
0x6C000030	2	DSPAFS	Регистр запуска FIFOA D
0x6C000032	2	DSPAFSTS	Peructp ctatyca FIFOA D
0x6C000100	16	DSPGPS0	Регистр управления линией GPIO0 по наполнению FIFO
0x6C000110	16	DSPGPE0	Регистр управления линией GPIO0 по переполнению FIFO
0x6C000120	16	DSPGPS1	Регистр управления линией GPIO1 по наполнению FIFO
0x6C000130	16	DSPGPE1	Регистр управления линией GPIO1 по переполнению FIFO
0x6C000140	16	DSPGPS2	Регистр управления линией GPIO2 по наполнению FIFO
0x6C000150	16	DSPGPE2	Регистр управления линией GPIO2 по переполнению FIFO
0x6C000160	16	DSPGPS3	Регистр управления линией GPIO3 по наполнению FIFO
0x6C000170	16	DSPGPE3	Регистр управления линией GPIO3 по переполнению FIFO
0x6C000180	16	DSPGPS4	Регистр управления линией GPIO4 по наполнению FIFO
0x6C000190	16	DSPGPE4	Регистр управления линией GPIO4 по переполнению FIFO
0x6C0001A0	16	DSPGPS5	Регистр управления линией GPIO5 по наполнению FIFO
0x6C0001B0	16	DSPGPE5	Регистр управления линией GPIO5 по переполнению FIFO
0x6C0001C0	16	DSPGPS6	Регистр управления линией GPIO6 по наполнению FIFO
0x6C0001D0	16	DSPGPE6	Регистр управления линией GPIO6 по переполнению FIFO
0x6C0001E0	16	DSPGPS7	Регистр управления линией GPIO7 по наполнению FIFO
0x6C0001F0	16	DSPGPE7	Регистр управления линией GPIO7 по переполнению FIFO
0x6C000200	16	DSPGPS8	Регистр управления линией GPIO8 по наполнению FIFO
0x6C000210	16	DSPGPE8	Регистр управления линией GPIO8 по переполнению FIFO
$0 \times 6 C 0 0 0 2 2 0$	16	DSPGPS9	Регистр управления линией GPIO9 по наполнению FIFO
0x6C000230	16	DSPGPE9	Регистр управления линией GPIO9 по наполнению FIFO
0x6C000230	16	DSPGPS10	Регистр управления линией GPIO10 по наполнению FIFO
0x6C000250	16	DSPGPE10	Регистр управления линией СРІО10 по переполнению FIFO
0x6000250	16	DSPGPS11	Регистр управления линией GPIO11 по наролнению FIFO
0x6C000200	16	DSPGPE11	Регистр управления линией GPIO11 по переполнению FIFO
0x60000270	16	DSPGPS12	Регистр управления линией СРІО12 по народнению ЕІЕО
0x0C000200	16	DSPCPE12	
0x6C000230	16	DSPCPS12	Регистр управления линией СРІО12 по переполнению ГПО
0x0C0002A0	10	DSFOF515	Ратистр управления линией ОГОТЗ по наполнению ГПО
0x6C0002B0	10	DSPOPEIS	Регистр управления линией ОРЮТЗ по переполнению ГГО
0x6C0002C0	10	DSPOPS14	Регистр управления линией ОРЮ14 по наполнению ГГО
0x6C0002D0	10	DSPOPE14	Регистр управления линией ОРЮ14 по переполнению ГГО
0x6C0002E0	10	DSPGPS15	Регистр управления линией СРЮТ5 по наполнению FIFO
0x6C0002F0	16	DSPGPEIS	Регистр управления линией GPI015 по переполнению FIFO
0x6C000400	2	DSPCFL0	Регистр нижнего порогового значения канального FIFO0
0x6C000402	2	DSPCFH0	Регистр верхнего порогового значения канального FIFO0
0x6C000404	2	DSPCFC0	Регистр управления канального FIFO0
0x6C000406	2	DSPCFS0	Регистр статуса канального FIFO0
0x6C000408	8		Регистры управления канального FIFO1
0x6C000410	8		Регистры управления канального FIFO2
0x6C000418	8		Регистры управления канального FIFO3
0x6C000420	8		Регистры управления канального FIFO4
0x6C000428	8		Регистры управления канального FIFO5
0x6C000430	8		Регистры управления канального FIFO6
0x6C000438	8		Регистры управления канального FIFO7
0x6C000440	8		Регистры управления канального FIFO8
0x6C000448	8		Регистры управления канального FIFO9



Адрес	Длина	Обозначение	Краткое описание
0x6C000450	8		Регистры управления канального FIFO10
0x6C000458	8		Регистры управления канального FIFO11
0x6C000460	8		Регистры управления канального FIFO12
0x6C000468	8		Регистры управления канального FIFO13
0x6C000470	8		Регистры управления канального FIFO14
0x6C000478	8		Регистры управления канального FIFO15
0x6C000600	8		Регистры управления FIFOA
0x6C000608	8		Регистры управления FIFOB
0x6C000610	8		Регистры управления FIFOC
0x6C000618	8		Регистры управления FIFOD
0x6C000700	2	DDCRESET	Регистр сброса DDC ISL5216
0x6C000708	2	DDCSTART	Регистр запуска каналов DDC ISL5216
0x6C000800	2	DDC0DATAL	Регистр данных DDC0 ISL5216. млалшие 16 разрядов
0x6C000802	2	DDC0DATAH	Регистр ланных DDC0 ISL5216, старшие 16 разрялов
0x6C000808	2	DDC0ADDRW	Регистр адреса DDC0 лля записи
0x6C00080C	2	DDC0ADDRR	Регистр адреса DDC0 для чтения
0x6C000820	32		Регистры управления DDC1
0x6C000840	32		Регистры управления DDC2
0x6C000860	32		Регистры управления DDC3
0x6C001000	2		илентификатор основной платы
0x6C001002	2		Илентификатор DSP0
$0 \times 6 C 0 0 1 0 0 4$	2		Идентификатор DSP1
0x6C001006	2		Илентификатор мезонина АДС
0x6C001008	2		Идентификатор мезонина РДДСО
0x6C00100A	2		Идентификатор мезонина PDDC1
0x6C00100C	2		Идентификатор мезонина ГВБСТ
0x6C00100E	2		Идентификатор субмодуля КГС
0x6C001010	4		Серийный номер устройтсва
$0 \times 6 C 0 0 1 0 1 4$	2		Номер процессора
$0 \times 6 C 0 1 0 1 0 0$	2	SYNCCTRL	Регистр управления направлением линий синуронизации
$0 \times 6 C 0 1 0 1 0 2$	2	SYNCDATA	Регистр управления уповнем линий синхронизации
$0 \times 6 C 0 1 0 1 1 0$	16	SYNCO	Регистр управления полключением линии синхронизации SYNCO
$0 \times 6 C 0 1 0 1 2 0$	16	SYNC1	Регистр управления подключением линии синхронизации SYNC1
$0 \times 6 C 0 1 0 1 3 0$	16	SYNC2	Регистр управления подключением линии синхронизации SYNC2
$0 \times 6 C 0 1 0 1 4 0$	16	SYNC3	Регистр управления подключением линии синхронизации SYNC3
$0 \times 6 C 0 1 0 1 5 0$	16	SYNC4	Регистр управления подключением линии синхронизации SYNC4
0xA0000000	4	DSPCED0	Регистр доступа к данным канального FIFO0
$0 \times A0000010$	4	DSPCFD1	Регистр доступа к данным канального ГП 00
0xA0000020	4	DSPCED2	Регистр доступа к данным канального ГП 01
0xA0000030	4	DSPCFD3	Регистр доступа к данным канального ГІГОЗ
0xA0000040	4	DSPCFD4	Регистр доступа к данным канального ГП 05
0xA0000050	4	DSPCFD5	Регистр доступа к данным канального ГП О Г
0xA0000060	4	DSPCFD6	Регистр доступа к данным канального ГП Об
0xA0000070	4	DSPCFD7	Регистр доступа к данным канального ГП ОО
0xA0000080	4	DSPCFD8	Регистр доступа к данным канального ГП О7
0×A0000090	4	DSPCFD9	Регистр доступа к данным канального ГІГО9
0xA00000A0	4	DSPCFD10	Регистр доступа к данным канального ГІГО10
0x100000010	4	DSPCED11	Регистр доступа к данным канального ГІГО11
0xA0000000	4	DSPCFD12	Регистр доступа к данным канального ГП ОТТ
0xA000000x0	4	DSPCFD12	Регистр доступа к данным канального ГП 012 Регистр доступа к данным канального FIFO13
0xA00000E0	4	DSPCFD14	Регистр доступа к данным канального ГП 013
0×20000010	4	DSPCFD15	Регистр доступа к данным канального ГП ОТЧ
0xA0000180	4	DSPCTEST	Тестовый пегистр
0×B0000000	4		Perform pointing k ground FIEOA
0×B0000000	4		
020000010	4		и огистр доступа к данным ГІГОВ Регистр доступа к данным БІЕОС
	4		понистр доступа к данным глгОС Вористр доступа к доници к ЕЕОО
020000030	4	DSPAPDD	гегистр доступа к данным ГІГОО



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



Приложение II. 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 ПП.1 Варианты основной платы RDMB

Идентификатор	Краткое описание
0x0100	RDMB ревизия 1.0

Таблица ПП.2 Варианты сигнальных процессоров

Идентификатор	Краткое описание
0x0120	TMS320C6416-600 без SDRAM
0x0121	TMS320C6416-720 без SDRAM
0x0122	TMS320C6416-840 без SDRAM
0x0123	TMS320C6416-1000 без SDRAM
0xFFFF	Соответствующий сигнальный процессор не установлен

Таблица ПП.З Варианты АЦП

Идентификатор	Краткое описание
0x0100	4 АЦП ADS5500
0x0101	2 АЦП ADS5500
0x0102	1 АЦП ADS5500
0x0110	4 АЦП AD\$5520
0x0111	2 АЦП AD\$5520
0x0112	1 АЦП ADS5520
0xFFFF	АЦП не установлены

Таблица ПП.4 Варианты DDC

Идентификатор	Краткое описание
0x0121	4 ISL5216
0x0122	2 ISL5216
0x0123	1 ISL5216
0xFFFF	DDC не установлены

Таблица ПП.5 Варианты генераторов частоты дискретизации

Идентификатор	Краткое описание
0x0152	Генератор частоты в диапазоне 80±1 МГц
0x0153	Генератор частоты в диапазоне 93±1 МГц
0x0154	Генератор частоты в диапазоне 40±1 МГц
0x0155	Генератор частоты в диапазоне 56±1 МГц
0x0156	Генератор частоты в диапазоне 105±1 МГц
0x0157	Генератор частоты в диапазоне 112±1 МГц
0x0158	Генератор частоты в диапазоне 125±1 МГц
0xFFFF	Генератор частоты не установлен



Список литературы

- 1. "RDMB. Руководство по эксплуатации", ООО "Резонанс РД", 2006.
- 2. TMS320C6000 Peripherals Reference Guide. Literature Number: SPRU190D, Texas Instruments, February 2001.
- 3. Рихтер Дж. P558 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 LibraryAPI 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 С++", Невский Диалект, СПб., 2001.



Список изменений

Список изменений в версии 1.2

Страница 8

Добавлен раздел 3.3.4, описывающий устройство схемы внешней синхронизации в модулях RDMB ревизии 2.

Страница 33

В пунктах 4.3.8.3 и 4.3.8.4 добавлены константы RDSP64162_OGSYNTH_5MHZ и RDSP64162_OGSYNTH_10MHZ для переключения встроенного генератора частоты дискретизации для синхронизации от внешнего стандарта частоты.

Страница 53

В Приложение I добавлены регистры управления схемой внешней синхронизации в модулях RDMB ревизии 2.

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 распространятся 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 "Pesonance PД" 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.