

ПРОГРАММИРОВАНИЕ СИГНАЛЬНЫХ ПРОЦЕССОРОВ СЕРИИ BLACKFIN ДЛЯ ЗАДАЧ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Лосев Г.И. Email: Losev1135@scientifictext.ru

Лосев Герман Игоревич - инженер, студент,
кафедра систем автоматического управления и контроля,
Национальный исследовательский университет
Московский институт электронной техники, г. Москва

Аннотация: в статье рассматриваются принципы программирования цифровых сигнальных процессоров серии BlackFin в среде VisualDSP++, необходимые для устройств цифровой обработки сигналов. На примере блоков PPL, UART и SPORT процессора описываются основы инициализации ядра процессора и компонентов, необходимых для приёма и передачи данных с АЦП на ПК. Статья также описывает процедуру приёма данных через Sport интерфейс, передачу через интерфейс UART и загрузку программы на процессор через UART. Кроме того, описывается функция цифровой фильтрации и работа с ней.

Ключевые слова: микропроцессоры, цифровая обработка, инициализация, прием/передача.

PROGRAMMING OF SIGNAL PROCESSORS OF THE BLACKFIN SERIES FOR DIGITAL SIGNAL PROCESSING TASKS

Losev G.I.

Losev German Igorevich - Engineer, Student,
DEPARTMENT OF AUTOMATIC CONTROL AND MONITORING SYSTEMS,
NATIONAL RESEARCH UNIVERSITY OF ELECTRONIC TECHNOLOGY, MOSCOW

Abstract: the article discusses the principles of programming of the BlackFin series signal processors in the VisualDSP ++ environment, which are necessary for digital signal processing devices. The examples of the PPL, UART and SPORT units describe the fundamentals of initialization of the processor core and the components necessary to receive and transfer data from the ADC to the PC. The article also describes the procedure for receiving data via the Sport interface, transferring through the UART interface, and downloading the program to the processor via the UART. In addition, describes the function of digital filtering and working with it.

Keywords: microprocessors, digital processing, initialization, reception / transmission.

УДК 004.31

Цифровая обработка сигнала (ЦОС или DSP – digital signal processing) является одной из самых новых и эффективных технологий, которая активно внедряется в широком диапазоне областей науки и техники: коммуникации, метеорологии, радиолокации и гидролокации, медицинской визуализации изображения, цифровом аудио и видео вещание, разведке нефти и газа и т.д. Можно сказать, что происходит глубокое проникновение цифровых технологий обработки сигналов во всех сферах человеческой деятельности. Сегодня технология DSP является одной из областей знаний, которые необходимы для ученых и инженеров всех отраслей без исключений.

Одним из основных инструментов цифровой обработки сигналов являются специализированные процессоры, которые могут производить такие сложные операции, как цифровая фильтрация или быстрые преобразования Фурье (БПФ), за минимально возможно время, при малых габаритах и цене микросхем. Одним из лидеров в этой области можно назвать процессоры серии Blackfin от Analog Device, чья производительность может достигать 1200 MMACS. Однако на своём собственном опыте я обнаружил, что система команд этих процессоров намного сложнее, чем, к примеру, у процессоров Atmel, которые для моих задач не подходили. Информации об их программировании на русском языке не очень много, а на иностранном в основном описывают программирование отдельных функций и компонентов процессора. Поэтому в этой статье я попробую просуммировать свой опыт работы с этими процессорами, который я использовал для написания программы приёма данных с АЦП через порт Sport, фильтрации этого сигнала с помощью КИХ фильтра, и передачи данных на персональный компьютер через интерфейс UART.

В первую очередь необходимо правильно настроить работу основных систем процессора, а именно: тактовую частоту процессора, работу блоков памяти SDRAM, модуль шины внешней шины и др. Настройку этих устройств процессора необходимо сделать до того, как будет загружена основная программа в процессор, для чего необходимо использовать код инициализации (init code). Когда загрузчик программ увидит указатель на код инициализации, то он сначала загрузит этот код в память

процессора и выполнит ее, а затем вернётся к исполнению кода загрузки. Программная среда VisualDSP++ поставляется с примерами кода загрузки для соответствующих отладочных плат. Однако, параметры работы отладочных плат могут не соответствовать параметрам требуемого устройства. Поэтому необходимо написать код инициализации для требуемого устройства. Сначала настраивается модуль автоматической фазовой настройки частоты (PLL). Этот модуль принимает на вход тактовую частоту с кварцевого резонатора и затем, с помощью делителей частоты, может увеличивать и уменьшать ее по необходимости. Блок способен генерировать тактовую частоту как самого ядра процессора, так и отдельных его систем. Так же этот блок взаимодействует с блоком динамического контроля питания (DPMC). Кроме того, настройкой этого модуля можно включить один из нескольких режимов работы процессора, характеризующиеся разной скоростью работы и потреблением питания. Настройка этого модуля осуществляется через регистры PLL_CTL, PLL_DIV, PLL_LOCKCNT и PLL_STAT. В регистре PLL_CTL устанавливается значения делителя обратной связи MSEL, с чьей помощью устанавливается тактовая частота контроллера напряжения VCO по формуле:

$$VCO = CLKIN \cdot MSEL. \quad (1)$$

Так же этот регистр используется для установки задержек в блоке EBUI, включения режима PLL и т.д. Регистр PLL_DIV устанавливает значение делителей CSEL и SSEL для генератора частоты ядра процессора и систем процессора. Регистром PLL_LOCKCNT задается задержка перед началом работы систем процессора после запуска основного ядра. Регистр PLL_STAT устанавливает текущий режим работы блока PLL.

Процессоры семейства Blackfin обладают большим количеством портов ввода/вывода общего назначения. Эти порты могут использоваться либо как стандартные порты ввода/вывода, либо как порты для работы с периферийными устройствами по определенным интерфейсам. Некоторые порты процессора также могут работать в нескольких режимах работы с периферийными устройствами. Поэтому, в первую очередь, необходимо активировать необходимые порты процессора в режим работы с периферийными устройствами и настроить их в необходимый режим работы. Настройка портов ввода/вывода общего назначения производится регистрами памяти PORTx_MUX и PORTx_FER где x – символ набора портов. Затем последовательно происходят инициализация и настройка основных функций и интерфейсов, которые были необходимы для работы устройства. Сначала происходит инициализация интерфейса UART, используемого для передачи массивов данных на персональный компьютер. Что бы начать работу с UART сначала его надо активировать. Для этого в регистр UART_GTCL необходимо записать единицу в бит UCEN (UART controller enabled или UART контроллер запущен). В отличие от предыдущих версий процессоров фирмы Analog Device и программного обеспечения, поставляемого с ними, среда программирования VisualDSP++ 5.0 имеет большое количество библиотек, облегчающих работы с конкретным процессором. Для каждого процессора написана библиотека, в которой уже заранее объявлены регистры всех основных модулей процессора и объявлены переменные, с помощью которых настраиваются функции этих регистров или считывается их состояние. В частности, в данном случае, для того чтобы активировать порт, достаточно указать на регистр pUART0_GTCL присвоить значение UCEN. Затем необходимо настроить скорость работы UART порта. Скорость порта задаётся 16-битным делителем, который делит тактовую частоту на выводе SCLK, которая может задаваться либо от внешнего генератора частоты, либо от внутреннего [5]. Формула, по которой рассчитывается скорость работы

$$\text{Скорость следования битов} = SCLK / (16 \cdot \text{делитель}). \quad (2)$$

Из этой формулы рассчитывается необходимый делитель. Затем необходимо активировать делитель в модуле UART записью значения DLAB в регистр UART_LCR (регистр контроля линии). Затем значения выбранного делителя записывается в два 8-битных регистра UART_DLL и UART_DLH. В регистр DLH делитель записывается со сдвигом на 8 бит. После настройки остается только настроить количество стартовых стоповых битов, количество битов данных и способ проверки четности (если требуется). Это также производится записью необходимых значений в регистр UART_LCR. Для начала передачи данных необходимо записать символ в регистр передачи UART_THR. Для правильной работы необходимо отправлять в процессор по одному символу за раз. Для того, чтобы предотвратить потерю данных или какие-либо искажения регистр статуса линии LSR предоставляет сигнальный флаг THRE. Флаг THRE устанавливается, когда модуль UART готов к передаче новых данных и не занят другими операциями, и сбрасывается, когда процессор загружает данные в регистр THR. Если записать данные в регистр, когда UART занят, то предыдущие данные будут перезаписаны и не будут отправлены. Кроме передачи данных, UART порт используется для загрузки программы управления на процессор устройства. В режиме загрузки в режиме “UART slave” устройства, процессор Blackfin получает загруженные данные с

внешнего хост устройства, подключенного через UART. Во время этой операции необходимо учитывать несколько критических моментов. В первую очередь необходимо запустить процедуру автоматического вычисления скорости потока кадров, которая установит скорость кадров интерфейса UART в соответствии с генератор тактовой частоты систем процессора. В отличие от других режимов загрузки через хост устройство (таких, как SPI или TWI), приёмному устройству необходимо правильно установить частоту кадров UART. После того, как символ @ (0x40) был загружен на вход UA0_RX, загрузочное ядро процессора распознает настройки передачи и хост начинает передачу загрузочного потока. Распознавание происходит за счет четырех переданных байтов: 0xBF, UARTx_DLL, UARTx_DLH, 0x00. Хост устройство не должно выполнять никаких других действий, пока не примет все эти байты для распознавания устройства. После получения байта, говорящего о том, что были приняты все данные для настройки, хост может начать загрузку процессора. Хосту необходимо знать точную длину в байтах отправляемой загрузочной программы, но при этом нет необходимости знать о составе отправляемой информации. Если вовремя загрузки изменяется режим работы PLL, необходимо это учитывать при программировании, иначе процедура загрузки даст сбой. Контролеру UART требуется, чтобы точное значение скорости кадров было установлено в регистрах UARTx_DLL и UARTx_DLH для корректного прерывания входящих данных. Если частота меняется, то необходимо изменять и частоту делителей частоты передачи данных. Для выполнения этого контроля можно использовать две поставляемые с VisualDSP++ функции: `u32 uart1_get_bitrate(void)` и `void uart1_set_bitrate(u32)`. Обе эти функции необходимо выполнить перед и после процедуры изменения PLL соответственно. Они вызываются автоматически, если был установлен режим загрузки BMODE=0111. Первая функция сохраняет текущую скорость UART, высчитывает новые значения делителей тактовой частоты, которое соответствует новым требованиям, и затем вторая функция сохраняет эти значения в регистрах делителя UARTx_DLL и UARTx_DLH. Функция `void uart1_set_bitrate(u32)` обеспечивает обратную связь с хост устройством [4].

Затем идет инициализация SPORT порта, по которому процессор принимает данные после обработки их аналого-цифровым преобразователем. Последовательный порт может принимать данные на скорости 1/2 от системной скорости (SCLK). Скорость этого порта должна быть обязательно меньше, чем системная скорость SCLK. Для этого присутствуют специальные счетчики для гибкой настройки, как приема данных, так и передачи. Настройка последовательного порта осуществляет также запись определенных значений в регистры управления. Настройка данного порта должна производиться до запуска работы этого порта. После его запуска процессор запрещает запись новых значений в регистр контроля. Поэтому сначала нужно убедиться, что порт выключен, записав 0 в SPORTx_RCR1, где x-номер порта.

Частота работы задается, так же, как и в случае с UART, делителями RCLKDIV и TCLKDIV, которые можно найти по формулам 3 и 4:

$$\text{Частота передачи} = SCLK / (2 \cdot (TCLKDIV + 1)) \quad (3)$$

$$\text{Частота приёма} = SCLK / (2 \cdot (RCLKDIV + 1)) \quad (4)$$

Также необходимо задать число циклов счетчика, которые необходимо отсчитать до генерации сигнала синхронизации кадров приёма или передачи (RFS и TFS соответственно). Частота следования этих сигналов также задается делителем и рассчитывается по формулам 3 и 4. Таким образом, синхронизация кадров будет активна. Тем не менее, значение RFSDIV и TFSDIV не должно быть меньше, чем длина слова данных +1. Значение меньше этого может повлечь к остановке работы внешнего устройства или к другим непредсказуемым результатам. Длина слова настраивается занесением в регистр SPORTx_RCR2 в поле SLEN значения необходимой длины слова+1. Также необходимо настроить последовательность битов (LSB или MSB) и тип данных, используемый в процессе приема/передачи. Это производится записью соответствующих значений в определенные поля регистра SPORTx_RCR1. Принятые данные и данные, которые необходимо передать, можно перемещать между SPORT и внутренней памятью одним из двух способов: пересылая по одному слову данных за раз или посредством передачи DMA блока. Если ни один канал DMA модуля SPORT не активен, то последовательный порт генерирует прерывание каждый раз, когда ему требуется передать слово данных или, когда ему нужно принять слово данных. Контроллер DMA предоставляет механизм, с помощью которого можно принимать или передавать целый блок данных или несколько блоков одновременно до того, как было сгенерировано прерывание. Контроллер DMA позволяет процессору продолжать свою работу пока не будет принят или передан полный блок данных. После этого обработчик прерываний может работать с целым блоком данных [4].

Как было описано выше, в независимости от выбранного способа передачи данных между памятью и SPORT модулем, процедура передачи и приема в этом интерфейсе происходит по прерываниям, поэтому их так же необходимо инициализировать. Стандартный алгоритм инициализации прерываний выглядит

следующим образом:

- Инициализация адреса вектора в таблице векторов событий (EVT)
- Инициализация регистра IMASK
- Размаскирование конкретного прерывания для необходимого периферийного устройства в регистре SIC_IMASK

Для того, чтобы запрограммировать прерывание, достаточно его описать подобно написанию функций в языках программирования. Разница в том, что прерывание будет вызываться по некоторому событию в нутрии процессора, а функция вызывается из самой программы.

Для проведения цифровой фильтрации сигнала, получаемого с АЦП используется функция `iir_fr16(input[],output[],length,iir_state_fr16 *filter_state)`, которая воспроизводит КИХ фильтр Чебышева второго порядка. Эта функция генерирует отфильтрованный сигнал из входного вектора `input[]` и записывает результат в выходной вектор `output[]`. Количество входных отсчётов и длина выходного вектора задаются аргументом `length`. Функция хранит своё состояние в структурной переменной `filter_state`, которая должна быть объявлена и инициализирована перед вызовом функции. Для этого используется макрос `iir_init(state, coeffs, delay, stages)`. Характеристики фильтра зависят от коэффициентов и количества биквадратных окон фильтра (`stages`). Каждая такая биквадратная секция имеет пять коэффициентов, которые должны храниться в порядке `A2, A1, B2, B1` и `B0`, где `A2` и `A1` коэффициенты знаменателя передаточной функции фильтра, `B2, B1` и `B0` – коэффициенты числителя. Значение коэффициента `A0` является единицей, и значения коэффициентов `A1` и `A2` должны быть масштабированы соответственно. Для этого значение коэффициентов `A0` должно быть больше коэффициентов `A1` и `A2` для всех секций. Коэффициенты `B` должны быть масштабированы в дробный диапазон $[-1...1]$. Коэффициент масштабирования должен быть степенью числа 2. Также каждый фильтр должен иметь свою линию задержки, чья длина должна быть равна удвоенному количеству секций фильтра. Вектор задержки должен быть изначально очищен и не подвергаться дальнейшей модификации [5].

Список литературы / References

1. ADSP-BF50x Blackfin ® Processor Hardware Reference Revision 1.2, February 2013. [Электронный ресурс]. Режим доступа: http://www.analog.com/media/en/dsp-documentation/processor-manuals/ADSP-BF50x_hwr_rev1.2.pdf/ (дата обращения: 04.05.2017).
2. Сотников А. Проектирование с использование процессоров Analog Devices. Цифровой КИХ-фильтр // Компоненты и технологии, 2010. № 10.
3. Вальна О.Д. Разработка устройств на основе цифровых сигнальных процессоров фирмы Analog Devices с использованием Visual DSP++. -М.: Горячая линия-Телеком, 2007. 270 с.