

Средства отладки для микроконтроллера 1901BЦ1Т

ПЕТР ДЬЯЧКОВ, инженер, отдел разработки электронных модулей, ЗАО «ПКК Миландр»,
НИКИТА КУЗИН, ведущий инженер, отдел разработки электронных модулей, ЗАО «ПКК Миландр»

В статье рассмотрены основные особенности программирования и отладки микроконтроллера (МК) 1901BЦ1Т компании «Миландр». Предложены несколько способов отладки и принцип создания RELEASE-версии приложения.

Микроконтроллер 1901BЦ1Т ЗАО «ПКК Миландр» относится к классу устройств цифровых сигнальных контроллеров (Digital Signal Controller, DSC). В нём совмещены два высокопроизводительных ядра: 32-разрядное RISC-ядро, разработанное специально для встраиваемых мобильных приложений и близкое по своей структуре и набору команд к ядрам семейства 1986BE9x, и сопроцессор ЦОС с системой команд, совместимой с процессором TMS320C54x. Микросхема выпускается в 132-выводном корпусе, имеет внешнее питание 3,3 В и содержит встроенный регулятор на 1,8 В для питания цифровой части. 96 выводов микросхемы доступны пользователям в качестве портов общего назначения. Максимальное потребление кристалла не превышает 180 мА в случае, если активны процессорные ядра RISC и DSP.

Главным ядром данной микросхемы является RISC, а DSP — подчиненным, т.е. для начала выполнения DSP-программы RISC-ядро должно произвести определенные настройки DSP-подсистемы и записать в программную область код для выполнения. Далее ядра могут работать независимо друг от друга.

RISC-подсистема имеет линейное адресное пространство. Вся память DSP (память данных, программ и регистры периферийных блоков), кроме регистров ядра, также отображена в адресном пространстве RISC. Таким образом RISC-ядро имеет практически полный доступ к DSP-подсистеме. Из этого факта следует, что RISC-подсистема может использовать периферийные блоки DSP без включения DSP-ядра. В свою очередь DSP-ядро доступа к адресному пространству RISC не имеет, но может оповещать его о каких-либо событиях посредством запросов прерывания.

Отладочные компоненты МК 1901BЦ1Т представлены JTAG-интерфейсом по стандарту JTAG (IEEE standard 1149.1), который позволяет управлять процессором при отладке программ, контролировать состояние аппаратных средств процессора и тестировать его.

Теперь рассмотрим возможные варианты программирования и отладки пользовательских приложений. Разрабатывать программы для RISC-ядра микросхемы можно, используя широко известные среды: CodeMaster-ARM фирмы «Фитон», uVision компании Keil или IAR Embedded Workbench фирмы IAR Systems. Для DSP-подсистемы рекомендуется использовать Code Composer Studio компании Texas Instruments.

При выборе сред программирования uVision и IAR Embedded Workbench разработчику приложения для отладки через JTAG-эмулятор (каждая среда потребует собственного JTAG-эмулятора) доступно только RISC-ядро. При этом отладка не отличается от аналогичной процедуры для контроллеров серии 1986BE9x компании «Миландр». А DSP-подсистему в некоторых случаях можно

отлаживать косвенными путями. Для этого необходимо получить COFF-файл (*.out) из Code Composer Studio, написать сервисную программу для RISC-ядра, которая будет необходимым образом конфигурировать DSP-подсистему, загрузить COFF-файл в область программ (если необходимо, инициализировать область данных), далее следует запускать DSP-ядро. Т.е. при использовании uVision и IAR Embedded Workbench программисту для отладки DSP-подсистемы доступны и память программ, и память данных, но у него нет возможности устанавливать точки останова.

Для проверки и отладки ЦОС-алгоритмов на DSP-ядре с непрерывной обработкой данных отладочные компоненты МК 1901BЦ1Т можно подключать к компьютеру. Задействовав RISC, можно организовать передачу данных на ПК посредством интерфейса RS-232 (один из самых простых вариантов), что позволяет сохранить обработанные данные в файле для последующего анализа. Вывод сигналов на светодиодную индикацию отладочной платы позволяет с некоторой погрешностью оценить время выполнения участков кода, и как результат — принять решение о необходимости модификации алгоритма ЦОС или управляющей программы. Набор встроенных контроллеров (USB, UART, SPI, ADC и т.д.) позволяет реализовать программатор для анализа работы алгоритмов и управляющих программ. Предложенный метод является достаточно трудоемким, и с его помощью можно производить отладку программ узкого круга задач. Но сам принцип разработчику стоит взять на вооружение, т.к. в окончательном варианте проекта, когда все «строительные леса» режима отладки нужно будет снимать, останется именно этот подход, т.е. — старт программы (запуск RISC-ядра и конфигурирование периферии), настройка DSP-подсистемы, загрузка и запуск DSP-программы, выполнение задач управления RISC-ядром.

При отладке взаимодействия двух ядер требуется работать и с RISC-ядром, и с DSP-ядром, и здесь нужно принять во внимание некоторые особенности. На рисунке 1 представлена структурная схема возможных вариантов совместной работы МК 1901BЦ1Т с отладочными средствами в DEBUG-режиме.

При параллельной отладке программ для RISC-ядра и для DSP-ядра необходимо пользоваться средами CodeMaster-ARM фирмы «Фитон» и Code Composer Studio Texas Instruments. Для начала этой процедуры запускается CodeMaster-ARM, загружается или создается проект, проводится его сборка и программирование контроллера, затем запускается отладка. Далее можно приостановить выполнение программы RISC-ядром на точке останова и приступить к работе с DSP.

Проектно-ориентированная среда Code Composer Studio для DSP позволяет создавать или модифицировать

как отдельный проект, так и мультипроектное решение. В дальнейшем можно добавлять или удалять файлы исходного кода, объектные lib-файлы, команды для линкера. Кроме того есть возможность конфигурирования проекта с целью определения настроек для различных фаз создания программ. Например, сначала, пока идет отладка программы, можно определить DEBUG-конфигурацию проекта, а для заключительного этапа разработки программы — RELEASE. После разработки программы с помощью JTAG-эмулятора COFF-файл (*.out), полученный после компиляции и линковки проекта, можно загрузить во внутреннюю память МК 1901ВЦ1Т и осуществить отладку посредством программных или аппаратных точек останова. Это позволяет отлаживать дизассемблерный код, отслеживать состояние регистров ядра или периферии, области памяти в определенные моменты выполнения программы. Переключаясь между двумя средами программирования при параллельной отладке программ, пользователь может заметить обновление содержимого памяти данных DSP-подсистемы.

Способы запуска для DSP-ядра в RELEASE- и DEBUG-версиях проекта несколько различаются. В отличие от DEBUG-версии проекта, в RELEASE необходимо предусматривать процедуры конфигурирования и разрешения работы DSP. После окончания процедуры инициализации DSP-ядро может работать независимо, и при необходимости RISC вводится в режим пониженного энергопотребления.

Не рекомендуется производить запись новой программы для DSP непосредственным выполнением процедуры RISC-ядром в процессе отладки, т.к. это приведет к ошибке Code Composer Studio, и дальнейший процесс отладки будет невозможен. Мы предлагаем производить отладку отдельных частей (модулей или подпрограмм)

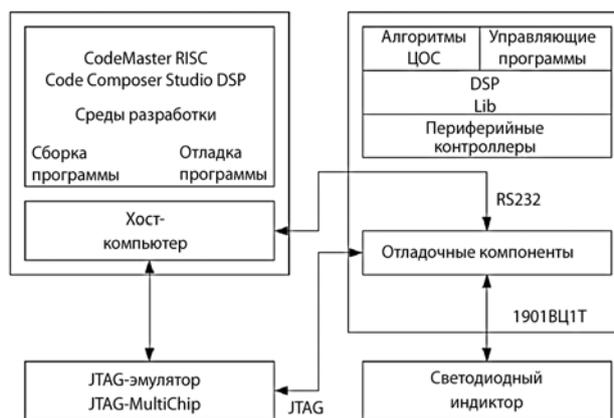


Рис. 1. Программно-отладочные средства микроконтроллера 1901ВЦ1Т

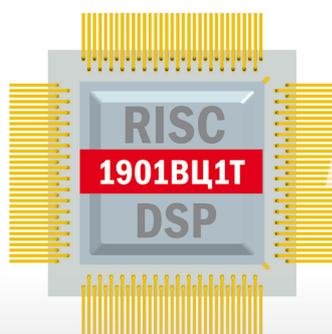
проекта, а не всего проекта в целом. Другими словами, если перед программистом поставлена задача изменения DSP-программы в зависимости от каких-либо условий, то необходимо разбить проект на небольшие подпрограммы (задачи), отладить каждую из них в отдельности и при получении окончательного работоспособного варианта создать RELEASE-версию проекта, в которой будет производиться перезапись подпрограмм в нужный момент времени.

Если возникает необходимость запустить другую подпрограмму на DSP-ядре, то в первую очередь необходимо остановить работу DSP, затем переписать в программную область новые данные, и далее вновь запустить ядро. Под остановкой работы DSP-ядра подразумевается установка бита сброса ядра в «1» (в регистре управления и статуса DSP) до полного обновления

ДВУХЪЯДЕРНЫЙ МИКРОПРОЦЕССОР 1901ВЦ1Т для высоконадежных применений



Тактовая частота RISC-ядра – до 100 МГц
Разрядность – 32 бит
Память программ – 128 Кбайт
Память данных – 32 Кбайт
Широкая периферия



-60°C
+85°C



Тактовая частота DSP-ядра – до 100 МГц
Тип данных – 16 бит с фиксированной запятой
Объем памяти – 256 Кбайт
Разрядность ALU – 40 бит
Аппаратный умножитель – 17x17
Цифровые фильтры



124498, Москва, Зеленоград, проезд 4806, дом 6
тел. +7 (495) 981-54-33, +7 (495) 981-54-36
www.milandr.ru info@milandr.ru



Рис. 2. Демонстрационно-отладочная плата для микроконтроллера 1901BЦ1T

программной области памяти, затем этот бит необходимо вернуть в состояние «0». В этом случае будет обеспечен корректный старт новой программы. Другими вариантами остановки выполнения программы DSP-подсистемой являются отключение тактового сигнала ядра или остановка конвейера DSP, но они неприемлемы, т.к. счетчик команд останется несброшенным, и, следовательно, загруженная программа будет выполняться

с неизвестного места, что приведет к непредсказуемым последствиям.

Как уже говорилось, в МК 1901BЦ1T DSP-ядро может выставлять запрос на прерывание RISC-ядру. Это удобный механизм оповещения программы, выполняемой на RISC-ядре, о событиях, которые произошли в DSP-подсистеме. Но если программист по некоторым причинам решил его не использовать или его недостаточно, то для взаимодействия между двумя ядрами и в качестве средств отладки можно выделить область ОЗУ требуемого объема в подсистеме DSP (к этой памяти может обращаться и RISC, и DSP) и разместить здесь регистры состояния процессов, в которых будет храниться информация о выполняемых операциях, наличии обработанных данных и т.д.

На рисунке 2 представлена демонстрационно-отладочная плата для данного микроконтроллера. В ее состав входит контактирующее устройство (колодка) для легкой замены микроконтроллера, интерфейсы RS-232 и USB, разъем для подключения microSD карт памяти, разъем для подключения внешних модулей к системной шине контроллера, ЖК-дисплей, клавиатура из 6 кнопок, аналоговые входы/выходы и т.д. Этот богатый набор периферии, реализованный на плате, позволяет создавать прототипы большого количества устройств различного назначения, а также программировать и отлаживать микроконтроллер.