

AVR-микроконтроллеры: семь ярких лет становления. Что дальше?

Часть 2. Ядро и периферийные блоки AVR второго поколения

Мы продолжаем цикл публикаций, посвященных одной из наиболее удачных 8-разрядных микроконтроллерных платформ — 8-разрядных RISC AVR производства Atmel Corp. Первая часть (см. «КиТ» № 1'2004), содержала краткий маркетинговый обзор мирового рынка микроконтроллеров и положения в нем AVR. Во второй части более детально рассмотрены некоторые особенности ядра и периферийных блоков AVR образца 2004 года. Третью часть планируется посвятить современным программным и аппаратным средствам поддержки для микроконтроллеров AVR. Интенсивный рост продаж AVR во всем мире явился стимулом для Atmel Corp. к их дальнейшему развитию и совершенствованию. Наиболее значительные изменения произошли в 2002–2003 годах, в результате чего появилось второе поколение AVR. Некоторым новым особенностям и функциям этих микроконтроллеров, не освещенным подробно в русскоязычной технической литературе, посвящена данная публикация.

Игорь Кривченко

ik@efo.ru

Елена Ламберт

elena@efo.ru

Романа Золотуха

roman@efo.ru

AVR представляет собой 8-разрядный RISC-микроконтроллер общего назначения с быстрым процессорным ядром. Он имеет Flash-память программ, память данных EEPROM и SRAM, порты ввода-вывода и интерфейсные схемы. Гарвардская архитектура AVR реализует полное логическое и физическое разделение как адресных пространств, так и информационных шин для обращения к памяти программ и данных. Такое построение кристалла в сочетании с одноуровневым конвейером команд обеспечивает существенное повышение производительности, которая в пределе может достигать значения 1 MIPS на 1 МГц. Отличительной чертой архитектуры AVR является регистровый файл, содержащий 32 регистра общего назначения. Шесть регистров файла могут использоваться как три 16-разрядных указателя адреса при косвенной адресации данных (X, Y и Z), что существенно повышает скорость пересылки данных при работе прикладной программы. В течение первых лет производства микроконтроллеров AVR корпорация Atmel выпускала их в трех семействах: «classic» (90S), «tiny» и «mega». В настоящее время, когда идет активное развитие AVR второго поколения, выпускаемых по технологии 0,35 мкм, семейство «classic» постепенно «распределяется» между семействами «tiny» и «mega». Поэтому здесь и далее будут рассматриваться только микроконтроллеры AVR семейств «mega» и «tiny», а семейство «classic», как не являющееся перспективным, рассматриваться не будет.

Архитектура и система команд AVR (до 133 различных инструкций) разрабатывались совместно с фирмой IAR Systems — производителем компиляторов языков программирования C/C++. Так, например, разрядность шины команд AVR составляет 16 бит, из которых 5 обеспечивают доступ к внутреннему регистровому файлу. Данное решение позволило отказаться от страничной организации памяти и традиционной банковской организации регистров общего назначения. Для микроконтроллеров AVR удастся получить высокую плотность кода при использовании языков высокого уровня, практически не теряя в производительности по сравнению с программами, написанными на языке Ассемблера. Разработчиками AVR был проведен тест по сравнению объема и времени выполнения кода, полученного в результате компиляции функции для определения максимального значения из массива 16 целых двухбайтных чисел. Функция написана на языке C и компилировалась для выполнения на современных микроконтроллерах различных производителей (см. таблицу).

```
int max(int *array)
{
    char a;
    int maximum=-32768;

    for (a=0;a<16;a++)
        if (array[a]>maximum)
            maximum=array[a];
    return (maximum);
}
```

Таблица. Размер и время выполнения кода для некоторых популярных микроконтроллеров

Производитель	Устройство	Максимальная скорость, МГц	Размер кода, кбайт	Количество циклов	Время выполнения, мкс
Atmel, AVR	ATmega16	16	32	227	14,2
Texas Instruments	MSP430	8	34	246	30,8
Atmel, C51	T89C51RD2	20	57	4200	210,0
Microchip	PIC18F452	40	92	716	17,9

Видно, что коды микроконтроллеров AVR и MSP430 близки по размеру, но скорость выполнения у MSP430 в два раза меньше, чем у AVR. По времени выполнения данной программы микроконтроллер PIC18F452 практически догоняет AVR, но размер кода у него в три раза больше. А микроконтроллером с ядром C51 нужно тактироваться с частотой 296 МГц, чтобы соответствовать AVR с тактированием на 16 МГц.

Память программ и данных

Программа заносится во Flash-память AVR при помощи параллельного программирования, последовательного низковольтного или высоковольтного программирования, а также самопрограммирования. Возможностью внутрисхемного программирования (функция ISP) через коммуникационный интерфейс SPI обладают все микроконтроллеры AVR, кроме tiny28. Причем для семейства «mega» — это аппаратный интерфейс, а в семействе «tiny» используется альтернативная функция линий ввода-вывода.

Все текущие версии кристаллов «mega» 2004 года имеют возможность самопрограммирования (функция SPM), которая реализуется с помощью блока загрузчика (Boot Loader Section). Эта функция позволяет обновлять программный код под управлением самого микроконтроллера, используя расположенную непосредственно во Flash-памяти программу загрузчика. Причем программа в блоке загрузчика может программировать весь массив Flash-памяти микроконтроллера, включая и саму область памяти загрузчика. При этом может использоваться любой доступный интерфейс для чтения и записи кода во Flash-память (USART, SPI, CAN и т. д.).

В микроконтроллерах семейства «mega» разрешается выполнять инструкции кода загрузчика при стирании или записи в область памяти программ для приложения («чтение во время записи»). В результате микроконтроллер может продолжать реагировать на запросы прерывания даже во время перепрограммирования. Для упрощения использования этой функции микроконтроллеры AVR поддерживают возможность переноса векторов прерывания из области памяти приложения в блок загрузчика.

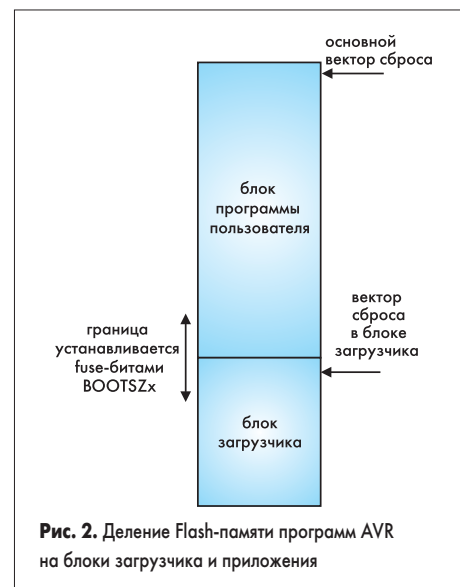
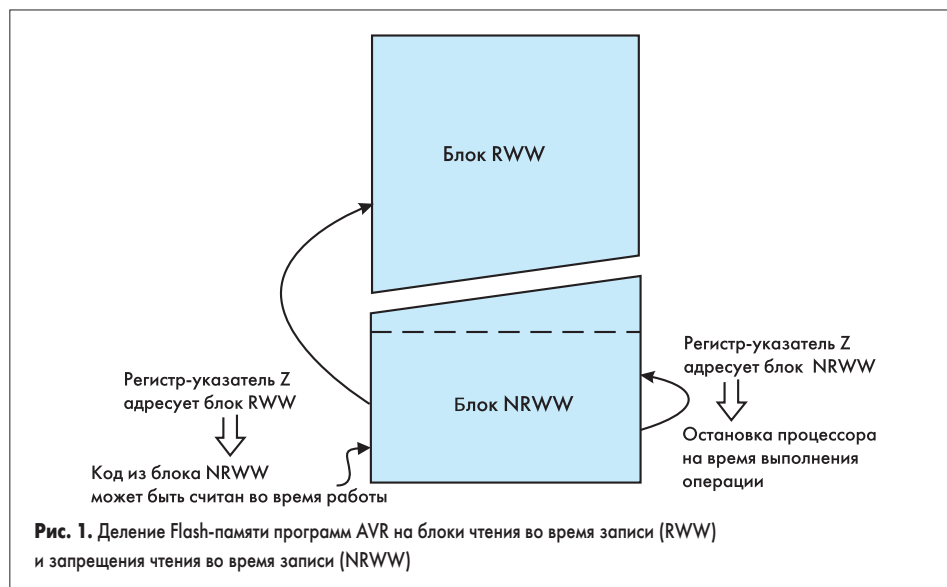
Механизм функции «чтение во время записи» следующий. Flash-память разделена на две фиксированных области: блок с функцией чтения во время записи (Read-While-Write, RWW) и блок без функции чтения во время записи (No Read-While-Write, NRWW). Основное различие между ними заключается в том, что при программировании блока RWW в блоке NRWW можно читать код, а при программировании блока NRWW процессор останавливается на все время выполнения операции (рис. 1). Загрузчик всегда располагается в блоке NRWW. Код, находящийся в NRWW, может быть прочтен, если загрузчик обновляет страницу в блоке RWW. Но если загрузчик стирает или программирует страницу в блоке RWW, то выбирать одновременно код из блока RWW нельзя, иначе может произойти сбой программы. Поэтому для таких случаев нужно или запретить прерывания, или переместить вектора прерываний в блок загрузчика. Вектора прерываний могут быть перемещены в начало блока загрузчика установкой бита IVSEL регистра управления MCUCR. Вектор сброса микроконтроллера также может быть перемещен в блок загрузчика установкой fuse-бита BOOTrST. Таким образом, с помощью управления состоянием бита BOOTrST можно определять

начало выполнения программы (из блока загрузчика или из блока памяти программ для приложения), а также размещать во Flash-памяти микроконтроллера две различные программы и запускать их на выполнение без необходимости перепрограммирования всего массива памяти программ.

Размер блока загрузчика микроконтроллеров AVR конфигурируется fuse-битами BOOTrSZx (см. рис. 2). Свободная память в области загрузчика может быть также использована в качестве дополнительного пространства для рабочего кода или для хранения редко обновляемых данных, полученных во время выполнения кода. Размеры блока загрузчика могут изменяться в пределах от 256 байт (используется только функция самопрограммирования) до 4 кбайт, когда необходимо реализовывать различные алгоритмы шифрации-дешифрации. Например, алгоритм по стандарту DES (Data Encryption Standard) занимает 2 кбайт памяти. Отметим, что в новых кристаллах tiny13 и tiny2313 весь массив Flash-памяти программ представляет собой блок загрузчика.

Гарантированное число циклов перезаписи Flash-памяти у микроконтроллеров AVR второго поколения составляет не менее 10 тыс. циклов при типовом значении 100 тыс. циклов. В официальной технической документации Atmel Corp. указывается значение 10 тыс. циклов. Следует отметить, что у AVR первого поколения гарантированное число циклов перезаписи Flash-памяти программ составляло всего одну тысячу циклов.

Все микроконтроллеры AVR (кроме tiny11 и tiny28) имеют также блок энергонезависимой памяти данных EEPROM, который обычно используется для хранения промежуточных данных, констант, таблиц перекодировок, калибровочных коэффициентов и т. п. Эта память может быть загружена извне как через интерфейс SPI (внутрисхемным программатором), так и с помощью параллельного программатора. Число циклов гарантированной перезаписи блока EEPROM составляет не менее 100 тыс. при типовом значении 1 млн. Срок хранения данных на EEPROM составляет 20 лет.



Особый интерес вызывает время гарантированного сохранения данных в энергонезависимой памяти на кристалле микроконтроллера (Data Retention) и устойчивость памяти к воздействию различных внешних факторов. Для микроконтроллеров AVR значение Data Retention составляет 20 лет как для Flash-памяти программ, так и для памяти данных EEPROM на кристалле. Но информация эта в корпорации Atmel является внутренней, и в официальных Data Sheet на микроконтроллеры она отсутствует. Тем не менее, Atmel Corp. по запросу клиента может выпустить специальный документ, в котором будет указано значение Data Retention для конкретного типа микроконтроллера с соответствующим номером партии и лота кремниевой пластины. Atmel Corp., естественно, признает, что подобные данные обязаны присутствовать в официальной документации на выпускаемую ею продукцию и обещает включить параметр Data Retention в будущие версии Data Sheet на микроконтроллеры AVR. К сожалению, точная дата этого события пока не называется.

Какие же внешние факторы главным образом влияют на сохранность данных во Flash- и EEPROM-ячейках памяти AVR? Основным параметром — температура кристалла. Известно, что с повышением температуры возрастают токи утечки в ячейках запоминающих устройств; зависимость эта экспоненциальная. Чем больше токи утечки, тем меньше период гарантированного сохранения информации в памяти микросхемы. По оценкам специалистов Atmel, если кристалл, например, работает при температуре 25 °C, то время сохранности данных будет практически бесконечным, десятки лет. Если этот же кристалл будет работать при температуре 105 °C, то оно уже составит максимум единицы лет. Но самый главный «убийца» — это термоциклирование, работа кристалла при значительных перепадах температур в несколько десятков градусов. В таких условиях время сохранности информации в ячейках энергонезависимой памяти сокращается еще больше. Поэтому следует всячески избегать термоциклирования, даже при хранении микросхем, но особенно — при работе микроконтроллера в активном режиме.

Несанкционированный доступ к массивам Flash-памяти программ и энергонезависимой памяти данных EEPROM может быть заблокирован установкой битов защиты микроконтроллера. Снятие установленных битов защиты возможно только путем стирания всего массива Flash-памяти, причем сама память стирается в первую очередь, а биты защиты — в последнюю. Благодаря этому энергонезависимая память в микроконтроллерах AVR надежно защищена от несанкционированного доступа.

Внутренняя оперативная память SRAM имеется у всех микроконтроллеров семейства «mega», а также у трех представителей семейства «tiny» — tiny26, tiny13 и tiny2313. Для некоторых микроконтроллеров семейства «mega» возможна организация подключения внешней памяти данных SRAM объемом до 64 кбайт.

Стандартные периферийные блоки

Большинство периферийных блоков не терпело каких-либо изменений и мы не будем подробно на них останавливаться. Микроконтроллеры AVR имеют от 1 до 4 таймеров-счетчиков общего назначения с разрядностью 8 или 16 бит, а также сторожевой таймер WDT. Примечательно, что микроконтроллеры AVR второго поколения имеют дополнительный вектор прерывания от сторожевого таймера. Для временных характеристик переполнения WDT в Data Sheet на AVR пока указываются только усредненные, типовые значения. Их следует использовать только в качестве оценки, так как частота собственного генератора WDT зависит нелинейно как от температуры, так и от напряжения питания микроконтроллера. В последующих «поколениях» технической документации Atmel Corp. будут указываться как минимальные, так и максимальные значения для времени переполнения сторожевого таймера.

Система реального времени RTC реализована во всех микроконтроллерах семейства «mega», кроме mega8515. Таймер-счетчик RTC имеет отдельный делитель, который может быть программным способом подключен или к источнику основной тактовой частоты, или к дополнительному асинхронному источнику опорной частоты (кварцевый резонатор или внешний синхросигнал). Для этой цели зарезервированы два внешних вывода микросхемы. Внутренний осциллятор, нагруженный на счетный вход таймера-счетчика RTC, оптимизирован для работы с внешним «часовым» кварцевым резонатором 32,768 кГц.

Аналого-цифровой преобразователь построен по схеме АЦП последовательных приближений с устройством выборки и хранения. Число независимых каналов преобразования определяется типом микроконтроллера, разрядность АЦП составляет 10 бит. Время преобразования выбирается программно с помощью установки коэффициента делителя частоты, входящего в состав блока АЦП. Важной особенностью аналого-цифрового преобразователя AVR является режим подавления шума, при котором на точность преобразования не оказывают влияние помехи, возникающие при работе процессорного ядра.

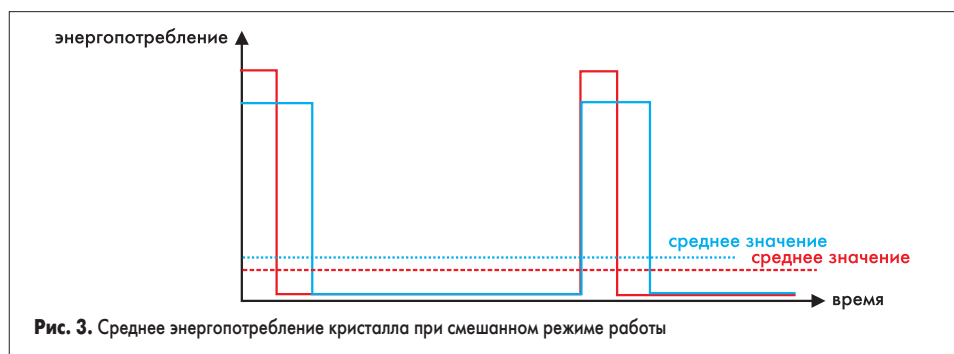
Порты ввода-вывода AVR имеют от 3 до 54 независимых линий ввода-вывода. Выходные драйверы обеспечивают токовую нагрузочную способность 20 мА на линию порта (втекающий ток), что позволяет непосред-

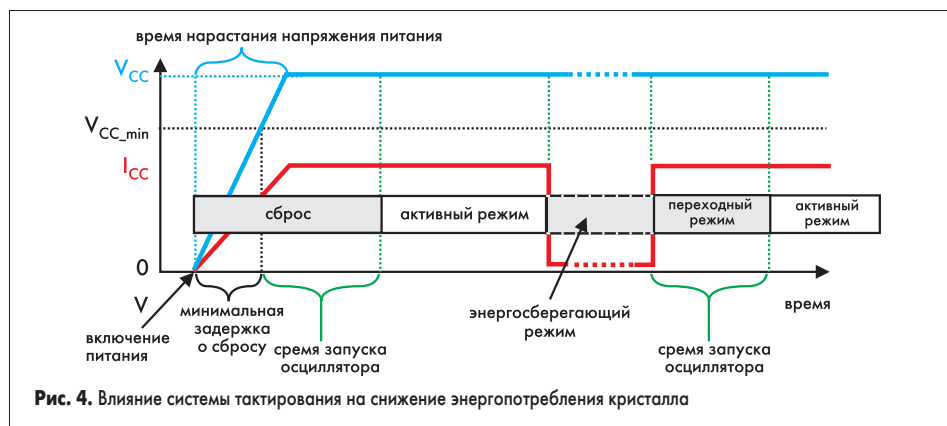
ственно подключать к микроконтроллеру светодиоды и биполярные транзисторы.

Микроконтроллеры AVR могут быть переведены программным путем в один из шести режимов пониженного энергопотребления. Для разных семейств AVR и различных микроконтроллеров в пределах каждого семейства изменяется количество и сочетание доступных режимов пониженного энергопотребления. В режиме холостого хода (IDLE) работу прекращает только процессор, а внутренний генератор синхросигналов, таймеры, система прерываний и сторожевой таймер продолжают функционировать. В режиме Power Down сохраняется содержимое регистрового файла, но останавливается внутренний генератор синхросигналов. Режим Power Save, который имеется только у AVR с RTC, похож на режим Power Down, но здесь допускается независимая работа дополнительного таймера-счетчика RTC. В режиме подавления шума при работе аналого-цифрового преобразователя (ADC Noise Reduction) останавливается процессорное ядро, но разрешена работа АЦП, двухпроводного интерфейса I²C и сторожевого таймера. Основной режим ожидания (Standby) также идентичен режиму Power Down, но работа тактового генератора не прекращается. Дополнительный режим ожидания (Extended Standby) идентичен режиму Power Save, но здесь работа тактового генератора тоже не прекращается.

Перевод микроконтроллера в один из энергосберегающих режимов дает значительное снижение суммарного энергопотребления кристалла. Если приложение выполняется на максимальной скорости за короткие промежутки времени, а потом микроконтроллер переводится в энергосберегающий режим, то среднее значение энергопотребления оказывается меньше по сравнению с тем, если выполнять ту же задачу на меньшей скорости, но соответственно за большие промежутки времени (рис. 3). В этом и заключается одно из преимуществ AVR — высокая плотность программного кода и одновременно большая скорость выполнения приложения по сравнению с другими микроконтроллерами (см., например, табл. 1) дают в среднем выигрыш энергопотребления кристалла, даже если в режиме Power Down параметры микроконтроллера AVR оказываются хуже, чем у конкурентов.

В разработке у Atmel в настоящее время находятся два новых аппаратных узла AVR, о которых говорилось в первой части. Это усовершенствованный контроллер пре-





рываний и новая система тактирования всего микроконтроллера — как самого ядра, так и его встроенных периферийных устройств.

Система тактирования для AVR второго поколения уже была существенно доработана по сравнению с реализацией тактирования у первых микроконтроллеров AVR. В качестве образца может быть рассмотрена система тактирования серийно выпускаемых кристаллов mega128 или mega169. Она стала достаточно гибкой, может использовать пять различных источников формирования тактового сигнала. Внутренний тактовый генератор может запускаться от внешнего генератора, внешнего кварцевого резонатора, а также от внешней RC-цепочки. Максимальная тактовая частота для новых кристаллов (например, tiny13) составляет 24 МГц при напряжении питания 5,5 В. Все AVR полностью статические, их минимальная рабочая частота ничем не ограничена вплоть до пошагового режима.

Для уменьшения длительности периодов работы микроконтроллера в активном режиме и, следовательно, для снижения энергопотребления кристалла, следует оптимально выбирать способ тактирования AVR (рис. 4). Время запуска осциллятора должно быть как можно меньше. Для этих целей удобно использовать внутренний RC-генератор, который выводит микроконтроллер на рабочий режим всего за 6 периодов тактовой частоты. Такое же время запуска можно получить, используя внешнюю RC-цепочку или внешний синхросигнал.

Выбор способа тактирования кристалла осуществляется путем установки или сброса fuse-битов CKSEL0...3, и пока еще этот выбор может быть сделан только на этапе программирования микроконтроллера. Чрезвычайно заманчиво было бы иметь возможность выбирать источник тактового сигнала «на лету», в процессе работы кристалла. Именно эта задача и стоит перед разработчиками AVR при создании новой системы тактирования кристалла.

Особый интерес представляет возможность тактирования AVR от встроенного RC-генератора, который калибруется на фабрике на разные частоты — от 1,0 до 9,6 МГц. Всего у AVR имеется восемь разновидностей встроенного RC-генератора в пяти официальных версиях. Большинство популярных микроконтроллеров AVR общего назначения имеют RC-генератор версии 3.0/3.1, который

с помощью установочных битов CKSELx может быть сконфигурирован на 4 возможные рабочие частоты: 1, 2, 4 и 8 МГц. Это не всегда бывает удобно. Например, для стандартной поддержки U(S)ART тактовая частота должна быть 3,56 МГц. В подобных случаях генератор можно откалибровать на работу с нужным значением тактовой частоты.

В последних версиях Data Sheet на микроконтроллеры AVR точность поддержания системной тактовой частоты внутренним RC-генератором приводится как $\pm 3\%$, хотя раньше она указывалась как $\pm 1\%$. Это сделано сознательно. В процессе изготовления на фабрике встроенный RC-генератор калибруется, как и прежде, с точностью $\pm 1\%$ и настройки фиксируются в регистре OSCCAL. Но калибровка эта проводится при температуре 25 °С, а микроконтроллер впоследствии может работать в диапазоне температур от -40 до +85 °С. В дополнение к этому значение частоты также может отличаться для разных версий выпускаемых кремниевых пластин с микроконтроллерами при одном и том же значении регистра OSCCAL. Поэтому в документации и указывается трехпроцентная точность поддержания системной тактовой частоты от установленного значения в случае работы кристалла от внутреннего RC-генератора в широком диапазоне рабочих температур и напряжений питания.

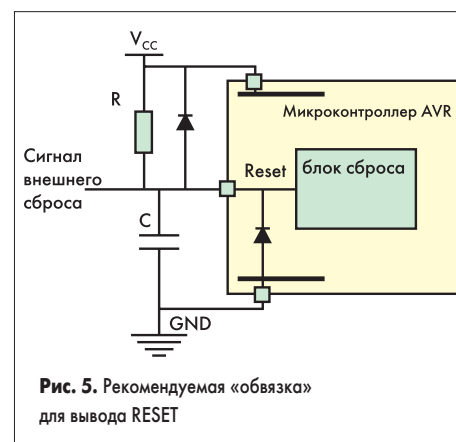
Как уже отмечалось, генератор калибруется на фабрике с точностью $\pm 1\%$. И он действительно работает с точностью $\pm 1\%$, но при определенных значениях температуры и напряжения питания микроконтроллера. Разработчик путем дополнительной калибровки и записи данных в регистр OSCCAL может довести точность поддержания системной тактовой частоты даже до $\pm 0,5\%$, что составляет минимальное значение погрешности. Дело в том, что величина шага калибровки не может быть определена точнее, так как зависимость частоты RC-генератора от значения OSCCAL также является нелинейной.

Рекомендации для разработчика здесь могут быть даны следующие: сначала нужно определить наиболее вероятный режим работы микроконтроллера по температуре и напряжению питания, а затем откалибровать генератор при этих условиях для работы с точностью $< 1\%$. Исключение здесь составляет лишь несколько кристаллов AVR — mega162/169 и tiny13/2313. Гарантированный «коридор» точности RC-генератора для этих

микроконтроллеров составляет $\pm 10\%$ при минимально достижимой погрешности калибровки $\pm 2\%$. Процесс калибровки внутреннего RC-генератора AVR не является сложным, он подробно описан в руководстве пользователя AVR053: «Calibration of the internal RC oscillator». Его можно осуществить с помощью программатора AVRISP, отладочной платы STK500 или эмулятора JTAG ICE.

Несколько слов про организацию системы сброса. Вывод RESET в микроконтроллерах AVR имеет активный сигнал низкого уровня, а для перехода в режим высоковольтного или параллельного программирования на него следует подать напряжение 11,5–12,5 В. Вывод RESET у AVR является асинхронным — для сброса микроконтроллера не обязательно наличие тактовой частоты кристалла. Это обстоятельство является очень удобным, особенно при реализации конечных приложений с батарейным питанием.

Линия сброса имеет внутренний резистор привязки к шине питания для увеличения помехозащищенности, который калибруется на фабрике при изготовлении микроконтроллера. На практике, особенно в условиях сильных внешних помех, сопротивление этого резистора (100–500 кОм) оказывается слишком большим, и при отсутствии на линии RESET сигнала высокого уровня может происходить случайный сброс микроконтроллера. Существует несколько способов снижения риска непреднамеренного сброса. Например, можно использовать внешний супервизор питания. При наличии внутреннего блока BOD (Brown-out Detection) используются более простые методы. А для сохранения возможности режима высоковольтного программирования и режима сброса к линии RESET подключается внешний подтягивающий резистор с рекомендуемым значением сопротивления от 4,7 до 10 кОм. Для дополнительной защиты линии RESET от внешних помех рекомендуется также шунтировать ее на землю с помощью внешнего конденсатора. На рис. 5 приведена типовая схема внешней «обвязки» для линии RESET. Здесь также показан еще один дополнительный внешний компонент — диод. Для поддержки режима высоковольтного программирования микроконтроллеры AVR не имеют стандартного внутреннего диода для защиты от избыточного напряжения на цифровом входе RESET. Поэтому, если



высоковольтное программирование не используется, то для защиты от помех рекомендуется подключать внешний диод между линией RESET и шиной питания микроконтроллера.

Как уже отмечалось, вход внешнего сброса RESET используется новым однопроводным интерфейсом debugWIRE при отладке программного обеспечения микроконтроллера. Но debugWIRE при своей работе ориентируется на длительность временных интервалов между импульсами в последовательности. Наличие конденсатора, подключенного параллельно входу RESET, будет приводить к сбоям в работе этого интерфейса. Поэтому, если планируется отлаживать микроконтроллер на целевой плате с помощью debugWIRE, то необходимо предусмотреть переключку, чтобы отключать этот конденсатор на время отладки прикладного программного обеспечения.

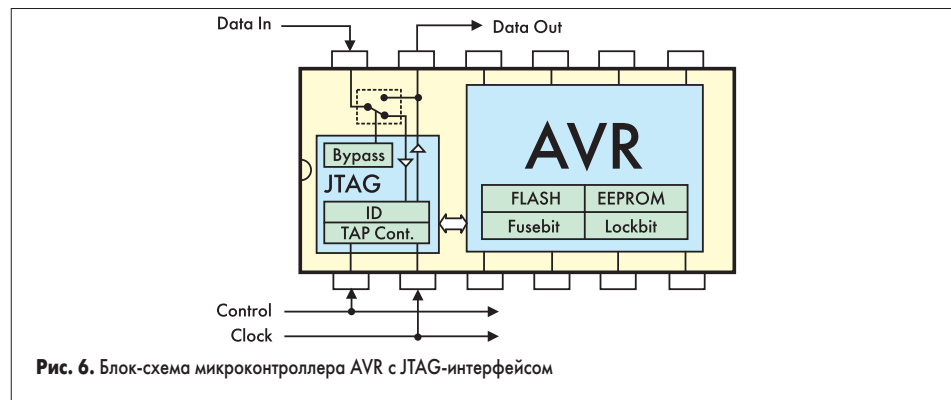
Если линия сброса не используется и внутрисхемное программирование не требуется, то вывод RESET может быть присоединен непосредственно к шине питания микроконтроллера.

Коммуникационные интерфейсы

Некоторые новые кристаллы AVR (tiny26, tiny2313, mega169) содержат универсальный последовательный интерфейс USI, который может быть программным образом сконфигурирован для работы в качестве коммуникационных интерфейсов SPI или I²C. Дополнительно USI может быть запрограммирован как полудуплексный UART или 4/12-разрядный счетчик. Использование USI с минимумом программного обеспечения позволяет существенно повысить скорость передачи данных и сократить объем программного кода по сравнению с программной эмуляцией последовательных интерфейсов.

Блок управления тактированием USI в режиме I²C позволяет генерировать прерывание при определении начального состояния на шине I²C. Режим I²C интерфейса USI не имеет ограничения по времени нарастания выходного сигнала и фильтрации входных шумов. Для этого режима используются выводы SCL и SDA. В режиме SPI интерфейс USI не имеет функции выбора ведомого (slave select, SS). При необходимости эта функция может быть реализована программно.

Когда USI используется в качестве UART, то при передаче кадра, например, из 10 бит (1 стартовый, 8 бит данных, один стоповый), USI не хватает 2 битов. Это ограничение «обходится» благодаря ресурсам USI: по прерыванию на изменение уровня на входе USI определяется начало кадра в режиме приема (стартовый бит), отсчитываются 8 бит данных, стартовый и стоповый биты игнорируются. Таймер-счетчик микроконтроллера управляет тактированием входных и выходных битов, а 4-разрядный счетчик USI генерирует прерывания при сдвиге определенного числа битов. Таким образом, при организации UART с помощью USI таймер-счетчик управляет скоростью передачи данных,



а 4-разрядный счетчик определяет конец кадра. Для передачи и приема данных необходимы оба счетчика, поэтому USI может быть сконфигурирован только как полудуплексный UART. В режиме передачи регистр данных USI должен быть загружен дважды для того, чтобы можно было послать все 10 бит кадра.

Если USI не применяется для последовательной передачи данных, то он может быть использован как отдельный 4-разрядный счетчик с прерыванием при его переполнении. Следует обратить внимание, что если счетчик тактируется извне, увеличение счета будет происходить по обоим фронтам тактового сигнала. Совместное применение 4-разрядного счетчика USI и стандартного таймера-счетчика позволяет организовать 12-разрядный счетчик.

Если программно установить счетчик USI на максимальное значение (0x0F), то его можно использовать в качестве еще одного источника внешнего прерывания по фронту. Для этого следует с помощью бита USIC_{S1} выбрать в регистре управления USICR внешний источник тактирования USI. Если же битом USIC_{S1} в качестве источника тактирования USI выбрать тактовый сигнал микроконтроллера, то этот счетчик может работать как источник программного прерывания: запись в счетчик максимального значения (0x0F) приведет к установке флага запроса прерывания на следующем тактовом импульсе.

Все новые микроконтроллеры семейства «mega» имеют интерфейсы JTAG или debugWIRE для внутрисхемного программирования и фоновой отладки. Из серийно выпускаемых в настоящее время микроконтроллеров семейства «mega» JTAG-интерфейс содержат mega32, mega64, mega128, mega162 и mega169. JTAG-интерфейс использует 4 вывода микроконтроллера и позволяет внутрисхемно запрограммировать Flash-память программ, память данных EEPROM, fuse-биты и биты защиты, осуществлять тестирование монтажа на печатной плате с помощью блока граничного сканирования и фоновую отладку приложения с помощью блока OCD (On-Chip Debug). Также через JTAG можно калибровать внутренний RC-генератор микроконтроллера AVR. JTAG-интерфейс соответствует стандарту IEEE 1149.1.

Управление JTAG-интерфейсом осуществляется через Test Access Port или TAP (рис. 6). TAP представляет собой конечный автомат с 16 состояниями, переход в различные со-

стояния зависит от сигнала на выводе TMS в момент переднего фронта на выводе TCK. Вывод TMS (Test Mode Select) используется для управления конечным автоматом, вывод TCK (Test Clock) — для синхронизации интерфейса JTAG. Вывод TDI (Test Data In) представляет собой последовательный вход данных для сдвига в регистр команд или регистр данных (Scan Chains), а вывод TDO (Test Data Out) — последовательный выход данных из регистра команд или регистра данных. Сигнал TRST (Test ReSeT), который дается в стандарте IEEE 1149.1 как дополнительный, здесь не используется.

Для программирования микроконтроллеров AVR через JTAG-интерфейс нужны только эти 4 вывода микроконтроллера и выводы питания (GND, Vcc), внешнее напряжение питания 12 В не требуется. Для разрешения программирования через JTAG должен быть установлен fuse-бит JTAGEN, а бит JTD в регистре MCUCR необходимо сбросить для разрешения доступа к контроллеру TAP. С помощью JTAG можно запрограммировать и верифицировать Flash-память программ, память данных EEPROM, fuse-биты и биты защиты. Отметим, что если биты защиты LB1 или LB2 установлены, то уже не имеет значения, установлен fuse-бит JTAGEN или нет. Для программирования микросхемы в этом случае необходимо предварительно полностью стереть кристалл.

Микроконтроллеры поставляются с установленным fuse-битом JTAGEN. Если сбросить JTAGEN, то четыре вывода TAP используются как стандартные линии ввода-вывода, и TAP-контроллер находится в неактивном состоянии.

Программирование через JTAG для AVR поддерживается различными JTAG-программаторами производства третьих фирм. В качестве входных данных необходим файл SVF (Serial Vector Format). Программу AVRsvf, которая преобразует выходной шестнадцатеричный файл для прошивки Flash и EEPROM через JTAG в файл SVF, можно скачать на сайте Atmel (http://www.atmel.com/dyn/resources/prod_documents/avrsvf.zip).

Рассмотрим функцию внутрисхемной отладки, которая выполняется через интерфейс JTAG благодаря блоку фоновой отладки OCD, являющегося частью самого кристалла AVR. С помощью блока OCD невозможно реализовать некоторые удобные функции отладки, например, запись трассы, неограниченное число точек аппаратного останова

и т. д. Для этих целей существуют внутрисхемные эмуляторы, имитирующие с той или иной степенью достоверности поведение микроконтроллера. Но блок OCD дает другие преимущества — программный код выполняется в самом устройстве, поэтому электрические и временные параметры системы полностью соответствуют реальности.

Блок OCD поддерживает доступ ко всем внутренним периферийным узлам, внутренней и внешней SRAM, регистровому файлу, программному счетчику, Flash-памяти программ и памяти данных EEPROM. Он состоит из следующих основных частей:

- цепи сканирования, расположенной между процессорным ядром AVR и внешними периферийными узлами;
- блока точек останова;
- коммуникационного интерфейса между процессором и системой JTAG.

Все операции чтения или записи, которые выполняет отладчик (например, AVR Studio), осуществляются подачей инструкций AVR через внутреннюю цепь сканирования процессора AVR. Процессор отправляет результат в область памяти, отображенную на пространство ввода-вывода, которая является частью коммуникационного интерфейса между процессором и системой JTAG.

Блок точек останова может осуществлять останов при изменении линейного выполнения программы, то есть при выполнении прерываний, переходов, ветвлений, вызовов подпрограмм. Этот блок также реализует останов на каждом шаге (Single Step Break), две точки останова памяти программ и две комбинированные точки останова. Вместе эти 4 точки останова могут быть сконфигурированы следующим образом:

- 4 одиночных точки останова памяти программ;
- 3 одиночных точки останова памяти программ + 1 одиночная точка останова памяти данных;
- 2 одиночных точки останова памяти программ + 2 одиночных точки останова памяти данных;
- 2 одиночных точки останова памяти программ + 1 точка останова памяти программ с маской («range Break Point»);
- 2 одиночных точки останова памяти программ + 1 точка останова памяти данных с маской («range Break Point»).

Отладчик, например, AVR Studio, может использовать для своих внутренних целей один (или более) из этих ресурсов, уменьшая таким образом возможности разработчика по использованию точек останова. Внутрисхемный эмулятор JTAG ICE производства Atmel Corp. имеет (согласно описанию) 4 аппаратных точки останова, но пользователю доступны только три, четвертая используется самим эмулятором. Дополнительно может использоваться прерывание при изменении линейного выполнения программы.

Также возможно использование неограниченного числа точек останова с помощью команды BREAK. При использовании команды BREAK во время отладки действующая команда заменяется на команду BREAK во Flash-па-

мяти программ, поэтому страница Flash-памяти каждый раз перепрограммируется при добавлении или удалении точки останова.

В дополнение к fuse-биту JTAGEN для разрешения работы блока фоновой отладки OCD должен быть также установлен fuse-бит OCDEN. При установке битов защиты режим фоновой отладки отключается. Аппаратная внутрисхемная отладка микроконтроллеров AVR с блоком OCD поддерживается стандартным отладчиком Atmel — AVR Studio.

Граничное сканирование — еще одна полезная возможность, которая реализуется с помощью JTAG-интерфейса. Она используется для проверки правильности разводки целевой печатной платы и ее монтажа. При установке fuse-бита JTAGEN и снятия бита JTD в регистре ввода-вывода MCUCR разрешается доступ к цепи сканирования. Эта цепь включает все физические выходы микроконтроллера. Внутренняя цепь сканирования, являющаяся частью системы OCD, при этом отключается. Отметим, что при использовании интерфейса JTAG для граничного сканирования, тактовая частота JTAG TCK может быть выше, чем тактовая частота микроконтроллера AVR.

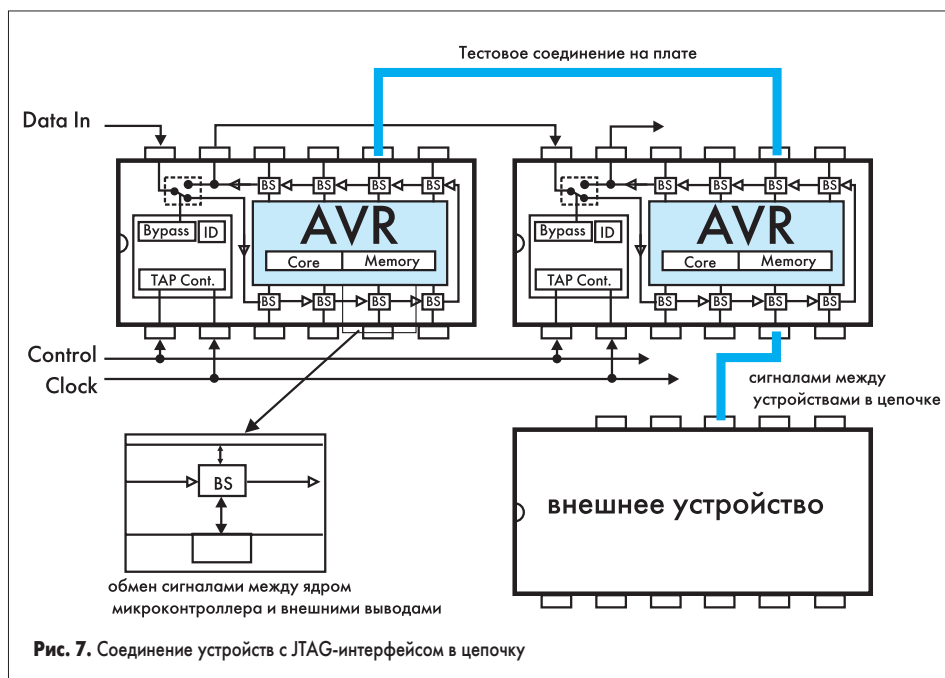
Блок граничного сканирования соответствует стандарту IEEE 1149.1. Цепь граничного сканирования позволяет подавать и наблюдать логические уровни на цифровых линиях ввода-вывода и состоит из ячеек граничного сканирования, расположенных на каждой линии ввода-вывода. Каждая ячейка может быть «прозрачной», выдавать логические сигналы, определять уровень действующего сигнала на линии ввода-вывода.

Микросхемы, имеющие JTAG-интерфейс, могут быть соединены последовательно в цепочку с помощью сигналов TDI/TDO (рис. 7). Логический анализатор управляет уровнями сигналов устройства, наблюдает за значениями входных сигналов от других устройств и сравнивает их с ожидаемыми значениями. Таким образом, блок граничного сканирова-

ния обеспечивает механизм для тестирования межсоединений и взаимодействия электронных компонентов на печатной плате, используя только 4 вывода контроллера TAP. Для реализации функции граничного сканирования необходимо описание устройства на языке BSDL (Boundary-scan Description Language). Порядок и функции битов в регистре данных граничного сканирования приводятся в технических описаниях на микроконтроллеры. Файлы BSDL для различных микроконтроллеров AVR доступны на сайте корпорации Atmel (http://www.atmel.com/dyn/resources/prod_documents/bsdl.zip).

Еще один коммуникационный интерфейс, который имеется во всех микроконтроллерах семейства «mega» и в микроконтроллере tiny2313 — это универсальный синхронный и асинхронный последовательный приемопередатчик USART. Вместо UART микроконтроллеры AVR содержат теперь USART, который, в отличие от своего предшественника, имеет синхронный режим передачи.

USART полностью совместим со старым интерфейсом UART в семействе AVR в отношении расположения битов во всех регистрах, скорости передачи данных, режимов передачи, работы регистра передачи и режима приема. Основное улучшение состоит в том, что регистр ввода-вывода данных приемника UDR стал двухуровневым. Два буферных регистра действуют как один циклический буфер FIFO, поэтому UDR интерфейса USART должен быть каждый раз считан при поступлении данных. Флаги ошибки (FE, DOR) и девятый бит данных (RxB8) сохраняются вместе с данными в регистре приемника. Поэтому эти биты следует считать перед обращением к регистру UDR, иначе статус ошибки будет потерян вместе со статусом регистра. Еще одно примечательное отличие состоит в том, что сдвиговый регистр в приемнике USART может работать как третий регистр, если принимаемым данным разрешено оставаться в последовательном сдвиговом регистре и если буферные регистры за-



полнены до того, как определено начало нового стартового бита. Таким образом, USART оказывается более устойчивым к возникновению сбойной ситуации по переполнению данных.

Выводы

Подводя итоги первой и второй части данного цикла статей с точки зрения новизны в аппаратной реализации микроконтроллеров AVR, можно сказать следующее. При сохранении процессорного ядра и архитектуры AVR, а соответственно, и совместимости микроконтроллеров «снизу вверх», корпорация Atmel продолжает совершенствовать характеристики периферийных блоков AVR и более полно использовать возможности усовершенствованной технологии 0,35 мкм. Микроконтроллеры функционируют в широком диапазоне питающих напряжений от 1,8 до 6,0 В. Увеличена производительность AVR второго поколения и снижено их энергопотребление при работе в энергосберегающих режимах. Максимальная тактовая частота при напряжении питания 5,5 В составляет 24 МГц, а при 1,8 В — 6 МГц. В режиме Power Down при напряжении питания 1,8 В ток потребления современных AVR не превышает 0,1 мкА. Для дополнительного управления энергопотреблением в микроконтроллерах семейства «mega» применяется специальный модуль, управ-

ляемый программно через 8-разрядный регистр PRR (Power Reduction Register). Все новые AVR имеют встроенный интерфейс для внутрисхемного программирования и фоновой отладки, многие содержат коммуникационный интерфейс USART. Микроконтроллеры AVR выпускаются теперь только для промышленного температурного диапазона с двумя диапазонами напряжения питания. Анонсирован первый кристалл AVR с контроллером CAN для промышленного и автомобильного температурного диапазонов, серийное производство которого планируется начать осенью 2004 года. Часть новых микроконтроллеров будет опрессовываться в корпуса с выводами, не содержащими свинца (Pb-free). Для обозначения таких микросхем в конце наименования вместо кода температурного диапазона проставляется литера «J».

В ближайшем будущем Atmel Corp. планирует для микроконтроллеров AVR еще ряд аппаратных новшеств. Будет добавлен еще один (четвертый) бит управления портами ввода-вывода с целью дальнейшего снижения энергопотребления кристалла путем отключения линий ввода-вывода от физического вывода микросхемы. Разрабатывается многоуровневый контроллер прерываний для облегчения реализации сложных задач на микроконтроллерах AVR с большим количеством периферийных устройств и большим объемом памяти программ. Также

в разработке находится новый аппаратный модуль для динамического переключения источника тактирования кристалла AVR без нарушения его текущей работы. Прорабатывается возможность дополнительной программной коммутации выхода внутреннего аналогового компаратора AVR на внешний вывод микросхемы. Также обсуждаются вопросы реализации аппаратного контроллера прямого доступа к памяти (DMA) на кристалле AVR.

Литература

1. Кривченко И. В. Микроконтроллеры общего назначения для встраиваемых приложений производства Atmel Corp. // Электронные компоненты № 5, 2002. с. 69–73.
2. Кривченко И. В., Ламберт Е. В. AVR-микроконтроллеры: семь ярких лет становления. Что дальше? Часть 1. Текущее положение AVR на мировом рынке микроконтроллеров и планы развития семейства в 2004 году // Компоненты и Технологии. 2004. № 1.
3. Королев Н. В., Королев Д. Н. AVR-микроконтроллеры второго поколения: новые аппаратные возможности // Компоненты и Технологии. 2003. № 4.
4. AVR Technical Training. Atmel Corp. Norway. December 2003.
5. <http://www.atmel.com/avr>.
6. <http://www.avrfreaks.net>.