

Архитектура AVR: развитие вширь и вглубь. Часть 2

Николай КОРОЛЕВ
korolev@argussoft.ru
Антон ШАБЫНИН
shabynin@argussoft.ru

В первой части статьи был приведен обзор основных характеристик новинок 8-разрядных и 32-разрядных AVR-микросхем. Во второй части более подробно рассматривается семейство контроллеров AVR32 с Flash-ПЗУ на кристалле, которое называется AT32UC3. Области применения этих недорогих микросхем являются промышленная автоматизация, медицина, измерительное и тестовое оборудование, коммуникации, POS-терминалы и т. д.

Введение в архитектуру AVR32 UC3

После выпуска микропроцессоров AVR32 AP7000 (Application Processor) с производительностью 210 DMIPS, специалисты Atmel обратились к «среднескоростному» сегменту — к контроллерам с тактовой частотой 50–70 МГц. Учитывая, что следующее поколение 8-разрядных AVR-контроллеров — Xmega — будет работать на частоте 32 МГц, получается достаточно плотное «покрытие» спектра рабочих частот, и соответственно, производительности (рис. 1).

Новые микроконтроллеры получили название AVR32 UC3, где аббревиатура UC означает «microcontroller». Оба ядра имеют 16/32-разрядный RISC-набор команд с расширениями инструкций, упрощающими применение AVR32-контроллеров в задачах цифровой фильтрации сигнала и управления двигателями. В частности, есть несколько типов команд умножения: группа команд mul (1 такт) — стандартное знаковое/беззнаковое умножение, группа команд mac (1/2 такта) — фракционное умножение с накоплением, группа команд mulMac64 (2/3/4 такта) — 64-разрядное умножение. Есть также группа команд 32-разрядного деления div (35 тактов) и команды работы с битами в памяти memc/memr/nemt (сброс, установка, переключение) с временем выполнения 1 такт. Можно сказать, что AVR32 UC3 — это усеченная версия AVR32 AP с пониженной скоростью ядра, без кэш-памяти и аппаратного векторного умножителя. Кроме того, вместо модуля управления памятью MMU (Memory Management Unit) использован более простой модуль защиты памяти MPU (Memory Protection Unit). Конвейер AVR32 UC3 имеет 3 стадии: IF (Instruction Fetch) — выборка команды, ID (Instruction Decode) — декодирование команды и EX (Instruction Execute) — выполнение команды. Принципиальное отличие микроконтроллеров AVR32 UC3 — наличие на кристалле Flash-памяти

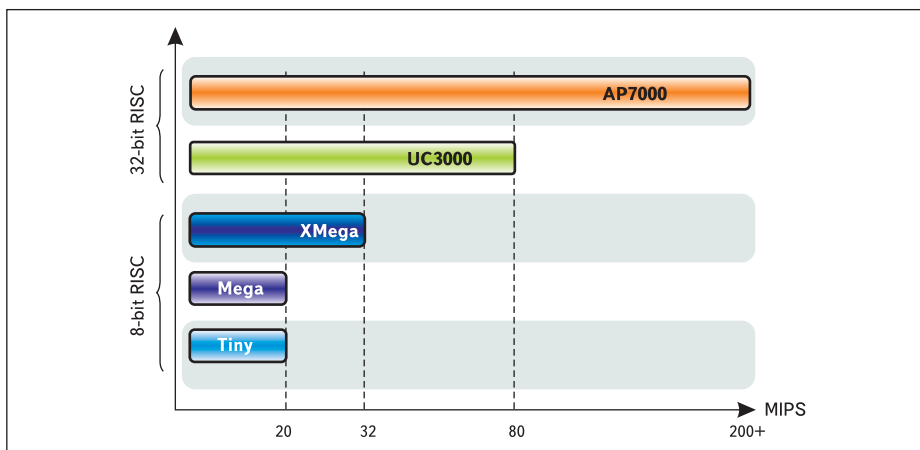


Рис. 1. Сводная диаграмма производительности AVR-контроллеров

объемом от 64 до 512 кбайт. Память может быть защищена от несанкционированного копирования. Программирование памяти может быть произведено промышленными программаторами (gang programming), а также через JTAG-интерфейс. Flash-память имеет максимальную частоту работы 30 МГц, для работы на повышенной частоте необходимо вводить такт ожидания. Однако, с учетом работы конвейера, общая скорость работы снижается не более чем на 15%.

В таблице 1 представлены отличия микросхем AP7000 и UC3.

Целью разработки нового класса микроконтроллеров было создание высокопроизводительных микросхем с низким энергопотреблением. Типовое относительное энергопотребление AT32UC3 составляет 1,65 мВт/DMIPS

Таблица 1. Отличия микросхем

| Микросхема AP7000 | Микросхема UC3 |
|----------------------|----------------------|
| Команды SIMD/DSP | Команды DSP |
| Java-ускоритель | Встроенное Flash-ПЗУ |
| 7-стадийный конвейер | 3-стадийный конвейер |
| Корпус 196–217 выв. | Корпус 48–144 выв. |

(0,6 мА/МГц) при скорости ядра 66 МГц и вычислительной производительности 80 DMIPS. Для сравнения: 8-разрядные AVR-микроконтроллеры имеют относительное энергопо-

Таблица 2. Основные характеристики серий AT32UC3A и AT32UC3B

| | Серия AT32UC3A | Серия AT32UC3B | | |
|--------------------------|---------------------|---------------------|------------|----|
| Частота ядра, МГц | 66 | 66 | | |
| Напряжение питания, В | 3,3 | 3,3 | | |
| Flash, кбайт | 512/256/128 | 256/128/64 | | |
| RAM, кбайт | 64/64/32 | 32/16/16 | | |
| Число внутренних шин | 6 | 5 | | |
| Число каналов DMA | 15 | 9 | | |
| Интерфейс EBI | Есть | Нет | | |
| Ethernet MAC | Есть | Нет | | |
| Интерфейс USB | Device & Host (OTG) | Device & Host (OTG) | | |
| Число таймеров | 3 | 3 | | |
| Число каналов PWM | 6+1 | 6+1 | | |
| Интерфейс USART | 4 | 3 | 2 | |
| Интерфейс SPI | 2 | 1 | | |
| Интерфейс TWI | 1 | 1 | | |
| Интерфейс SSC | 1 | 1 | Нет | |
| Число каналов ADC | 8 (10 бит) | 8 (10 бит) | 6 (10 бит) | |
| Тип корпуса | QFP144 QFP100 | QFP/QFN 64 | QFP/QFN 48 | |
| Выходы (ток 16 мА) | Нет | Нет | 4 | |
| Число линий ввода/вывода | 109 | 69 | 44 | 28 |

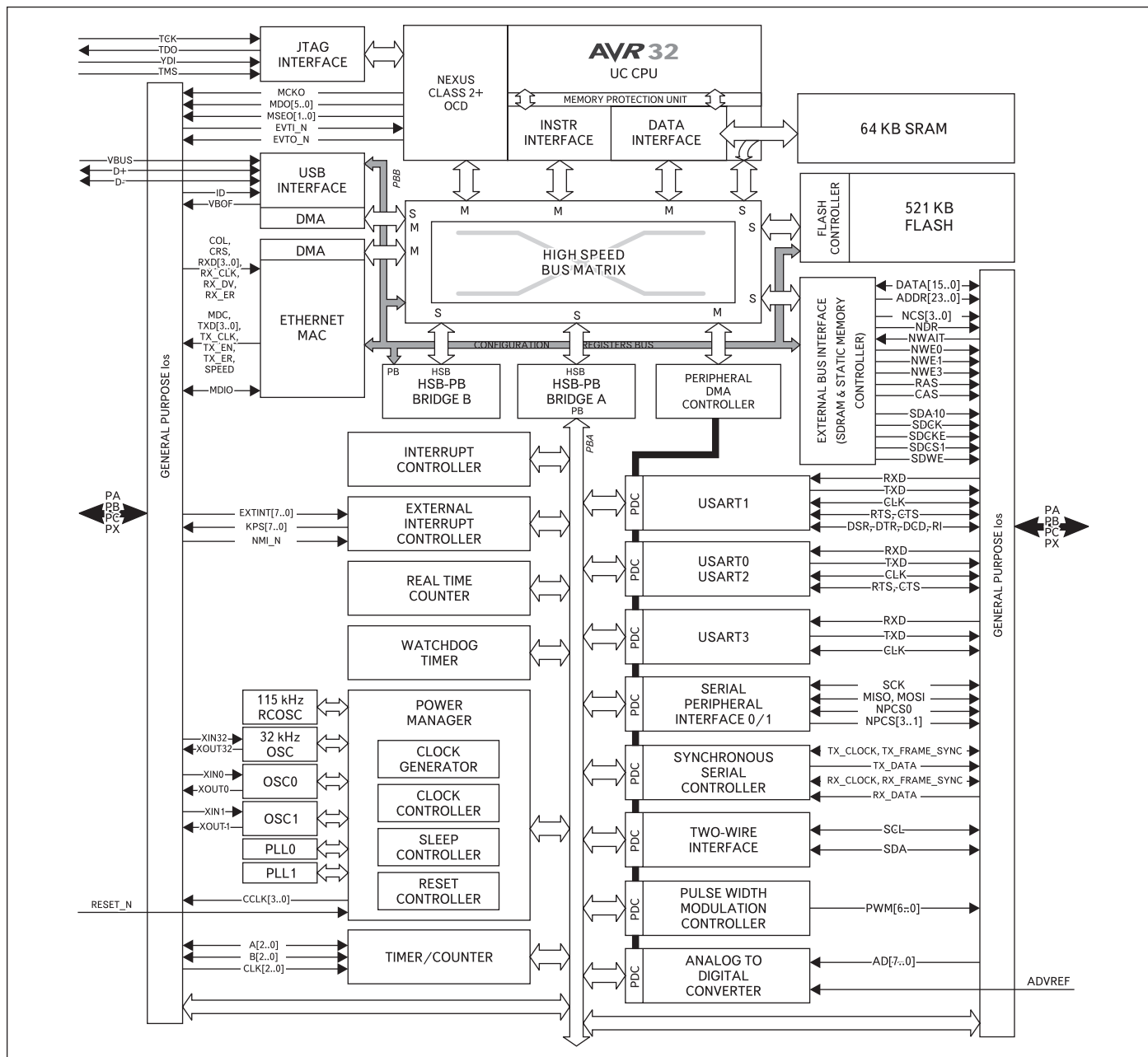


Рис. 2. Структурная схема контроллеров AT32UC3A

требление 5 мВт/DMIPS, 32-разрядные контроллеры LPC246x — 8,44 мВт/DMIPS.

Микроконтроллеры AT32UC3A имеют высокую степень интеграции. Помимо Flash-ПЗУ большой емкости и стандартных интерфейсов, таких как UART/USART, TWI, SPI и SSC, на кристалле размещены модуль MAC Ethernet 10/100 и USB-интерфейс Full Speed (12 Мбит/с) с поддержкой режима On-The-Go. 15 каналов прямого доступа к памяти освобождают ядро от рутинной работы по передаче информации. Для подключения внешней памяти RAM/SDRAM/Flash используется внешняя шина, которая содержит 24 адресных разряда и 16 разрядов данных.

Важной характеристикой кристалла является применение внутренней 6-слойной 32-разрядной шинной матрицы. Внутренние

шины работают на частоте ядра процессора, благодаря чему общая пропускная способность шин составляет 264 Мбайт/с. Структурная схема контроллеров AT32UC3A представлена на рис. 2.

Фактически, серия AT32UC3A является основой большого семейства AVR32-микроконтроллеров с Flash-памятью на кристалле. Вслед за микросхемами AT32UC3A выходят микросхемы AT32UC3B, отличающиеся отсутствием внешней параллельной шины и, как следствие, упакованные в более компактные корпуса. В таблице 2 представлены основные технические параметры этих серий. В разработке находится серия AT32UC3C/AT32UC3D, в которые будут добавлены 12-разрядный АЦП и ЦАП, а также CAN-контроллер. Серия AT32UC3E, в дополнение к пе-

речисленному, будет иметь на кристалле USB-контроллер High Speed (480 Мбит/с).

Аппаратные средства разработчика

Для микроконтроллеров AT32UC3 выпущены отладочные платы ATEVK1100 (для AT32UC3A) и ATEVK1101 (для AT32UC3B). Плата ATEVK1100 представлена на рис. 3.

Плата построена на базе микроконтроллера ATAVR32UC3A512. Кроме контроллера и кварцевого резонатора на 12 МГц на плате установлены следующие компоненты:

- микросхема SDRAM-памяти;
- микросхема Flash-памяти серии AT45;
- цифробуквенный ЖКИ, 4 строки, 20 символов с регулятором контраста;

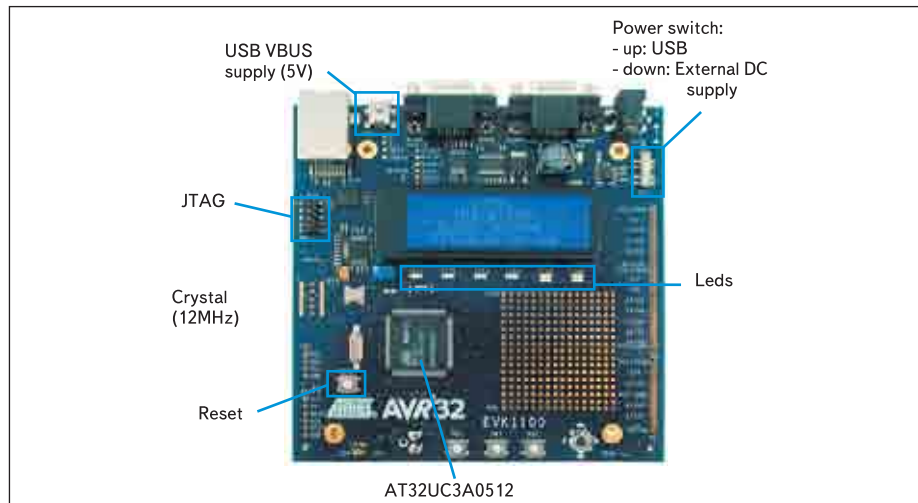


Рис. 3. Плата ATEVK1100

- шесть пользовательских светодиодов;
- контроллер физического уровня и разъем Ethernet со встроенным трансформатором;
- импульсный источник питания и разъем питания;
- 2 разъема портов RS-232 с микросхемами преобразователя уровней;
- разъем для подключения внутрисхемного эмулятора;
- разъем mini-USB;
- терморезистор, фоторезистор и потенциометр;
- джойстик;
- три пользовательские кнопки и кнопка сброса.

На плате также предусмотрено макетное поле и место для распайки разъемов расширения портов микроконтроллера. Плата может быть запитана либо от порта USB, либо от внешнего источника питания напряжением 8–20 В. С платой поставляется компакт-диск с примерами исходных кодов, а также библиотеками и необходимыми драйверами.

Также будет выпущена плата ATSTK600, на которой можно (при помощи соответствующего сменного адаптера) работать со всеми типами микросхем. Внешний вид платы ATSTK600 показан на рис. 4.

Для отладки программ можно использовать тот же внутрисхемный эмулятор ATJTAGICE2, который применяется для отладки проектов на 8-разрядных AVR-контроллерах.



Рис. 4. Плата ATSTK600

Программная поддержка микроконтроллеров AVR32

Функциональные возможности микроконтроллеров становятся богаче с выходом на рынок каждого нового семейства, и сегодня трудно себе представить проект, который начинается без всякой программной поддержки со стороны производителя микросхем. На освоение контроллеров «с нуля» и написание удобных библиотек уходит неоправданно большое количество времени, поэтому все производители стремятся сопроводить свои изделия программной поддержкой в виде примеров проектов, с подробным описанием, обеспечить их поддержку существующими программными продуктами, предоставить пользователям удобную среду разработки, то есть всеми доступными способами облегчить жизнь пользователям.

Программные средства разработки для микроконтроллеров семейства AVR32 делятся на две группы: средства, разработанные под Windows (коммерческая среда разработки

IAR), и средства на основе свободно распространяемого пакета GCC, работающие под Windows 2000/XP и под Linux. К последним относятся: вышедшая в апреле 2007 года AVR32 Studio, AVR32 GNU Toolchain (компилятор, набор необходимых вспомогательных программ и библиотек).

Средства разработки под Windows

Для тех, кто уже работал со средой IAR Embedded Workbench от IAR Systems, переход окажется совершенно незаметным (рис. 5). На сайте IAR после бесплатной регистрации доступна версия IAR Kickstart для работы с AVR32, имеющая ограничение длины выходного бинарного файла в 32 кбайт. Эта среда разработки была неофициально представлена на техническом семинаре Atmel в январе этого года. Участникам семинара была предоставлена возможность самостоятельно запустить и доработать подготовленный организаторами пример. Пользователи, еще не знакомые с IAR Embedded Workbench, смогли познакомиться с процессом разработки и отладки. Постоянные же пользователи лишний раз смогли убедиться, что реализация лозунга «Одна среда разработки для любых архитектур» сильно упрощает переход на новую архитектуру.

Основные отличия новой версии: в списке устройств присутствуют оба семейства: AP7000 и UC3 с настройками набора инструкций, применяемых при компиляции (DSP, SIMD, RMW). В список поддерживаемых эмуляторов в настоящее время включен только JTAGICE-mkII. Также в состав пакета входит симулятор. Примером для работы с IAR послужит порт FreeRTOS под архитектуру AVR32. С точки зрения программиста высокого уровня, написание кода под AVR32 ничем не отличается от написания кода под ARM7 или ARM9, и немного отличается от написания кода для AVR — богатые аппаратные возможности архитектуры, такие как PDC, FIFO, HMATRIX, поз-

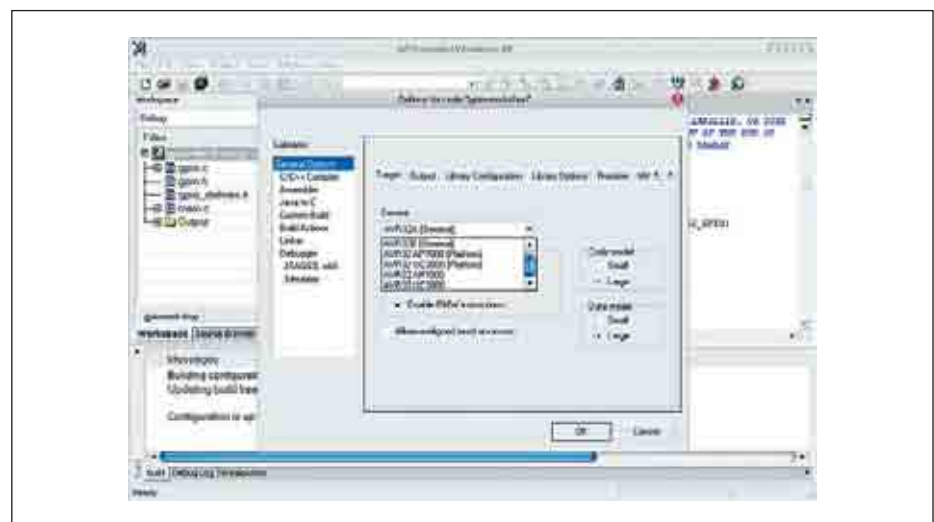


Рис. 5. Внешний вид IAR Embedded Workbench для AVR32

воляют многие действия выполнять аппаратно, не загружая ядро процессора. Главными недостатками среды IAR остаются зависимость от платформы Windows и невозможность компиляции большинства проектов, написанных для GCC, в том числе ядра Linux. Это несущественно для разработок на основе UC3 или AP7000 без использования Linux. Другим недостатком IAR является высокая стоимость полной версии пакета.

Средства разработки для Linux/Windows

Альтернативой компилятору от IAR является GCC. Поддержку в нем архитектуры AVR32 на себя взяла компания Atmel. Если для управления UC3 достаточно операционной системы реального времени, такой как FreeRTOS, то для AP7000 (от Application Processor — процессор для приложений) более чем удобным становится использование операционной системы Linux. Способствует этому и наличие модуля MMU, необходимого для полноценной работы ОС (в UC3 такого модуля нет), и специальных механизмов, заложенных в архитектуру AVR32, ускоряющих выполнение наиболее часто используемых вызовов ядра, и, естественно, поддержка архитектуры в ядре Linux. Заниматься разработками под Linux для микроконтроллеров значительно удобнее, имея эту ОС и на «большом» компьютере. К сожалению, использовать Linux в качестве основной ОС на рабочем месте удобно далеко не всем, и в Atmel это хорошо понимают. Уже после выхода первой части данной статьи Atmel опубликовала большое количество материалов по работе с AVR32. В первую очередь, представляет интерес информация о новой среде разработки на основе Eclipse — AVR32 Studio, а также AVR32 GNU Toolchain, который можно использовать как отдельно, так и совместно с AVR32 Studio. Здесь прослеживается аналогия между AVR Studio и пакетом WinAVR.

AVR32 Studio

Eclipse — это гибкая платформа и среда разработки, написанная на Java и ориентированная на использование плагинов, позволяющих добавлять любой необходимый функционал. AVR32 Studio включает в себя средства для комфортного редактирования исходного кода с подсветкой синтаксиса и прочими удобными средствами, ничем не уступающими по возможностям известной Microsoft Visual Studio (рис. 6). Имеется функционал для визуальной отладки, в том числе через удаленный GDB-сервер. В качестве компилятора используется AVR32 GNU Toolchain на основе GCC четвертой версии. Поддерживается эмулятор JTAGICE-mkII, платы на UC3 (EVK1100, STK600) и AP7000 (NGW100, STK1000).

Если с программированием UC3 все относительно понятно, то как быть, если необхо-

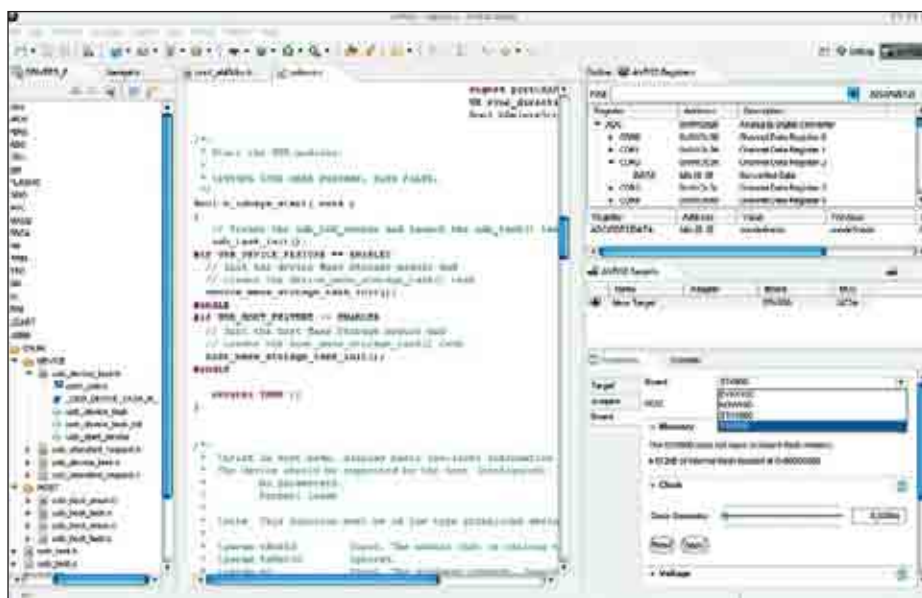


Рис. 6. AVR32 Studio

димо отлаживать пользовательскую программу для AP7000 под управлением Linux? Среда от IAR таких возможностей не предоставляет. В AVR32 Studio эта задача решается очень просто: нужно при создании проекта указать, что он будет выполняться под управлением Linux, и это одновременно позволит пользоваться функциями ядра, подключит при компиляции необходимые пути к заголовочным файлам и позволит выполнять отладку через удаленный GDB-сервер. AVR32 Studio включает в себя подробную документацию, в которой по шагам рассказано, как использовать возможности по компиляции и отладке проектов, описаны тонкости настройки правил компиляции, загрузки и выполнению отлаживаемой программы. Несмотря на применение GCC, который использует Makefile в качестве правил, пользователю не требуется писать их самому — эту функцию берет на себя среда разработки, оставляя при этом возможность ручной «тонкой настройки» проекта.

Atmel предлагает пользователям AVR32 GNU Toolchain для трех популярных дистрибутивов Linux: Ubuntu (.deb), Fedora Core (.rpm), OpenSUSE (.rpm), а также инсталлятор под Windows. Для работы в Windows используется пакет Cygwin. На скорости работы это, однако, почти не сказывается. Поскольку «родными» для GCC являются Unix-подобные системы, их выбор предпочтительнее. Пакет AVR32 Studio также существует в двух версиях: под Linux и под Windows. Для пользователя они полностью идентичны. AVR32 GNU Toolchain можно свободно загрузить с сайта www.atmel.com/avr32. Для загрузки AVR32 Studio требуется бесплатная регистрация, которую разработчики пакета объяснили необходимостью минимальной обратной связи с пользователями.

Программное обеспечение микроконтроллеров

Кроме удобной среды разработки, пользователям доступно большое количество примеров программ. Для UC3 доступны в виде исходного кода библиотеки на языке Си для работы со всеми периферийными модулями микроконтроллера, реализация нескольких классов USB-устройств, работа с USB On-The-Go (OTG), примеры использования шестислойной шинной матрицы для быстрой передачи данных от одного периферийного модуля к другому, работа АЦП, внешней шины EBI, подключение внешних SRAM, SDRAM и многое другое. Некоторые примеры, например проект, показывающий одновременную работу под FreeRTOS веб-сервера и USB-устройства, написаны как под IAR, так и под AVR32 Studio/GCC, однако большая часть примеров представлена в версии только под GCC. Реализация поддержки архитектуры AVR32 в IAR и GCC происходила при участии разработчиков архитектуры, поэтому качество оптимизации компиляторами программного кода оказалось примерно одинаковым. После этого становится понятно, почему разработчики в Atmel предпочитают использовать собственную, более функциональную и при том совершенно бесплатную среду разработки.

Для старшего подсемейства AVR32 — AP7000 выпущен диск AVR32 Linux BSP CD Image версии 2.0. В него вошли все средства, необходимые для работы с Linux на микроконтроллерах: документация по использованию, компилятор AVR32 GNU Toolchain под три дистрибутива Linux и Windows, исходные коды всех используемых на плате программ, ядра Linux, все примененные патчи и, что особенно приятно, набор из четырех скриптов, которые компилируют из исходных кодов ядро Linux с поддержкой всей

периферии, а также файловую систему с широким набором пользовательских приложений. Это избавляет пользователя от необходимости глубокого понимания механизма компиляции и дает ему возможность сразу сосредоточиться на разработке своего приложения или настройке имеющейся системы. При использовании Linux разработка уходит от привычного написания кода в сторону системного администрирования. Например, вместо того чтобы писать сетевое приложение, скажем, для передачи данных через Интернет, можно использовать веб-сервер и CGI-скрипты или интерпретируемые скрипты, перенаправляя вывод пользовательского приложения на сетевой интерфейс. Для приема входящих

соединений можно использовать «демона» `inetd`, вызывающего пользовательское приложение в ответ на входящее соединение. Разработка переходит на качественно новый уровень. Как уже упоминалось, отлаживать пользовательские приложения под Linux можно из AVR32 Studio.

Таким образом, на рынке 32-разрядных микроконтроллеров появилась прогрессивная, мощная архитектура, сочетающая в себе высокую производительность, развитую периферию и малое энергопотребление. Распространению этой архитектуры способствует продуманная структура интерфейсов на кристалле, простота написания программного кода, а также наличие доступной среды разработки под различные операционные сис-

темы, содержащей примеры исходных кодов для иллюстрации работы всех периферийных модулей. ■

Литература

1. Технические описания.
www.atmel.com/products/avr32/
2. Техническое руководство по архитектуре ядра AVR32UC. http://www.atmel.com/dyn/resources/prod_documents/doc32002.pdf
3. Описание Си-компилятора фирмы IAR.
www.iar.com
4. Справочная система из состава AVR32 Studio.
5. Материалы европейского дистрибьюторского семинара, январь 2007 года.
<http://atmel.argussoft.ru/seminars/>