

Программирование сигнальных процессоров компании Analog Devices в среде VisualDSP++

В настоящее время наблюдается непрерывный рост интереса специалистов-разработчиков к использованию в различных устройствах обработки сигналов цифровых сигнальных процессоров (DSP) производства компании Analog Devices. Это обусловлено как удобствами применения, доступностью и широкими возможностями самих процессоров, так и документации на русском языке, изданной к настоящему времени. К этим двум преимуществам можно добавить то, что в сейчас свободно доступны тестовые версии средств разработки для этих процессоров и различные отладочные комплекты, а также есть возможность пройти дополнительную подготовку по программированию DSP в учебных центрах при университетах.

**Ю.С. Юрченко
М.Н. Петров
В.П. Шатилов**

vadim@eltech.spb.ru

Эта статья является первой из цикла статей, посвященных работе с цифровыми сигнальными процессорами компании Analog Devices. Материалом для этих статей послужил более чем годовой опыт проведения коммерческих тренингов для специалистов и разработчиков в центре цифровых сигнальных процессоров (ЦСП) при Санкт-Петербургском государственном электротехническом университете (СПбГЭТУ).

В статьях рассматриваются основы работы в среде VisualDSP++ (создание проекта, компиляция, отладка, графическое отображение данных), а также приложение цифровой обработки сигналов (проектирование цифровых фильтров на базе DSP, реализация БПФ, генерация сигналов и пр.).

Нить тестирования платы с помощью специального аппаратно-программного модуля — эмулятора. Этот модуль управляет цифровым сигнальным процессором через JTAG-интерфейс и позволяет подробно отследить выполнение программного кода. Еще одной важной особенностью, на которую следует обратить внимание, является то, что интерфейс пакета VisualDSP++ единый для всех серий сигнальных процессоров, включая 16-, 32-битные, а также TigerSHARC и BlackFin. Отличия касаются лишь набора установленных компиляторов и имеющейся лицензии.

В первой части нашей публикации будет рассматриваться работа с VisualDSP++ в режиме моделирования.

Начало работы с VisualDSP++

Процесс отладки проекта после написания кода программы и успешной компиляции включает три основных этапа: моделирование (Simulation), оценка (Evaluation) и эмуляция (Emulation). На этапе моделирования работает моделирующая программа (симулятор), которая имитирует работу процессора. Симулятор используется для проверки и отладки программного кода до того, как будет изготовлена плата с процессором. На втором этапе используется оценочная плата EZ-KIT для того, чтобы определить, какой процессор наилучшим образом подходит для решения вашей задачи. Плата подключается к компьютеру с помощью кабеля через параллельный, последовательный или USB-порт. К настоящему времени оценочные платы существуют для всех типов процессоров, начиная с ADSP-2181 и заканчивая новейшим процессором BlackFin. На третьем этапе, когда устройство уже изготовлено, можно выпол-

Установка VisualDSP++

Компания Analog Devices свободно предлагает тестовую 30-дневную версию VisualDSP++ на своем сервере в Интернет по адресу <http://www.analog.com/dsp>. Кроме самого пакета, необходимо скачать инсталлятор лицензий и пройти регистрацию на сервере для получения тестовой 30-дневной лицензии. Кроме Интернет, тестовую версию VisualDSP++ можно получить в представительстве компании Analog Devices или в офисе ее официального дистрибьютора, фирмы ЭЛТЕХ. На сервере Analog Devices доступно большое количество различных примеров, документации, заметок по применению. Использование этого объема информации существенно облегчает процесс освоения как самого VisualDSP++ так и сигнальных процессоров компании Analog Devices.

Процесс установки VisualDSP++ проходит в три этапа: первоначально устанавливается сам пакет, для его установки необходимо иметь не менее 100 Мб

свободного дискового пространства, затем, после установки пакета, устанавливается и запускается инсталлятор лицензий, который запрашивает лицензию, представляющую собой строку, состоящую из 3 букв и 15 цифр. После установки лицензии устанавливается дополнительное программное обеспечение, файлы примеров, документация и драйверы отладочных плат. На этом процесс установки завершается и можно приступать к работе. Системные требования скромные: процессор Pentium 166 и выше, от 32 Мб памяти, Windows 98 и выше.

Запуск VisualDSP++ и создание проекта

После установки доступ к VisualDSP осуществляется из стартового меню Windows. Щелкните мышкой на кнопке «Пуск», затем выберите пункт «Программы», «VisualDSP», «VisualDSP++ for 21xx». При первом запуске на экране монитора появится окно New session, в котором необходимо задать семейство процессоров, платформу и тип процессора, с которым будет выполняться работа (рис. 1). Выберите в окне Debug target пункт ADSP-218x Family Simulator, а в окне Platform — режим Simulation, тип процессора ADSP-2181.

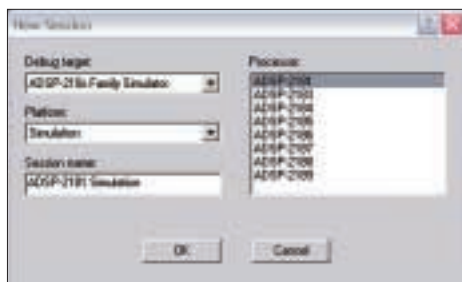


Рис. 1. Выбор сессии

После нажатия кнопки «OK» появится окно VisualDSP++ (рис. 2). При первом запуске вид этого окна будет несколько отличаться от изображенного на рисунке. Основные окна — окно проекта, в котором размещаются содержащиеся в проекте файлы, окно дизассембли-

рованного кода; окно редактирования, окно выходной информации, содержащее стандартные текстовые сообщения ввода-вывода, сообщения об ошибках и пр.

На рис. 2 представлен общий вид среды разработки в процессе отладки программы в симуляторе. Внешний вид среды разработки полностью настраивается по вкусу пользователя, позволяя отображать именно ту информацию, которая на данный момент необходима.

Процесс работы с пакетом начинается с открытия существующего или создания нового проекта. Для этого надо войти в меню Project → New и в открывшемся диалоговом окне ввести название проекта, затем определить, где он будет храниться на жестком диске. После нажатия кнопки «Сохранить» появится новое окно (рис. 3). Рассмотрим его подробнее. Окно содержит несколько закладок, в которых задаются параметры построения системы.

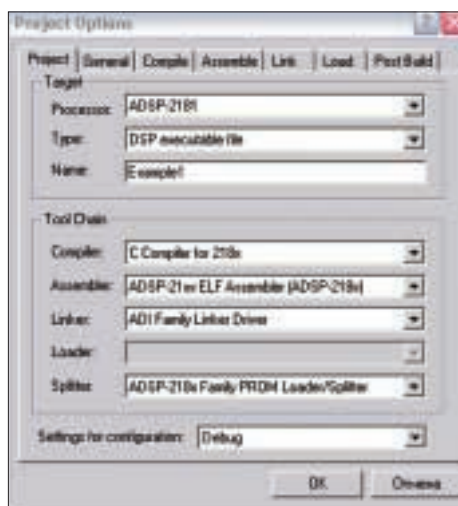


Рис. 3. Диалоговое окно задания свойств проекта

Выберите закладку «Project». Установите параметры, руководствуясь таблицей 1. Согласно этим установкам исполняемый файл генерируется для процессора ADSP-2181, отладчик может использоваться для отслеживания процесса выполнения программы.

Таблица 1

Поле	Значение
Processor	ADSP-2181
Type	DSP executable file
Name	Example1
Setting for configuration	Debug

На закладке «Assemble» установите флажок «Generate debug information». Это заставляет компилятор формировать дополнительную отладочную информацию необходимую для работы симулятора.

Задав всю необходимую информацию, нажмите кнопку «OK». Теперь вы готовы к работе с программой. Необходимо заметить, что, как правило, на этом этапе определяются и устанавливаются такие параметры, как необходимость генерации отладочной информации, карты распределения памяти, формат выходных файлов, возможность оптимизации при компиляции кода, пути доступа к собственным библиотекам. Однако на начальном этапе можно обойтись стандартными настройками, прописанными в открываемом или вновь создаваемом проекте.

Работа с симулятором в режиме моделирования

При работе в этом режиме не требуется наличие сигнального процессора, поскольку VisualDSP++ имитирует его работу. При моделировании вы можете свободно управлять прерываниями, организовывать работу устройства ввода-вывода и т. п. Рассмотрим для примера несколько стандартных программ. Одной из наиболее простых и понятных является программа сложения двух векторов, находящаяся в каталоге C:\Program Files\Analog Devices\VisualDSP\218x\Examples\Example1. После загрузки проекта, на экране можно отобразить исходный текст программы, как показано на рис. 4.

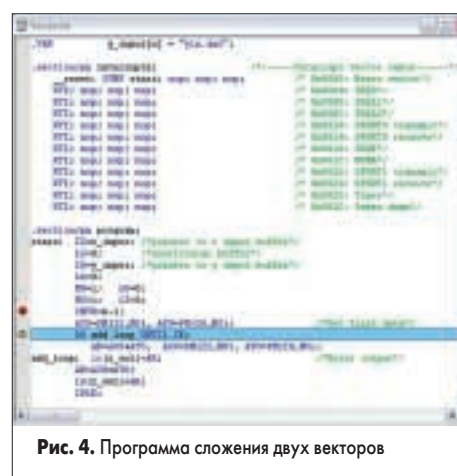


Рис. 4. Программа сложения двух векторов

В этом примере очень наглядно по шагам можно посмотреть, как производится начальная инициализация входных массивов, настройка генераторов адреса, а затем — выборка чисел из памяти и их сложение в цикле. Результаты сложения пересылаются в область ввода-вывода с адресом `z_out = 0x100`. Первым заданием, как правило, является задача изменить программу таким образом, чтобы

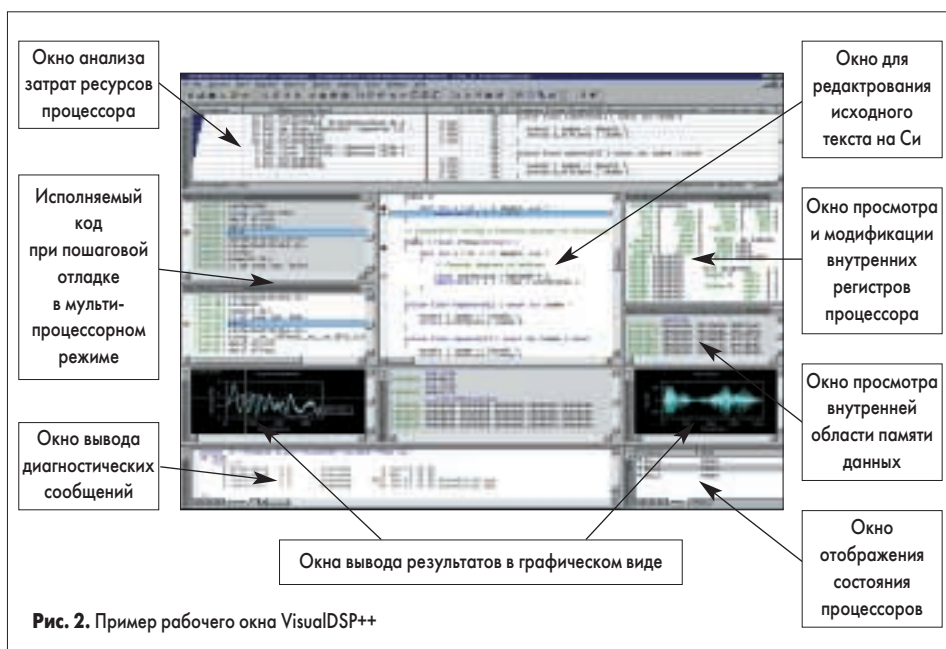


Рис. 2. Пример рабочего окна VisualDSP++

результаты сложения сохранялись в выходном массиве.

Для того чтобы посмотреть распределение исходных массивов данных в памяти и содержимое регистров в процессе пошагового выполнения программы, необходимо, выбрав пункт меню «Memory -> Data», вызвать окно отображения содержимого области памяти данных. Затем подобную процедуру необходимо выполнить для отображения памяти программ, при этом необходимо выбрать начальный адрес отображаемого сегмента памяти. Содержимое регистров в процессе работы можно посмотреть, используя пункт меню «Register», выбирая необходимые регистры из появившегося списка. На рис. 5 показано, как должно выглядеть окно отображения.

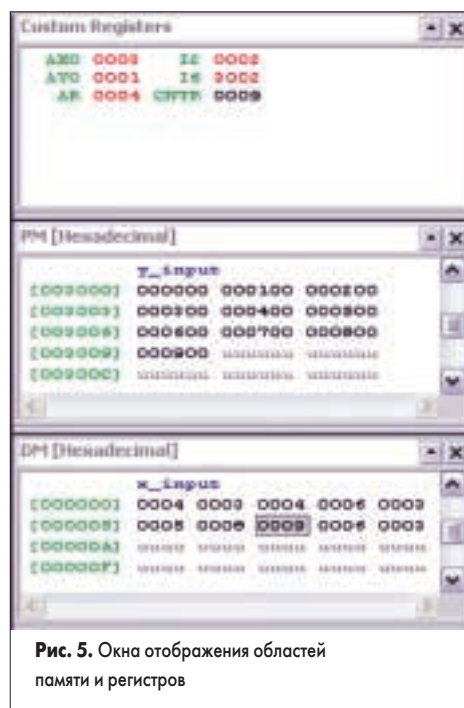


Рис. 5. Окно отображения областей памяти и регистров

Следующим примером, на котором хотелось бы остановить внимание, является пример программы для работы с последовательным портом, предназначенный для освоения работы с симулятором и потоками ввода-вывода.

В этой программе выполняется передача данных через последовательный порт (SPORT1). Процессор сначала получает данные через последовательный порт, а затем передает их через тот же порт. В программе организован бесконечный цикл, то есть процессор находится в состоянии ожидания до тех пор, пока не сгенерируется прерывание приема через SPORT1. Вся обработка данных выполняется подпрограммой обработки прерывания (в данной программе — передача данных через приемный регистр SPORT1 в передающий регистр SPORT1). Входные и выходные потоки данных моделируются с помощью файлов: для входного используется файл «serin.dat», для выходного — «out.dat».

Добавление файлов в проект

Вы можете открыть уже готовый и настроенный проект, находящийся в каталоге

C:\Program Files\Analog Devices\VisualDSP\218x\Examples\Example2 или создать и настроить свой собственный, а затем добавить туда необходимые файлы. Для этого находим файлы Sport.asm, Adsp-2189.ldf, serin.dat и копируем их в папку, в которой вы создали и сохранили свой проект. С помощью кнопки «Add File» или через меню «Project -> Add to Project -> File(s)» добавляем в проект файлы. Скомпилируем проект, нажав клавишу «F7» или используя меню «Project -> Build Project». Если компиляция прошла успешно, в каталоге Debug должен находиться исполняемый файл с расширением *.dxe, который автоматически загружается в отладчик-симулятор.

Настройка потоков ввода-вывода

Перед запуском этой программы необходимо настроить потоки ввода-вывода. Нужно отметить, что внешний вид окна настройки потоков различный в разных версиях пакета VisualDSP++, и, кроме того, в зависимости от типа выбранного процессора меняется доступный набор периферийных устройств. Для настройки нужно подключить к входу порта файл «serin.dat» и к выходу файл «out.dat». Заходим в меню «Setting -> Streams» и выполняем следующие действия:

Подключаем входной файл. В группе настроек «Source» выбираем «File», находим файл «serin.dat», изменяем формат на «Binary». В группе настроек «Destination» выбираем устройство Sport1, нажимаем кнопку «Connect».

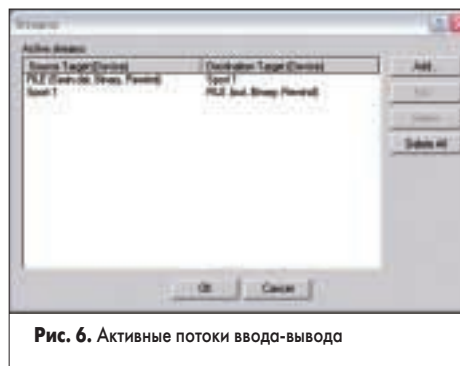


Рис. 6. Активные потоки ввода-вывода

Подключаем выходной файл. В группе настроек «Source» выбираем устройство Sport1, в группе настроек «Destination» находим файл «out.dat», изменяем формат на «Binary», нажимаем кнопку «Connect».

На закладке «Active» проверяем правильность настроек. Мы должны видеть два настроенных потока (рис. 6).

Нажимаем «OK».

Перед запуском программы, используя меню «Register» организуем окна «Program Control» и «Custom», в котором собираем регистры AX0, RX1, TX1. Можно добавить все используемые в программе регистры. Изменение содержимого этих регистров можно наблюдать в процессе пошагового выполнения программы.

В исходной программе кадровый сигнал формируется внутри порта. Для того, чтобы воспроизвести работу от внешнего кадрового сигнала заменим строку программы 0036 в окне дисассемблированной программы на

AX0=0x06a27. Затем установим параметры внешнего прерывания кадровым сигналом с помощью меню «Setting -> Interrupts». Выбираем источник прерывания кадровый сигнал приема RF1 и ставим число циклов Min 0, Max 10. Нажимаем кнопку «Restart» и проходим программу по шагам, нажимая кнопку F11.

Запуск программы

Ставим точку остановки на метку Sample, запускаем программу кнопкой «Run», оцениваем число циклов процессора и затем проходим программу по шагам кнопкой F11. При этом выполняется прием и передача обратно первого числа. Чтобы передать все числа, необходимо убрать точку остановки.

После окончания выполнения программы в окне выходной информации появится сообщение:

«Encountered end of file: FILE (C:\Program Files\Analog Devices\Visual DSP\218x\Examples\Example2\Serin.dat, Binary) Error in reading from stream for Sport1».

Это означает, что отладчик достиг конца файла. Проверьте содержимое выходного файла. Оно должно совпадать с входными значениями в случае правильной работы программы.

Исполнение программы можно проследить либо вызвав на экран исходный текст Sport.asm, либо дисассемблированный текст из памяти программы.

Следующим шагом, который может быть сделан для начального освоения пакета VisualDSP++, может быть как попытка модификации и объединения этого примера и предыдущего, так и исследование оставшихся примеров программ для сигнальных процессоров. Более подробно эти программы будут описаны в следующих статьях.

Заключение

В рамках статьи сложно полно описать возможности пакета и особенности его настройки. Кроме того, для успешной работы с сигнальными процессорами необходимо ознакомиться с особенностями их архитектуры, а также мнемоникой языка ассемблера, которые достаточно подробно описаны в переведенных сотрудниками центра ЦСП руководствах пользователя. В настоящее время напечатаны две книги: «Руководство пользователя по сигнальным микропроцессорам семейства ADSP-21XX» и «Руководство пользователя по сигнальным процессорам семейства SHARC ADSP-2106X». Обе эти книги, а также необходимые отладочные средства всегда можно приобрести в офисе фирмы ЭЛТЕХ. Кроме того, существует большое количество дополнительной информации на английском языке, в том числе и подробное описание возможностей и настроек пакета VisualDSP++. По умолчанию, после установки, вся эта документация находится в каталоге C:\Program Files\Analog Devices\VisualDSP\Docs.

Продолжение следует