Проектирование с использованием процессоров Analog Devices. Расширенные возможности симулятора

В предыдущей статье мы создали первый проект в среде VisualDSP++ и получили первые общие представления об отладке проектов при помощи симулятора. Теперь мы рассмотрим более продвинутые возможности симулятора, попутно обсудив некоторые особенности компилятора языка C для процессоров Blackfin. Осваивать эти концепции мы будем на примере простого проекта, осуществляющего ввод сигнала в процессор из внешнего устройства по последовательному порту (SPORT) и вычисление быстрого преобразования Фурье. Поскольку симулятор процессора ADSP-BF527 не поддерживает некоторые из функций, которые планируется продемонстрировать в проекте, в качестве платформы выбран процессор ADSP-BF537. Предлагаемая статья состоит из двух частей. В первой части будет дан обзор аппаратных средств, задействованных в проекте, а вторая часть будет посвящена непосредственно разработке и отладке программного обеспечения.

Александр СОТНИКОВ

Последовательный порт (SPORT)

Процессор Blackfin ADSP-BF537 имеет два синхронных последовательных двунаправленных порта, которые называются SPORT (Serial Port). Эти порты поддерживают различные стандартные протоколы последовательной связи (протокол передачи стереофонических звуковых сигналов I²S, стандарт цифровой телефонии H.100 и др.), а также могут быть сконфигурированы для работы с разнообразными АЦП, ЦАП и аудиокодеками с последовательным интерфейсом или для организации связи между несколькими процессорами в многопроцессорной системе.

Каждый из последовательных портов процессора обладает следующими функциональными возможностями:

- Независимая одновременная работа на прием и передачу.
- Программный выбор длины (от 3 до 32 бит) и формата (начиная со старшего или младшего бита) слова.
- Поддержка аппаратной компрессии данных в соответствии со стандартом G.711 по A-закону или μ-закону.
- Возможность работы с сигналом кадровой синхронизации, выделяющим начало каждого последовательно передаваемого/принимаемого слова.
- Возможность работы от внутреннего источника или от внешних источников сигналов тактовой и кадровой синхронизации.
- Прием/передача данных отдельными словами по прерываниям или блоками в режиме прямого доступа к памяти (DMA, Direct Memory Access).
- Поддержка многоканального режима приема/передачи с временным разделением каналов.

Аппаратный интерфейс каждого последовательного порта состоит из восьми сигналов, которые перечислены в таблице 1 (символ x соответствует номеру порта, 0 или 1). Наличие вторичных линий приема/передачи позволяет повысить пропускную способность за счет подключения к последовательному порту сразу двух устройств, которые при этом будут работать синхронно (поскольку линии тактовой и кадровой синхронизации у них будут общими).

Таблица 1. Сигналы последовательного порта

Линия	Описание
DTxPRI	Основная линия передачи данных
DTxSEC	Вторичная линия передачи данных
TSCLKx	Тактовая синхронизация передачи
TFSx	Кадровая синхронизация передачи
DRxPRI	Основная линия приема данных
DRxSEC	Вторичная линия приема данных
RSCLKx	Тактовая синхронизация приема
RFSx	Кадровая синхронизация приема

Блок-схема последовательного порта процессора Blackfin ADSP-BF537 приведена на рис. 1. Последовательная передача данных инициируется ядром или контроллером DMA путем записи слова, которое необходимо передать из памяти в регистр SPORTx_TX. Записанное слово через FIFO и промежуточный регистр хранения передается во внутренний сдвиговый регистр передачи, из которого оно побитно извлекается по активному фронту сигнала TSCLKx и передается, начиная с младшего или старшего разряда (в зависимости от настроек регистра управления), во внешнее устройство. Пока в буфере FIFO передачи есть место, контроллер последовательного порта генерирует прерывание или запрос DMA для записи в SPORTx_TX нового слова данных.

На приемной стороне последовательный порт побитно принимает данные по активному фронту сигнала RSCLKx в соответствующий сдвиговый регистр. После того как слово принято полностью, оно записывается в буфер FIFO. Когда в буфере FIFO есть хотя бы одно слово, контроллер порта выдает запрос прерывания или DMA, сигнализирующий, соответственно, ядру процессора или контроллеру DMA о том, что данные готовы и должны быть считаны.

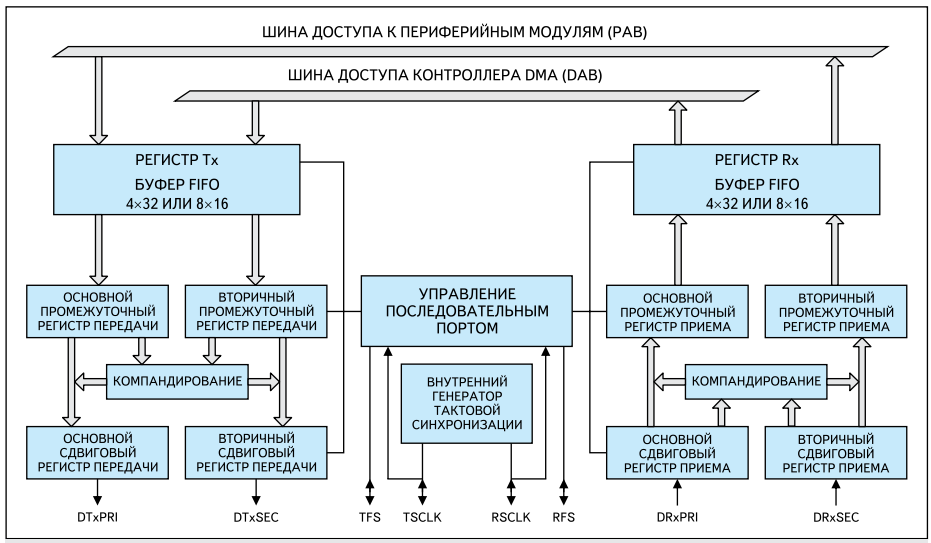

Рис. 1. Блок-схема последовательного порта

Для управления последовательными портами используются следующие регистры:

- Регистры управления передачей (SPORTx_TCR1 и SPORTx_TCR2) и регистры управления приема (SPORTx_RCR1 и SPORTx_RCR2), которые управляют длиной последовательного слова, активацией вторичного набора линий приема/передачи, настройками сигналов кадровой и тактовой синхронизации и т. п.
- Регистры делителей сигналов тактовой синхронизации передачи (SPORTx_TCLKDIV) и приема (SPORTx_RCLKDIV), которые управляют тактовой частотой внутренне генерируемого сигнала тактовой синхронизации.
- Регистры делителей сигналов кадровой синхронизации передачи (SPORTx_TFSDIV) и приема (SPORTx_RFSDIV), которые управляют тактовой частотой внутренне генерируемого сигнала кадровой синхронизации.
- Регистр конфигурации многоканального режима (SPORTx_MCMC1 и SPORTx_MCMC2), которые используются для настройки параметров последовательного порта при работе в режиме многоканального приема/передачи.
- Регистры выбора активных каналов при работе в многоканальном режиме (SPORTx_MRCS1, SPORTx_MRCS2, SPORTx_MRCS3, SPORTx_MRCS4).

Информация о состоянии последовательного порта отображается в регистре SPORTx_STAT.

В нашем примере мы сосредоточимся на приеме данных через последовательный порт. Существует два возможных способа обработки данных, поступающих от внешнего устройства,— обработка каждого отдельного отсчета и обработка блока отсчетов. Первый способ часто используется при фильтрации данных в фильтрах с конечной или бесконечной импульсной характеристикой (КИХ и БИХ). В этом случае на каждый принятый отсчет данных генерируется отдельное прерывание, а прием, в зависимости от источника сигнала, может осуществляться пословно или в режиме DMA. (Второй вариант характерен, например, для стереокодеков, которые могут выдавать два или три слова на один временной отсчет — данные правого и левого каналов, а также информацию состояния.) При использовании второго способа данные обычно вводятся через порт во внутреннюю память в режиме DMA, а процессор для обеспечения непрерывной обработки должен успеть обработать полученный блок данных до заполнения следующего блока. Такой способ характерен, например, для различных алгоритмов спектрального оценивания, и именно он будет использоваться в рассматриваемом проекте.

Прямой доступ к памяти (DMA)

Режим прямого доступа к памяти (DMA) — это эффективный метод обмена данными между внутренней памятью и периферийными устройствами, который происходит под управлением специального контроллера без вмешательства ядра процессора. Контроллер DMA самостоятельно осуществляет формирование адресов и обращение к памяти и сигнализирует о завершении приема/передачи блока данных установкой битов в регистре состояния канала и/или формированием прерывания. Ядро процессора отвечает только за корректную инициализацию канала обмена данными (канала DMA) и своевременную подготовку (при работе канала DMA на передачу) или обработку (при работе канала DMA на прием) буфера данных.

Контроллер DMA процессора ADSP-BF537 поддерживает 12 каналов DMA для обмена данными с интегрированными периферийными модулями (Ethernet MAC, SPORT, UART, PPI и SPI), два канала MemoryDMA (MDMA), предназначенных преимущественно для пересылок между внутренней и внешней памятью процессора, и два канала Handshake MemoryDMA (MDMA с квитированием) для обмена данными с внешними микросхемами периферийных модулей. Каждый из каналов может обеспечивать пересылки линейных (1-D DMA) или двумерных (2-D DMA) буферов.

Существует две разновидности режима DMA, поддерживаемых процессорами ADSP-BF537: с непосредственной инициализацией регистров и с использованием дескрипторов. К первому типу относятся однократные пересылки, при которых работа канала останавливается по завершении пересылки одиночного линейного буфера, и пересылки в режиме с автоматической буферизацией. В режиме с автоматической буферизацией после инициализации регистров канала DMA контроллер осуществляет чтение/запись буфера в памяти по кругу, генерируя прерывания по считыванию/заполнению всего буфера или его части. Для настройки канала DMA в режиме с непосредственной инициализацией регистров в программе необходимо выполнить следующие действия (для 1-D DMA):

- Запись адреса буфера, являющегося источником или приемником данных, в регистр DMAx_START_ADDR (x — номер канала DMA).
- Запись количества пересылаемых слов в регистр DMAx_X_COUNT.
- Запись инкремента адреса, прибавляемого к текущему адресу после пересылки отдельного элемента данных, в регистр DMAx_X_MODIFY. Обычно инкремент устанавливается равным 1, 2 или 4 для байтовых, 16-битных и 32-битных пересылок соответственно.
- Запись настроек рабочего режима канала в регистр DMAx_CONFIG (в поле FLOW регистра должно быть указание значение 0 или 1 для настройки однократной пересылки или пересылки с автоматической буферизацией соответственно).
- Установка бита DMAEN в регистре DMAx_CONFIG для запуска пересылки.

При организации пересылок в режиме 2D-DMA для описания двумерных массивов совместно с регистрами DMAx_X_COUNT и DMAx_X_MODIFY используются регистры DMAx_Y_COUNT и DMAx_Y_MODIFY.

В режиме DMA с использованием дескрипторов контроллер DMA получает настройки для канала из структуры, которая хранится в памяти процессора и называется дескриптором. Для запуска пересылок DMA в дескрипторном режиме необходимо записать 32-битное значение адреса первого дескриптора в регистр DMAx_NEXT_DESC_PTR (или DMAx_CURR_DESC_PTR в одном из вариантов режима) и загрузить в регистр DMAx_CONFIG слово со значением поля FLOW, равным 0x4, 0x6 или 0x7, и единичным значением бита DMAEN. Эти действия вызывают загрузку контроллером DMA дескриптора из ячейки памяти, на которую

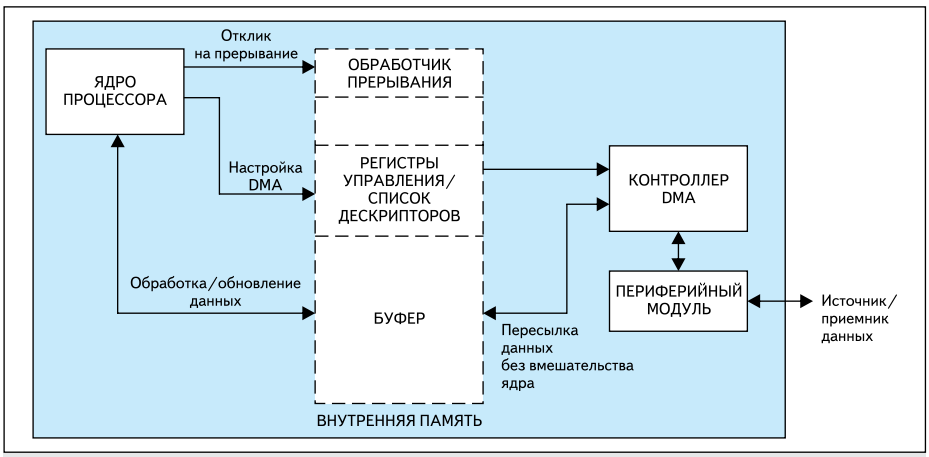

Рис. 2. Обмен данными с периферийным модулем в режиме DMA

указывает содержимое регистра DMAx_NEXT_DESC_PTR. Если после перезаписи регистра DMAx_CONFIG содержимым из дескриптора бит DMAEN сохраняет единичное значение, то контроллер DMA начинает пересылку данных. Содержимое регистра DMAx_CURR_DESC_PTR указывает на дескриптор, который будет загружен после пересылки текущего блока данных. Используя дескрипторы, разработчик может настраивать самые разнообразные последовательности пересылок DMA.

Обобщенная схема организации обмена данными между внутренней памятью и периферийным модулем процессора в режиме DMA изображена на рис. 2.

В рассматриваемом примере мы ограничимся самым простым вариантом DMA — с однократной пересылкой линейного массива данных и прерыванием по заполнению буфера в памяти.

Система прерываний процессоров Blackfin

Прерывание — это сигнал, который оповещает ядро процессора о наступлении того или иного события, которое требует приостановки выполнения текущей последовательности команд и внеочередного обслуживания. В процессоре Blackfin можно выделить пять основных типов таких событий:

- Вход в режим эмуляции для управления процессором через интерфейс JTAG.
- Подача сигнала сброса (/RESET).
- Немаскируемое прерывание, которое вызывается модулем сторожевого таймера или подачей на процессор внешнего сигнала/NMI и часто используется в качестве индикатора отключения питания системы.
- Исключения — синхронные по отношению к процессу выполнения команд прерывания, как правило, вызываемые различными некорректными действиями программы (некорректная адресация, попытка выполнения неопределенных команд и т. п.).
- Асинхронные прерывания, генерируемые различными периферийными модулями процессора или специальной командой ассемблера RAISE.

За управление прерываниями в процессоре отвечает модуль, который называется контроллером событий и состоит из двух блоков, контроллера событий ядра (CEC, Core Event Controller) и контроллера прерываний системы (SIC, System Interrupt Controller), образующих двухуровневую структуру.

Контроллер событий ядра работает на тактовой частоте ядра (CCLK) и выдает сигналы прерываний непосредственно на ядро процессора. Сигналы CEC имеют разные приоритеты, и каждому из них соответствует отдельный элемент таблицы векторов прерываний (EVT, Event Vector Table). События, поддерживаемые CEC, перечислены в порядке убывания приоритета в таблице 2 (из прерываний общего назначения IVG7 имеет высший приоритет, IVG15 — низший). Как правило, одно или два прерывания общего назначения с наименьшим приоритетом (IVG15 и/или IVG14) резервируются под программные прерывания, а семь или восемь оставшихся прерываний используются для обслуживания прерываний периферийных модулей.

Таблица 2. События ядра

Источник	Название прерывания
Эмуляция (высший приоритет)	EMU
Сброс процессора	RST
NMI	NMI
Исключение	EVX
Аппаратная ошибка	IVHW
Таймер ядра	IVTMR
Прерывания общего назначения	IVG7–IVG15

Контроллер прерываний системы работает на тактовой частоте системы (SCLK) и определяет соответствие между многочисленными одноранговыми сигналами запроса прерывания от периферийных модулей процессора (всего 32 возможных сигнала) и девятью входами прерываний общего назначения CEC. При помощи регистра маскирования системных прерываний (SIC_IMASK) и четырех регистров распределения системных прерываний (SIC_IARx) контроллера SIC разработчик разрешает/запрещает обработку сигналов отдельных прерываний и группирует их в соответствии с желаемыми приоритетами. Поскольку в одно прерывание общего назначения CEC может отображаться сразу несколько прерываний от периферийных модулей, для определения конкретного источника прерывания может использоваться регистр состояния прерываний SIC (SIC_ISR).

Программная инициализация прерываний включает в себя следующие шаги:

- Инициализация таблицы векторов прерываний (запись в соответствующие элементы EVT адресов соответствующих подпрограмм обслуживания прерываний).
- Инициализация регистра IMASK, управляющего маскированием прерываний ядра.
- Назначение соответствий между прерываниями от периферийных модулей и входами прерываний общего назначения CEC в регистрах SIC_IARx. (Этот шаг необходим, только если разработчика программы по той или иной причине не устраивает назначение прерываний, устанавливаемое по умолчанию после сброса.)
- Отмена маскирования отдельных прерываний от периферийных модулей в регистре SIC_IMASK.

В подпрограмме обслуживания прерывания периферийного модуля, в общем случае, необходимо определить источник прерывания, обеспечить необходимую обработку, а также позаботиться о снятии запроса прерывания периферийного модуля, чтобы прерывание не было обслужено повторно (биты регистра SIC_ISR не сбрасываются до тех пор, пока не будет сброшен механизм, вызвавший прерывание).

В рассматриваемом проекте будет задействовано прерывание канала 3 DMA, который отвечает за прием данных через SPORT0, по умолчанию отображаемое в прерывание IVG9 ядра.

После того как мы вкратце рассмотрели аппаратные средства, используемые в проекте, можно перейти к написанию кода приложения, речь о котором пойдет в следующем номере.