

# Электроника

Юрий Ревич



**Практическое  
программирование  
микроконтроллеров**

**Atmel AVR**  
**на языке ассемблера**

**3-е издание**

Юрий Ревич

**Практическое  
программирование  
микроконтроллеров**

**Atmel AVR**  
**на языке ассемблера**

**3-е издание**

Санкт-Петербург  
«БХВ-Петербург»

2014

УДК 004.431.4  
ББК 32.973.26-018.1  
P32

**Ревич Ю. В.**

P32 Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. — 3-е изд., испр. — СПб.: БХВ-Петербург, 2014. — 368 с.: ил. — (Электроника)

ISBN 978-5-9775-3311-9

Изложены принципы функционирования, особенности архитектуры и приемы программирования микроконтроллеров Atmel AVR. Приведены готовые рецепты для программирования основных функций современной микроэлектронной аппаратуры: от реакции на нажатие кнопки или построения динамической индикации до сложных протоколов записи данных во внешнюю память или особенностей подключения часов реального времени. Особое внимание уделяется обмену данными микроэлектронных устройств с персональным компьютером, приводятся примеры программ. В книге учтены особенности современных моделей AVR и сопутствующих микросхем последних лет выпуска. Приложение содержит основные параметры микроконтроллеров AVR, перечень команд и тексты программ для них, а также список используемых терминов и аббревиатур.

*Для учащихся, инженерно-технических работников и радиолюбителей*

УДК 004.431.4  
ББК 32.973.26-018.1

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Редактор	<i>Леонид Кочин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 31.03.14.  
Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 29,67.  
Тираж 1500 экз. Заказ №  
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3311-9

© Ревич Ю. В., 2014  
© Оформление, издательство "БХВ-Петербург", 2014

# Оглавление

<b>ВВЕДЕНИЕ. Микроконтроллеры, их возникновение и применение</b> .....	7
Предыстория микроконтроллеров .....	8
Электроника в греческом стиле.....	10
Почему AVR?.....	12
Что дальше? .....	15
<b>ЧАСТЬ I. ОБЩИЕ ПРИНЦИПЫ УСТРОЙСТВА И ФУНКЦИОНИРОВАНИЯ ATMEL AVR</b> .....	17
<b>Глава 1. Обзор микроконтроллеров Atmel AVR</b> .....	19
Семейства AVR.....	21
Особенности практического использования МК AVR.....	23
<b>Глава 2. Общее устройство, организация памяти, тактирование, сброс</b> .....	27
Память программ .....	27
Память данных (ОЗУ, SRAM) .....	29
Энергонезависимая память данных (EEPROM).....	31
Способы тактирования.....	32
Сброс.....	34
Особенности подключения дополнительной внешней памяти данных .....	36
<b>Глава 3. Знакомство с периферийными устройствами</b> .....	39
Порты ввода-вывода.....	40
Таймеры-счетчики.....	42
Аналого-цифровой преобразователь.....	43
Последовательные порты.....	45
UART .....	45
Интерфейс SPI.....	49
Интерфейс TWI (I <sup>2</sup> C).....	52
Универсальный последовательный интерфейс USI.....	53
<b>Глава 4. Прерывания и режимы энергосбережения</b> .....	55
Прерывания .....	55
Разновидности прерываний .....	59
Режимы энергосбережения.....	61
Потребление МК AVR.....	62
Потребление МК AVR и режимы энергосбережения.....	65

<b>ЧАСТЬ II. ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ</b>	
<b>ATMEL AVR.....</b>	<b>67</b>
<b>Глава 5. Общие принципы программирования МК семейства AVR.....</b>	<b>69</b>
Ассемблер или C? .....	69
Способы и средства программирования AVR.....	73
Редактор кода .....	73
Об AVR Studio.....	74
Обустройство ассемблера .....	76
Программаторы.....	77
О hex-файлах .....	82
Команды, инструкции и нотация AVR-ассемблера .....	84
Числа и выражения .....	86
Директивы и функции .....	87
Общая структура AVR-программы .....	91
Обработка прерываний.....	93
<i>RESET</i> .....	96
Простейшая программа.....	97
Задержка .....	99
Программа счетчика .....	101
Использование прерываний .....	103
Задержка по таймеру .....	104
Программа счетчика с использованием прерываний .....	105
О конфигурационных битах.....	108
<b>Глава 6. Система команд AVR .....</b>	<b>111</b>
Команды передачи управления и регистр <i>SREG</i> .....	111
Команды проверки-пропуска.....	117
Команды логических операций.....	119
Команды сдвига и операции с битами .....	120
Команды арифметических операций .....	122
Команды пересылки данных.....	124
Команды управления системой .....	128
Выполнение типовых процедур на ассемблере.....	129
О стеке, локальных и глобальных переменных .....	131
<b>Глава 7. Арифметические операции.....</b>	<b>133</b>
Стандартные арифметические операции .....	134
Умножение многоразрядных чисел.....	135
Деление многоразрядных чисел .....	137
Операции с дробными числами.....	140
Генератор случайных чисел.....	142
Операции с числами в формате BCD.....	144
Отрицательные числа в МК.....	149
<b>Глава 8. Программирование таймеров.....</b>	<b>153</b>
8- и 16-разрядные таймеры .....	153
Формирование заданного значения частоты .....	156
Отсчет времени.....	159
Точная коррекция времени .....	164

Частотомер и периодомер.....	167
Частотомер .....	167
Периодомер .....	171
Управление динамической индикацией.....	174
LED-индикаторы и их подключение .....	175
Программирование динамической индикации.....	178
Таймеры в режиме PWM .....	181
<b>Глава 9. Использование EEPROM.....</b>	<b>187</b>
Еще раз о сохранности данных в EEPROM.....	187
Запись и чтение EEPROM.....	189
Хранение констант в EEPROM .....	191
<b>Глава 10. Аналоговый компаратор и АЦП.....</b>	<b>195</b>
Аналого-цифровые операции и их погрешности .....	195
Работа с аналоговым компаратором .....	198
Интегрирующий АЦП на компараторе.....	201
Принцип работы и расчетные формулы .....	202
Программа интегрирующего АЦП.....	206
Встроенный АЦП.....	209
Пример использования АЦП .....	213
Программа.....	215
<b>Глава 11. Программирование SPI.....</b>	<b>223</b>
Основные операции через SPI.....	223
Аппаратный вариант.....	224
Программный вариант.....	226
О разновидностях энергонезависимой памяти.....	227
Запись и чтение flash-памяти через SPI .....	229
Программа обмена с памятью 45DB011B по SPI .....	232
Запись и чтение flash-карт.....	232
Подключение карт MMC.....	234
Подача команд и инициализация MMC .....	235
Запись и чтение MMC .....	240
<b>Глава 12. Интерфейс TWI (I<sup>2</sup>C) и его практическое использование.....</b>	<b>245</b>
Базовый протокол I <sup>2</sup> C.....	245
Программная эмуляция протокола I <sup>2</sup> C .....	248
Запись данных во внешнюю энергонезависимую память.....	249
Режимы обмена с памятью AT24 .....	250
Программа.....	252
Часы с интерфейсом I <sup>2</sup> C.....	256
Запись данных .....	263
Чтение данных .....	267
<b>Глава 13. Программирование UART/USART .....</b>	<b>269</b>
Инициализация UART.....	270
Передача и прием данных.....	272
Пример установки часов DS1307 с помощью UART .....	275

Приемы защиты от сбоев при коммуникации.....	279
Проверка на четность .....	280
Как организовать корректный обмен.....	283
Дополнительные возможности USART .....	285
Реализация интерфейсов RS-232 и RS-485.....	286
Преобразователи уровня для RS-232 .....	290
RS-485 .....	294
<b>Глава 14. Режимы энергосбережения и сторожевой таймер .....</b>	<b>297</b>
Программирование режима энергосбережения .....	299
Пример прибора с батарейным питанием .....	300
Доработка программы .....	302
Использование сторожевого таймера .....	306
<b>ПРИЛОЖЕНИЯ .....</b>	<b>311</b>
<b>Приложение 1. Основные параметры микроконтроллеров Atmel AVR.....</b>	<b>313</b>
<b>Приложение 2. Команды Atmel AVR.....</b>	<b>321</b>
Арифметические и логические команды .....	322
Команды операций с битами .....	323
Команды сравнения .....	324
Команды передачи управления .....	325
Команды безусловного перехода и вызова подпрограмм .....	325
Команды проверки-пропуска и команды условного перехода .....	326
Команды переноса данных .....	327
Команды управления системой .....	328
<b>Приложение 3. Тексты программ .....</b>	<b>329</b>
Демонстрационная программа обмена данными с flash-памятью 45DB011B по интерфейсу SPI .....	329
Процедуры обмена по интерфейсу I <sup>2</sup> C .....	333
<b>Приложение 4. Обмен данными с персональным компьютером и отладка программ через UART .....</b>	<b>341</b>
Работа с COM-портом в Delphi .....	342
COM-порт и Windows API .....	342
Работа с COM через готовые компоненты .....	348
Установка линии RTS в DOS и Windows.....	349
Программа COM2000.....	351
Отладка программ с помощью терминальной программы .....	353
<b>Приложение 5. Словарь часто встречающихся аббревиатур и терминов .....</b>	<b>355</b>
Соответствие терминов на русском их переводу на английский.....	357
Соответствие терминов на английском их переводу на русский.....	358
<b>Литература .....</b>	<b>361</b>
<b>Предметный указатель .....</b>	<b>363</b>

# ВВЕДЕНИЕ

## Микроконтроллеры, их возникновение и применение

Говорят, что в 1960-е годы, наблюдая за участниками студенческих демонстраций протеста, Гордон Мур заметил: «Истинные революционеры — это мы». Ученик и сотрудник одного из изобретателей транзистора У. Шокли, в числе прочего считающегося основателем знаменитой Кремниевой долины, в свою очередь основатель и лидер компаний, которым суждено было сыграть ведущую роль в развитии микроэлектроники, Мур знал, что говорил. Парадоксальным образом именно изобретения Мура и его сотрудников было суждено стать основой того мира, в котором впоследствии сконцентрировалась деятельность «бунтующей молодежи» 1960-х. Современные хакеры (не компьютерные хулиганы из газет, а настоящие увлеченные своим делом компьютерщики) — прямые идеологические наследники сорбонских студентов и американских демонстрантов, сменившие девиз «Make love not war»<sup>1</sup> на «Не пишите лозунги — пишите код». Неслучайно многие известные деятели электронно-компьютерной индустрии, авторы изобретений, сформировавших лицо современного мира, — выходцы из среды, близкой той самой «бунтующей молодежи».

Наша история о микроконтроллерах началась с того, что в 1957 г. Гордон Мур совместно с Робертом Нойсом, ставшим впоследствии одним из изобретателей микросхемы, и еще шестью сотрудниками Shockley Semiconductor Labs (Шокли назвал их «предательской восьмеркой»), основал компанию Fairchild Semiconductor. Ей мы обязаны не только развитием полупроводникового рынка и внедрением микросхем в инженерную практику, но и тем, что она стала своеобразной кузницей кадров и генератором идей для молодой отрасли.

Вот только некоторые из исторических фактов. Сам Мур с Нойсом в конце 1960-х создали фирму Integrated Electronics, которая под сокращенным названием Intel сейчас знакома каждому школьнику. Джереми Сандерс, основатель другой известнейшей компании — AMD, также вышел из Fairchild, где отличился открытием современной экономической модели производства и продаж полупроводниковых компонентов, в которой себестоимость изделия стремится к нулю по мере повыше-

---

<sup>1</sup> «Занимайтесь любовью, а не войной» — лозунг хиппи 1960-х, протестующих против войны во Вьетнаме.

ния объема партии. Чарли Спорк, один из ключевых менеджеров Fairchild, в 1967 г. стал директором National Semiconductor, которой впоследствии руководил четверть века. Половина «предательской восьмерки» — Джин Хоерни, Евгений Клайнер, Джей Ласт и Шелдон Робертс — в 1961 г. основала компанию Amelco, из которой впоследствии выросли всем известные теперь Intersil, Maxim и Ixys. Сотруднику Fairchild Роберту Видлару мы обязаны изобретением операционных усилителей — разновидности микросхем, и по сей день уступающей по популярности разве что микропроцессорам. Мало того, с историей Fairchild связано возникновение известной венчурной (т. е. «рисковой») модели финансирования, сыгравшей определяющую роль в развитии всех отраслей, ориентированных на электронику, компьютеры и телекоммуникации. Недаром Fairchild нередко называют «праматерью всей электроники».

## Предыстория микроконтроллеров

Из всего этого урагана событий для нашего повествования важно то, что в числе прочих инноваций сотрудники Fairchild первыми стали продвигать полупроводниковую память. Сейчас, в век CD и DVD, жестких дисков и flash-карточек, нам трудно представить себе, что в начале 1960-х годов программы для компьютеров хранились в основном на картонных листочках (перфокартах), конструкторы ломали голову над дорогушими модулями ОЗУ на ртутных линиях задержки, осциллографических трубках и ферритовых колечках, где каждый бит «прошивался» вручную. Самое компактное в те годы электронное устройство для хранения данных на магнитных дисках под названием RAMAC 305 емкостью 5 Мбайт было размером с промышленный холодильник и сдавалось в аренду за 5 тыс. долларов в месяц.

Единственным «лучом света» в темном царстве этих монстров стало изобретение сотрудника корпорации American Bosch Arma Йен Чоу, который в 1956 г. получил патент на устройство, известное теперь как «однократно программируемое ROM» (OTP ROM<sup>1</sup>). В этом патенте, между прочим, впервые был употреблен термин «прожиг» (burn) — микромодуль состоял из матрицы с плавкими перемычками, которые при программировании пережигались подачей на них большого напряжения. OTP ROM долгое время оставались единственными устройствами для компактного хранения данных, и не потеряли своего значения до самого последнего времени — не меньше четверти микроконтроллеров в мире, особенно из тех, что попроще, до сих пор выпускается именно с такой однократно программируемой встроенной памятью, ввиду крайней ее дешевизны. И лишь в последние годы «прожигаемая» память стала постепенно вытесняться более удобной flash-памятью, когда последняя подешевела настолько, что смысл в использовании OTP ROM почти пропал.

Но вернемся в 1960-е. Компактная полупроводниковая память была нужна абсолютно всем — от военных и NASA до изготовителей бытовых приборов. Сначала Fairchild предложила то, что сегодня называется DRAM, в частности, на таких мик-

<sup>1</sup> Расшифровку некоторых аббревиатур см. в *приложении 5*.

росхемах (32 768 чипов емкостью 256 бит каждый) была построена память знаменитого суперкомпьютера ILLIAC-IV, конкурента отечественной БЭСМ-6. Почувяв, откуда дует ветер, в 1968 г. Мур с Нойсом оставили Fairchild и основали Intel, как специализированную компанию по разработке и производству памяти. Они еще не ведали, что самым популярным детищем Intel станет вовсе не память, а небольшой приборчик (рис. В1), названный микропроцессором, разработка которого первоначально затевалась как вспомогательный этап в проектировании обычного калькулятора.

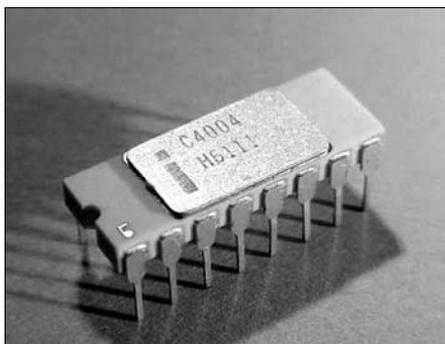


Рис. В1. Микропроцессор Intel 4004

### **ИЗОБРЕТЕНИЕ МИКРОПРОЦЕССОРА**

В 1969 г. в Intel появились несколько человек из Busicom — молодой японской компании, занимающейся производством калькуляторов. Им требовался набор из 12 интегральных схем в качестве основного элемента нового дешевого настольного калькулятора. Проект был разработан Масатоши Шима, который и представлял японскую сторону. Тед Хофф (Marcian E. Ted Hoff, р. 1937 г.), руководитель отдела, занимавшегося разработкой применений для продукции Intel, ознакомившись с проектом, понял, что вместо того, чтобы создавать калькулятор с некоторыми возможностями программирования, можно сделать наоборот, компьютер, программируемый для работы в качестве калькулятора. Развивая идею, в течение осени 1969 г. Хофф определился с архитектурой будущего микропроцессора. Весной в отдел Хоффа пришел все из той же уже известной нам Fairchild) новый сотрудник Фредерик Фэггин (Federico Faggin), который и придумал название для всей системы: «семейство 4000». Семейство состояло из четырех 16-выводных микросхем: 4001 содержал ROM на 2 Кбайта; 4002 — RAM с 4-разрядным выходным портом для загрузки программ; 4003 представлял собой 10-битовый расширитель ввода-вывода с последовательным вводом и параллельным выводом для связи с клавиатурой, индикатором и другими внешними устройствами; наконец 4004 был 4-битовым ЦПУ (центральным процессорным устройством). Это ЦПУ содержало 2300 транзисторов и работало на тактовой частоте 108 кГц. 15 ноября 1971 г. было объявлено о создании первого микропроцессора. Busicom приобрела разработку, заплатив Intel \$60 000. Но в Intel решили вернуть Busicom эти деньги, чтобы вернуть себе права на микропроцессор.

По вычислительной мощности i4004 был сравним с первым электронным компьютером ENIAC (1946). Свое первое практическое применение 4004-й нашел в системах управления дорожными светофорами и анализаторах крови. Этот микропроцессор был использован в бортовой аппаратуре межпланетного зонда Pioneer-10, который поставил рекорд долгожительства среди подобных аппаратов: запущенный в 1972 г. к сентябрю 2001 г. Pioneer-10 удалился от Земли на 11,78 млрд км и все еще работает и, вполне вероятно, работает по сей день, хотя в феврале 2003 г. NASA официально с ним попрощалась.

Так началось победное шествие микропроцессоров, которые позднее разделились на несколько разновидностей, в основном относящихся к двум главным группам: собственно микропроцессорам (МП) и микроконтроллерам (МК). Первые предназначены для использования в составе вычислительных систем, самые распространенные из которых — персональные компьютеры (ПК), поэтому их еще часто называют «процессорами для ПК» (хотя к этой группе обычно относят также и производительные МП для серверов и некоторые другие). МК отличаются от МП тем, что они в первую очередь предназначены для управления различными системами, поэтому при относительно более слабом вычислительном ядре они включают в себя много дополнительных узлов. То, что для обычного МП предполагается размещать во внешних чипсетах или дополнительных модулях (память, порты ввода-вывода, таймеры, контроллеры прерываний, узлы для обработки аналоговых сигналов и пр.), в МК располагается прямо на кристалле, отчего их первоначально было принято называть «computer-on-chip» («однокристалльный компьютер»).

И действительно, в простейшем случае для построения полностью функционирующего компьютера достаточно единственной микросхемы МК с подсоединенными к ней устройствами ввода-вывода. Современные модели рядовых однокристалльных МК превышают вычислительные возможности IBM PC AT на 286-м процессоре образца второй половины 1980-х. Есть области, где границу между МП и МК провести трудно: таковы, например, процессоры для мобильных устройств, от телефонов и карманных компьютеров до цифровых камер, в которых процессорный узел должен обладать развитыми вычислительными функциями и управлять многочисленными внешними компонентами.

## Электроника в греческом стиле

В 1962 г. в Калифорнии появилась семья Перлегос, греческих эмигрантов, уроженцев города Триполис. Родители занялись, как и на родине, виноградарством, а сыновья, Джордж и Гюст Перлегос, выбрали модную специальность инженера-электронщика: оба окончили вначале университет Сан-Хозе, а затем Стэнфордский университет. В 1974 г. в возрасте 24 лет младший из братьев Джордж Перлегос начал работать в компании Intel, где попал на одно из самых передовых направлений: разработку электрически стираемой памяти для замены «прожигаемой» OTP ROM. Еще до Перлегоса, почти одновременно с изобретением микропроцессора в 1971 г., сотрудник Intel Дон Фрохман изобрел «плавающий» затвор и создал первую УФ-стираемую EPROM объемом 2К (256×8).

### **ЗАМЕТКИ НА ПОЛЯХ**

В «обычной» жизни употреблять сокращение для единиц информации из одной буквы «К» (так же, как и «М») не рекомендуется: очень трудно иногда понять, идет ли речь о килобитах, килобайтах, «килословах» или вообще килобитах в секунду. Тем не менее такие сокращения часто встречаются, в том числе и в технической документации, и нам придется иногда следовать этому примеру. Для определенности примем следующие правила: одиночная прописная буква «К» означает двоичные килобиты (1024 бита), «М» — двоичные мегабиты (1024 кбита). Хотя в литературе часто еще принято килобайты сокращать, как «КБ», а килобиты, как «Кб», мы постараемся избежать этой пу-

таницы, и во всех остальных случаях писать полностью: кбайт и Мбайт, кбит/с, Мбайт/с. Исключение — обозначение объема памяти программ микроконтроллеров, если он измеряется в двухбайтовых словах: например, 4К слов будет обозначать 4096 ячеек-слов (8 кбайт).

Джордж Перлегос активно включился в этот процесс и сначала при его участии, а затем и под его непосредственным руководством были созданы две технологии, ставшие точкой роста для всей отрасли по производству flash-памяти — одного из главных столпов современной «цифровой революции». Это было сначала изобретение чипа 2716 — 16К (2048×8) EPROM с одним напряжением питания +5 В, а затем 2816 — первой EEPROM, электрически стираемого ПЗУ, ставшего прообразом flash-памяти.

В 1981 г. Перлегос покидает Intel и с несколькими сотрудниками (в числе которых был Гордон Кэмпбелл, будущий основатель другой известной фирмы Chips & Technologies) создает компанию Seeq. Это было время спада в электронной промышленности и через три года компанию пришлось покинуть в связи с претензиями инвесторов. Не доверяя им больше, Джордж с братом Гюстом и еще несколькими сотрудниками Seeq в 1984 г. создает в складчину на личные средства компанию, полное название которой звучит как Advanced Technology Memory and Logic или сокращенно — Atmel.

Сначала продукцией Atmel были микросхемы энергонезависимой памяти всех разновидностей — как OTP EPROM и EEPROM с последовательным и параллельным доступом, так и flash. В 1985 г. Atmel выпустила первую в мире EEPROM по доминирующей ныне КМОП-технологии, а в 1989 г. — первую flash-память с питанием от одного напряжения +5 В. В конце 1980-х Intel вознамерилась наказать ряд компаний-производителей EPROM, в том числе и Atmel, якобы за нарушение патентов, но в конце концов удалось договориться об обмене лицензиями. Причем в конечном итоге Atmel перепала лицензия на производство классического микроконтроллера 8051, от поддержки которого Intel уже в то время постепенно отходила, сосредоточившись на процессорах для ПК.

### **ПОДРОБНОСТИ**

Напомним, что EEPROM отличается от flash-памяти тем, что первая допускает отдельный доступ к любой произвольной ячейке, а вторая — лишь к целым блокам. Поэтому EEPROM меньше по объему (характерный объем специализированных микросхем EEPROM — от единиц килобит до единиц мегабит) и дороже; в настоящее время ее используют в основном для хранения данных, в том числе в составе микроконтроллеров. Flash-память проще и дешевле, и к тому же дает значительный выигрыш в скорости при больших объемах информации, особенно при потоковом чтении/записи, характерном для медиаустройств (вроде цифровых камер или MP3-плееров). В составе микроконтроллеров flash-память служит для хранения программ. Некоторые подробности о различных типах памяти и их функционировании см. в *главе 11*.

Так Atmel оказалась «втянута» в число производителей микроконтроллеров и очень быстро заняла передовые позиции: в 1993 г. началось производство первых в отрасли МК AT89C51 со встроенной flash-памятью программ. Это означало начало переворота во всей инженерной практике, потому что существовавшие ранее МК обладали либо однократно программируемой OTP-памятью, либо УФ-стираемой,

которая значительно дороже в производстве и работа с ней приводит к большим потерям времени разработчиков. Число циклов перезаписи для УФ ППЗУ не превышает нескольких десятков, а прямой дневной свет, попавший на такой кристалл, может привести к стиранию информации. Поэтому даже мелкосерийные устройства приходилось изготавливать преимущественно с использованием OTP ROM, что значительно рискованнее: изменить в случае даже малейшей ошибки записанную программу уже было невозможно. Появление flash-памяти изменило весь «ландшафт» в этой области: именно в результате ее внедрения стали возможными такие вещи, как программное обновление BIOS компьютера или «перешивка» управляющих программ для бытовых электронных устройств.

В 1995 г. два студента Норвежского университета науки и технологий в г. Тронхейме, Альф Боген и Вегард Воллен, выдвинули идею 8-разрядного RISC-ядра, которую предложили руководству Atmel. Имена разработчиков вошли в название архитектуры AVR: Alf + Vergard + RISC. Идея настолько понравилась, что в 1996 г. был основан исследовательский центр Atmel в Тронхейме и уже в конце того же года выпущен первый опытный микроконтроллер новой серии AVR под названием AT90S1200. Во второй половине 1997 г. корпорация Atmel приступила к серийному производству семейства AVR.

## Почему AVR?

У AVR-контроллеров «с рождения» есть две особенности, которые отличают это семейство от остальных МК. Во-первых, система команд и архитектура ядра AVR разрабатывались совместно с фирмой-разработчиком компиляторов с языков программирования высокого уровня IAR Systems. В результате появилась возможность писать AVR-программы на языке C без большой потери в производительности по сравнению с программами, написанными на ассемблере. Подробнее этот вопрос мы обсудим в *главе 5*.

Во-вторых, одним из существенных преимуществ AVR стало применение конвейера. В результате для AVR не существует понятия машинного цикла: большинство команд выполняется за один такт. Для сравнения отметим, что пользующиеся большой популярностью МК семейства PIC выполняют команду за 4 такта, а классические 8051 — вообще за 12 тактов (хотя есть и современные модели x51 с машинным циклом в один такт).

Правда, при этом пришлось немного пожертвовать простотой системы команд, особенно заметной в сравнении с x51, где, например, любые операции пересылки данных внутри контроллера, независимо от способа адресации, выполняются единственной командой `mov` в различных вариантах, в то время как в AVR почти для каждого способа есть своя команда, к тому же иногда с ограниченной областью действия. Существуют некоторые сложности и при операциях с битами. Тем не менее это не приводит к заметным трудностям при изучении AVR-ассемблера: наоборот, тексты программ получаются короче и больше напоминают программу на языке высокого уровня. Следует также учесть, что из общего числа команд (от 90 до 130, в зависимости от модели) только 50–60 уникальных, остальные взаимоза-

няемые. И наконец, этот недостаток полностью нивелируется при использовании языка C, фактически уравнивающего разные архитектуры с точки зрения особенностей программирования.

Огромное преимущество AVR-архитектуры — наличие 32 оперативных регистров, не совсем равноправных, но позволяющих в ряде случаев вообще не обращаться к оперативной памяти и к стеку (что в принципе невозможно в том же семействе x51), более того, в младших моделях AVR стек вообще недоступен для программиста. Потому структура ассемблерных программ для AVR стала подозрительно напоминать программы на языке высокого уровня, где операторы работают не с ячейками памяти и регистрами, а с абстрактными переменными и константами.

Еще одна особенность AVR со схемотехнической точки зрения — все выводы в них могут пребывать в трех состояниях (вход — отключено — выход) и электрически представляют собой КМОП-структуры (т. е. имеет место симметрия выходных сигналов и высокое сопротивление для входных). В общем случае это значительно удобнее портов x51 (двустабильных и TTL-совместимых) и предполагает лучшую помехозащищенность (по крайней мере, от помех по шине «земли»).

Суммировав мнения из различных источников и опираясь на собственный опыт, автор пришел примерно к такому подразделению областей применения трех самых распространенных семейств контроллеров.

- Контроллеры классической архитектуры x51 (первые микросхемы семейства 8051 были выпущены еще в начале 1980-х) лучше всего подходят для общего изучения предмета. Отметим, что кроме Atmel, x51-совместимые изделия выпускают еще порядка десятка фирм, включая такие гиганты, как Philips и Siemens, есть и отечественные аналоги (серии 1816, 1830 и др.), что делает эту архитектуру наиболее универсальной.
- Семейство AVR рекомендуется для начинающих электронщиков-практиков, в силу простоты и универсальности устройства, преемственности структуры для различных типов контроллеров, простоты схемотехники и программирования (в данном случае под «программированием» понимается процесс записи программ в микросхему).
- PIC фирмы Microchip идеально подходят для проектирования несложных устройств, особенно предназначенных для тиражирования.

Эта классификация во многом субъективна, и автор не будет оспаривать другие точки зрения: различные семейства МК постепенно сближаются по параметрам, становятся полностью взаимозаменяемыми и, как и во всей современной электронике, выбор того или иного семейства часто носит характер «религиозного».

Кроме 8-разрядных МК AVR, Atmel выпускает еще несколько разновидностей МК, к которым относятся не только упомянутые наследники 8051, но и ARM-процессоры, а также специализированные МК для различных применений. Среди МК со встроенной flash-памятью Atmel принадлежит треть мирового рынка, а российский рынок универсальных 8-разрядных МК Atmel практически поделила с Microchip, хотя в мире они не входят в число лидеров сегмента 8-разрядных микроконтрол-

леров, уступая таким производителям, как ST, Renesas, Freescale или Texas Instruments.

С середины 2000-х микроконтроллеры Atmel оказались в центре нового течения в разработке — в среде электронщиков появился аналог явления, давно известного в программировании под названием Open Source. Течение возникло около 2004 г. в среде сотрудников Interaction Design Institute (что можно перевести, как «Институт конструирования взаимодействий»), находящегося в итальянском городке Ивреа. Оно выросло из задачи научить студентов создавать электронные устройства, причем быстро и, желательно, без опоры на углубленное изучение электроники, электротехники и программирования. Группа инженеров и программистов, руководимая Массимо Банци, создала универсальную аппаратную платформу на основе дешевых и доступных микроконтроллеров Atmel AVR и решила ее распространять на принципах open source. Свое почти толкиеновское название — Arduino — платформа получила от имени реально существовавшего короля Ардуина, правившего местностью вокруг Ивреа в начале прошлого тысячелетия.

Существует более десяти модификаций базовых плат одной только оригинальной Arduino (не считая многочисленных клонов — Freeduino, Craftduino, Carduino и др.). В основе почти всех из них лежат 8-разрядные МК AVR: в старых версиях это был ATmega8, в новых применяется в основном ATmega328, но есть и версии с ATmega168, ATmega2560 и др. (исключение составляют две «продвинутые» модификации на основе 32-разрядных ARM-процессоров). Плата Arduino состоит из упомянутого микроконтроллера, а также обычно содержит стабилизатор питания и элементы обвязки для программирования и интеграции с другими схемами. В микроконтроллер предварительно прошивается загрузчик BootLoader, поэтому внешний программатор оказывается не нужен. Для Arduino создана специальная среда программирования (основанная на адаптированной под непрофессионалов версии C/C++ с названием Processing), хотя эти платы можно программировать и напрямую на ассемблере, как в примерах в этой книге. Результатом всей этой деятельности стала растущая популярность Arduino среди любителей электроники и огромное количество всяческих конструкторов-наборов, ориентированных именно на нее. Arduino стал самой известной аппаратной системой нашего времени с открытым исходным кодом. Имеющиеся в продаже конструкторы для создания достаточно сложных электронных самоделок, за исключением тех, что производятся фирмой Lego, скорее всего, будут тем или иным боком связаны с Arduino. На основе Arduino любители создали множество традиционных конструкций роботов-игрушек, в том числе и такие сложные устройства, как робот, балансирующий на шаре, или механическая рука, повторяющая движения человеческой кисти. Есть проект, посвященный управлению пылесосами iRobot Roomba с помощью Arduino. Есть и серьезные профессиональные разработки, такие как анализатор ДНК или 3d-принтер. Интерес к Arduino повысила Google, которая в 2011 г. создала вспомогательный комплект разработчика Android ADK, позволяющий смартфону взаимодействовать с устройствами на Arduino. Вся эта деятельность открыла перед платформой Atmel новые горизонты на будущее.

## Что дальше?

Эта книга адресована читателю, который хочет изучить структуру и схемотехнические особенности МК AVR и научиться грамотно использовать их основные возможности. Поэтому автор ограничивается языком ассемблера (подробнее вопрос выбора среды программирования мы рассмотрим в *главе 5*). Упор в книге делается на то, чтобы дать читателю практические советы, описать готовые алгоритмы для типовых задач, возникающих перед разработчиками при реализации тех или иных функций МК. Автор вместе с читателями подробно разбирает ряд вопросов, которые обычно выходят за рамки пособий по программированию МК: работу с последовательными интерфейсами, арифметические операции, сопряжение с ПК, практическую реализацию режимов энергосбережения.

Вместе с тем, автор не ставил задачу разобрать подробно абсолютно все возможности МК AVR: для этого не хватило бы и нескольких томов. Мы вынуждены обойти такие вопросы, как отладка и программирование по интерфейсу JTAG, перспективы, которые открывает самопрограммирование контроллеров, лишь вскользь коснемся интереснейшей задачи синтеза звука и других применений PWM-режимов таймеров.

В этой книге большинство примеров ориентировано на применение младших (с объемом памяти 8 Мбайт) моделей подсемейства Mega, т. к. именно они наиболее универсальны и пригодны для широкого круга задач без излишнего усложнения схемы. В самых простых случаях автор ориентировался на наиболее универсальную из младших моделей ATtiny2313 (о ее совместимости с «классической» версией AT90S2313 см. *разд. «Программа счетчика с использованием прерываний» главы 5*). Большинство приведенных примеров можно практически без переделок адаптировать к другим моделям AVR, обладающим соответствующей конфигурацией.

Книга не заменяет фирменные справочники по структуре и системе команд конкретных моделей AVR — их число постоянно растет, некоторые снимаются с производства, другие приходят на их место. Поэтому автор старался давать максимально обобщенные примеры, которые, по возможности, пригодны для большинства существующих моделей. Необходимое дополнение к этой книге — справочник А. В. Евстифеева [1, 2], где на русском языке собраны фирменные технические описания для большинства моделей AVR и, в частности, приведена полная таблица команд AVR-ассемблера с подробным формальным описанием каждой из них<sup>1</sup>.

Читатели, хорошо владеющие техническим английским, смогут обойтись англоязычными описаниями (так называемый Data Sheets) конкретных моделей, которые можно скачать с сайта Atmel.com. Знакомство с этими материалами так или иначе

---

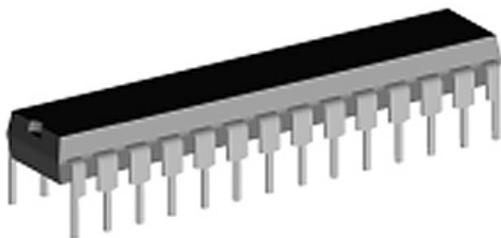
<sup>1</sup> Справочник А. В. Евстифеева переиздавался множество раз. Автор рекомендует именно указанные в списке литературы издания 2007 г. (раздельные для Tiny и Mega), потому что общий для обеих серий справочник того же автора и издательства (5-е стереотипное издание, 2008 г.) содержит некоторые ошибки и упущения, не только унаследованные от первого издания, но и добавленные позже.

потребуется, т. к. в построении отдельных моделей AVR слишком много нюансов, делающих их в некоторых частных случаях незаменимыми, и все алгоритмы применительно к конкретной ситуации следует проверять. Рекомендации по применению (Application Notes) Atmel также полезны для изучения, но они, к сожалению, довольно отрывочны и не охватывают всего круга задач. Примеры законченных устройств читатель может найти в источниках [3–8, 10]. В книге автора [8] приведены элементарные сведения по логическим элементам, системам счисления и другие азы микроэлектроники для тех, кто в этом не ориентируется.

Тем, кто уже знаком с семейством AVR, разумеется, можно читать настоящую книгу выборочно, пользуясь ею, как справочником. Всем остальным автор советует хотя бы один раз изучить ее подряд, глава за главой, иначе будет трудно осознанно применить разрозненные сведения из отдельных глав. В третьем издании книги исправлены замеченные ошибки и принята общая ориентация на современные модели семейств Mega и Tiny. Примеры для давно снятого с производства семейства Classic приводятся в виде исключения, с учетом того, что в различных источниках можно найти еще много таких программ.

Схемы, рисунки и фотографии выполнены автором. Все остальные иллюстрации взяты из источников, допускающих свободное копирование. Автор выражает благодарность Юрию Певзнеру за консультации и предоставленные тексты программ для микроконтроллеров семейства Atmel AVR.

Пожелания, вопросы и указания на неточности можно направлять по адресу: [revich@lib.ru](mailto:revich@lib.ru).



# ЧАСТЬ I

## Общие принципы устройства и функционирования Atmel AVR

- Глава 1. Обзор микроконтроллеров Atmel AVR
- Глава 2. Общее устройство, организация памяти, тактирование, сброс
- Глава 3. Знакомство с периферийными устройствами
- Глава 4. Прерывания и режимы энергосбережения



# ГЛАВА 1



## Обзор микроконтроллеров Atmel AVR

Atmel AVR представляет собой семейство универсальных 8-разрядных микроконтроллеров на основе общего ядра с различными встроенными периферийными устройствами. Возможности МК AVR позволяют решить множество типовых задач, возникающих перед разработчиками радиоэлектронной аппаратуры.

Особенности микроконтроллеров Atmel AVR.

- **Производительность порядка 1 MIPS/МГц.** MIPS (Millions of Instructions Per Second, миллион команд в секунду) — старая и во многом формальная характеристика производительности процессоров, т. к. наборы команд для различных процессоров различаются, и, соответственно, одно и то же число инструкций на различных системах даст разную полезную работу. Тем не менее для простых 8-разрядных вычислительных систем, не содержащих команд, оперирующих с большими числами, числами с плавающей точкой и массивами данных, это неплохой показатель для сравнения их производительности. Вычислительное ядро AVR на ряде задач по производительности превосходит 16-разрядный процессор 80286.
- **Усовершенствованная RISC-архитектура.** Концепция RISC (Reduced Instruction Set Computing, вычисления с сокращенным набором команд) предполагает наличие набора команд, состоящего из минимума компактных и быстро выполняющихся инструкций; при этом такие более громоздкие операции, как вычисления с плавающей точкой или арифметические действия с многоразрядными числами, предполагается реализовать на уровне подпрограмм. Концепция RISC упрощает устройство ядра (в типовом ядре AVR содержится лишь 32 тыс. транзисторов, в отличие от десятков миллионов в процессорах для ПК) и ускоряет его работу: типовая инструкция выполняется за один такт (кроме команд ветвления программы, обращения к памяти и некоторых других, оперирующих с данными большой длины). В AVR имеется простейший двухступенчатый конвейер, выполняющий команду в одном такте с выборкой следующей. В отличие от Intel-архитектур, в «классическом» AVR нет аппаратного умножения/деления, однако в подсемействе Mega присутствуют операции умножения.
- **Раздельные шины памяти команд и данных.** AVR (как и большинство других микроконтроллеров) имеет так называемую *гарвардскую архитектуру*, где об-

ласти памяти программ и данных разделены (в отличие от классической архитектуры фон Неймана в обычных компьютерах, где память общая). Раздельные шины для этих областей памяти значительно ускоряют выполнение программы: данные и команды могут выбираться одновременно.

- **32 регистра общего назначения (РОН).** Atmel была первой компанией, далеко отошедшей от классической модели вычислительного ядра, в которой выполнение команд предусматривает обмен данными между АЛУ и запоминающими ячейками в общей памяти. Введение РОН в таком количестве (напомним, что в архитектуре x86 всего четыре таких регистра, а в x51 понятие РОН, как таковое, отсутствует) в ряде случаев позволяет вообще отказаться от расположения глобальных и локальных переменных в ОЗУ и от использования стека, операции с которым усложняют и загромождают программу. В результате структура ассемблерной программы приближается к программам на языках высокого уровня. Правда, это привело к некоторому усложнению системы команд, номенклатура которых для AVR больше, чем в других RISC-семействах (хотя значительная часть инструкций — псевдонимы).
- **Flash-память программ** (10 000 циклов стирание/запись) с возможностью внутрисистемного перепрограммирования и загрузки через последовательный канал прямо в готовой схеме. О преимуществах такого подхода, ныне ставшего общепринятым, подробно рассказано во *введении*.
- **Отдельная область энергонезависимой памяти (EEPROM, 100 000 циклов стирание/запись)** для хранения данных, с возможностью записи программным путем, или внешней загрузки через SPI-интерфейс.
- **Встроенные устройства для обработки аналоговых сигналов:** аналоговый компаратор и многоканальный 10-разрядный АЦП.
- **Сторожевой таймер,** позволяющий автоматически перезагружать контроллер через определенные промежутки времени (например, для выхода из «спящего» режима).
- **Последовательные интерфейсы SPI, TWI (I<sup>2</sup>C) и UART (USART),** реализующие аппаратный обмен данными с большинством стандартных датчиков и других внешних устройств (в том числе таких, как персональные компьютеры).
- **Таймеры-счетчики** с предустановкой и возможностью выбора источника счетных импульсов: как правило, один-два 8-разрядных и как минимум один 16-разрядный, в том числе поддерживающие режим многоканальной 8-, 9-, 10-, 16-битовой широтно-импульсной модуляции (PWM).
- **Возможность работы при тактовой частоте** от 0 Гц до 16–20 МГц.
- **Диапазон напряжений питания от 2,7 до 5,5 В** (в некоторых случаях от 1,8 или до 6,0 В).
- **Многочисленные режимы энергосбережения,** отличающиеся числом узлов, остающихся подключенными. Выход из «спящих» режимов по сторожевому таймеру или по внешним прерываниям.

□ **Встроенный монитор питания** — детектор падения напряжения (Brown-out Detection).

Здесь перечислены далеко не все особенности, характерные для различных моделей AVR. С некоторыми другими мы познакомимся в дальнейшем, а также на практике рассмотрим перечисленные подробнее. Но сначала дадим общую характеристику различных семейств AVR с точки зрения их преимущества назначения.

## Семейства AVR

В 2002 г. фирма Atmel начала выпуск новых подсемейств 8-разрядных МК на базе AVR-ядра. С тех пор все МК этого семейства делятся на три группы (подсемейства): Classic, Tiny и Mega. В 2008 г. к ним добавилось семейство X Mega. МК семейства Classic (AT90Sxxx) уже не выпускаются; дольше всего в производстве «задержалась» очень удачная (простая, компактная и быстродействующая модель) AT90S2313, но и она была в 2005 г. заменена на ATtiny2313. Все «классические» AVR с первыми цифрами 2 и 8 в наименовании модели (означающими объем памяти программ в килобайтах) имеют аналоги в семействах Tiny и Mega. Для Mega при программировании возможна установка специального бита совместимости, который позволяет без каких-либо изменений использовать программы, созданные для семейства Classic. Примеры в этой книге мы по возможности постарались изложить в версиях для новых модификаций МК, но в Интернете «гуляет» еще много примеров программ для семейства Classic, потому основные отличия старых и новых версий приведены в соответствующих разделах.

Примеры различных типов корпусов, в которых выпускаются микросхемы AVR, приведены на рис. 1.1. Более подробную информацию на эту тему можно найти в *приложении 1* (табл. П1.2), а также в технической документации на устройства. Отметим, что для радиолюбительских нужд и макетирования удобнее всего микросхемы в PDIP-корпусах, но в них производятся не все модели МК.

Все семейства могут иметь две модификации: буква «L» в обозначении говорит о расширенном диапазоне питания 2,7–5,5 В, отсутствие такой буквы означает диапазон питания 4,5–5,5 В. При выборе конкретного типа микросхемы нужно быть внимательным, т. к. L-версии одновременно также и менее быстродействующие, у большинства из них максимальная тактовая частота ограничена значением 8 МГц. Для «обычных» версий максимальная частота составляет 16 или 20 МГц. Хотя, как правило, при запуске L-микросхем с напряжением питания 5 В на частотах до 10–12 МГц неприятностей ожидать не следует (аналогично версии без буквы L вполне могут работать при напряжении питания около 3 В, разумеется, не на экстремальных значениях частот), тем не менее при проектировании высоконадежных устройств нужно учитывать это требование.

Микросхемы Tiny имеют Flash-ПЗУ программ объемом 1–8 кбайт и размещаются в основном в корпусах с 8–20 выводами (кроме ATtiny28), т. е. они в целом предназначены для более простых и дешевых устройств. Это не значит, что их возможности во всех случаях более ограничены, чем у семейства Mega. Так, например, ATtiny26 при цене менее 2 долларов содержит таймер с высокоскоростным ШИМ-режимом (в других моделях такого нет), а также 11-канальный АЦП с возмож-

ностью работы в дифференциальном режиме, с регулируемым входным усилителем и встроенным источником опорного напряжения, что характерно для старших моделей. Микросхема ATtiny2313, как уже говорилось, представляет собой улучшенную версию одного из наиболее универсальных и удобных «классических» AVR AT90S2313. Старший в семействе Tiny в настоящий момент МК ATtiny828 с 8 килобайтами памяти программ и корпусом с 32 выводами по некоторым параметрам даже превышает младшие Mega: например, у него уникальное число каналов АЦП — 28.

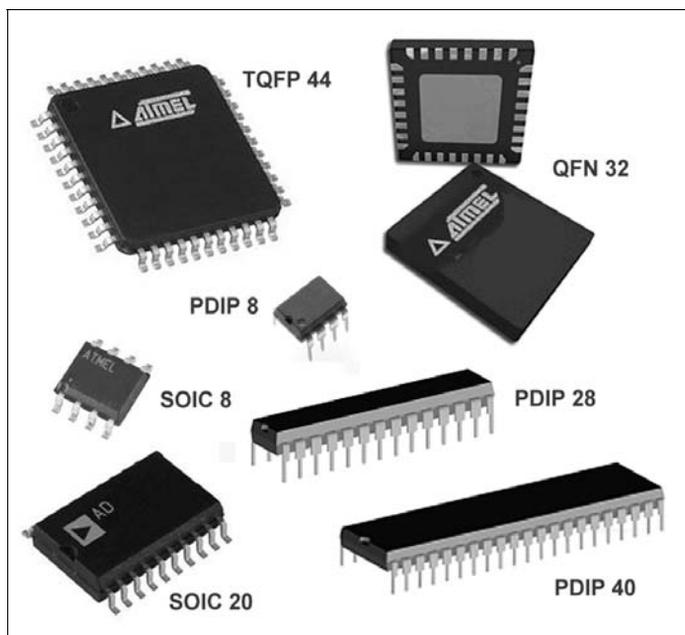


Рис. 1.1. Примеры различных типов корпусов для МК AVR

Подсемейство Mega оснащено Flash-ПЗУ программ объемом 8–256 кбайт и корпусами с 28–100 выводами. В целом МК этой группы более «навороченные», чем Tiny, имеют весьма разветвленную систему встроенных устройств с развитой функциональностью.

Таблицы с основными характеристиками некоторых моделей Tiny и Mega из числа самых ходовых приведены в *приложении 1*. Там же даны некоторые общие технические характеристики семейства AVR. Более подробные сведения можно почерпнуть из [1, 2] и фирменной технической документации, которая доступна на сайте Atmel для каждой модели.

Кроме этих трех семейств, на базе AVR-ядра выпускаются специализированные микросхемы для работы с USB-интерфейсом (AT90USBxxx), промышленным интерфейсом CAN (AT90CANxxx), для управления ЖК-дисплеями (ATmega329 и др.), с беспроводным интерфейсом IEEE 802.15.4 (ZigBee) для предприятий торговли и некоторые другие. В последнее время некоторые микроконтроллеры серий Tiny и Mega стали выпускаться в версиях со сверхмалым потреблением (технология

рiсoPower с напряжением питания от 1,8 В, в конце наименования МК этой серии добавлена буква «Р») и высокотемпературных для использования в автомобильной промышленности (версии Automotive).

Из новых моделей стоит обратить внимание на АТmega32HVE2 и АТmega64HVE2. Они изначально предназначены для мониторинга состояния свинцовых и литий-ионных батарей, однако их с успехом можно применять и в других приложениях. Их особенностями являются, во-первых, регулятор напряжения питания кристалла, позволяющий подключать микросхему напрямую к источнику +12 В, во-вторых, АЦП повышенной точности — 17–18 двоичных разрядов («дельта-сигма» типа, в отличие от обычных, имеющих АЦП последовательного приближения). В дополнение к этому на кристалле размещен 32/64-разрядный математический сопроцессор, выполняющий арифметические и логические операции с 32-разрядными операндами за 1–2 такта, а с 64-разрядными операндами — за 2–6 тактов. Такое построение МК типа HVE2 позволяет в значительной степени упростить выполнение арифметических операций с многоразрядными (в том числе действительными) числами (см. главу 7).

Семейство XМega предназначено для относительно «продвинутых» устройств: контроллеры из этой серии работают при напряжении питания 1,8–3,6 В, обладают повышенным быстродействием (тактовая частота до 32 МГц), 12-разрядным 16-канальным АЦП и 2–4 каналами ЦАП (в других семействах они отсутствуют), несколькими каналами UART и/или USB и других последовательных портов (причем с возможностью работы в автономном режиме, при остановленном ядре), встроенной поддержкой криптографии, усовершенствованным режимом рiсoPower и другими «наворотами» [23]. Существует также отдельное семейство 32-разрядных МК AVR32, предназначенное для высокоскоростных приложений, таких как обработка видеопотока или распознавание образов в реальном времени.

## Особенности практического использования МК AVR

При использовании AVR возникает ряд вопросов практического характера, игнорирование которых может иногда привести к неработоспособности или сбоям устройства (а в некоторых случаях — даже к невозможности его запрограммировать). Например, одна из таких проблем — возможность потери содержимого EEPROM при выключении питания. Эту и подобные проблемы мы подробно рассмотрим в соответствующих главах. О вопросах, связанных с энергопотреблением МК, см. главу 4, раздел «Режимы энергосбережения и потребление». Здесь же остановимся на некоторых общих вопросах включения МК AVR.

У большинства выводов МК имеется встроенный подключаемый «подтягивающий» (т. е. подсоединенный к шине питания) резистор, что, казалось бы, решает одну из обычных схемотехнических проблем, когда наличие такого резистора требуется для подключения двухвыводных кнопок или выходов с «открытым коллектором». Однако в критичных случаях необходим внешний резистор сопротивлением 2–5 кОм (в случае схем, критичных для потребления — 10–30 кОм).

«Подтягивающий» резистор следует устанавливать не только на выводе /RESET (о чем пойдет речь в *главе 2*), но и в том случае, когда выводы SCK, MOSI и MISO соответствующих портов используются для программирования и подключены к программирующему разъему ISP (см. *главу 5*), а также по выводам внешних прерываний, если они задействованы. Если эти выводы не «подтягивать» к напряжению питания дополнительными резисторами (хотя это и не оговорено в технической документации), то не исключены ложные срабатывания внешних прерываний, перезапуск системы, а при очень мощных помехах — даже порча данных в памяти программ. С другой стороны, когда выводы программирования выполняют функцию обычных портов, сконфигурированных на выход, а в устройстве применяются режимы энергосбережения, наличие «подтягивающих» резисторов может привести к лишнему потреблению тока (при установке вывода в логический ноль через резистор потечет ток от источника питания на вход МК). Если реализован один из режимов энергосбережения, то нужно тщательно проанализировать схему, чтобы исключить ситуации, при которых через эти резисторы протекает ток.

Также всегда следует устанавливать внешние резисторы при работе выводов МК на общую шину, как в интерфейсе I<sup>2</sup>C (или просто при подсоединении входа МК к выходу другого устройства с открытым коллектором, например, мониторов питания, описанных в *главе 3*), при подключении к двухвыводным кнопкам (особенно при наличии внешнего прерывания, см. *главы 4 и 5*). Сопротивление встроенного резистора (на самом деле представляющего собой, разумеется, полевой транзистор) в таких случаях слишком велико для того, чтобы электромагнитные помехи («наводки») на нем эффективно «садились».

Все без исключения МК AVR работают при напряжении питания в диапазоне 4,5–5,5 В, обеспечивая при этом заданное быстродействие. Максимально допустимое напряжение составляет 6 В и превышать его не рекомендуется. Для подавляющего большинства моделей нижний порог питания составляет 2,7 В, для многих он еще ниже — 1,8 или даже 1,7 В. Однако при выборе напряжения источника нужно быть внимательным: как правило, при пониженном питании максимальная тактовая частота тоже оказывается сниженной, потому следует ориентироваться не только на цифры в таблице характеристик на сайте Atmel, но и заглянуть в документацию. Например, для ATtiny2313, который мы будем часто упоминать далее, максимальная тактовая частота равна 20 МГц, однако это действительно только в диапазоне напряжения 4,5–5,5 В. При минимально допустимом напряжении 1,8 В возможная частота составляет всего 4 МГц, а при 2,7 В — 10 МГц. Кроме того, следует обращать внимание на букву в конце наименования модели: так, ATmega8535 работает лишь в диапазоне питающего напряжения 4,5–5,5 В (частота до 16 МГц), а для диапазона 2,7–5,5 В (частота до 8 МГц) нужна модель ATmega8535L.

Микросхемы AVR, как и всякая КМОП-логика, благодаря высокому порогу срабатывания эффективно защищены от помех по шине «земли»<sup>1</sup>. Однако они сильно

<sup>1</sup> Как и в других книгах автора (например, [8]), здесь термин «земля» в кавычках означает то же самое, что «общий провод» (обозначается в нашей литературе значком  $\perp$ , а в западной треугольником  $\blacktriangledown$ ). Его следует отличать от заземления («настоящей земли» в электротехническом смысле), обозначаемом значком  $\perp$ .

подвержены помехам по шине питания. Поэтому не забывайте о развязывающих конденсаторах, которые нужно устанавливать непосредственно у выводов питания (керамические 0,1–0,5 мкФ), а также про качество сетевых выпрямителей и стабилизаторов.

Принадлежность к семейству КМОП сказывается и еще в одном отношении: МК AVR плохо переносят изменение полярности питания. По входу КМОП всегда (за исключением некоторых специальных кристаллов) имеются защитные диоды, через которые при неправильном подключении питания начинает течь ничем не ограниченный прямой ток. Опыт показывает, что, в зависимости от мощности источника, МК могут выдержать такой режим на протяжении нескольких секунд, но потом бесповоротно «сгорают». Зато выходы AVR легко выдерживают перегрузки и режим КЗ в течение длительного времени — как и любая КМОП-логика, они переходят в режим ограничения тока.

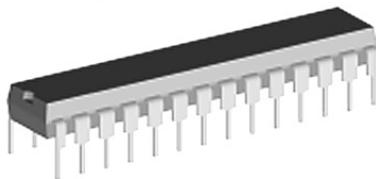
### **ЗАМЕТКИ НА ПОЛЯХ**

Из-за этого свойства к выходам AVR иногда напрямую — без токоограничивающего резистора — подключают светодиоды, однако я бы такое рекомендовать не стал. Стандартная КМОП-логика (серии 4000В) такое «издевательство» еще выдерживает, но быстродействующая (серий АС и НС, а также МК AVR) работает гораздо хуже. При напряжении 2,7–3 В ток короткого замыкания выхода AVR (порядка 30 мА) будет близок к предельно допустимому для светодиодов, а при питании 5 В возрастет до 70–80 мА и значительно превысит любые нормы для обычных светодиодов в корпусах 3–5 мм диаметром. Кроме того, большинство светодиодов выдают нормальную яркость уже при токах около 3–5 мА, а в режиме прямого подключения, даже если ток и не превышает максимально допустимого, будут сверкать, как боевой лазер. Потому такой режим не только опасен для сохранности светодиодов, но и просто нецелесообразен.

При проектировании устройств на МК AVR необходимо помнить про общее ограничение тока через выводы питания, которое для абсолютного большинства моделей составляет 200 мА. Для многоконтактных корпусов это может вырасти в серьезную проблему: например, в модели ATmega640/1280/2560 среди 100 контактов 86 выводов портов общего назначения, каждый из которых можно нагрузить отдельно от других. Зная хорошие нагрузочные свойства этих выводов, возникает соблазн применить их непосредственно для управления какими-нибудь матричными светодиодными табло или шаговыми двигателями. Но при этом нужно всегда тщательно подсчитать суммарный ток через вывод питания, чтобы не превысить допустимое значение. Расчет показывает, что всего двух-трех таких выводов, оказавшихся в режиме КЗ, хватит для превышения максимального значения — наверное, с первого раза МК не сгорит, но при систематическом превышении общего тока едва ли проработает долго.



## ГЛАВА 2



# Общее устройство, организация памяти, тактирование, сброс

Все основные компоненты МК AVR (за исключением модуля JTAG) показаны на общей структурной схеме (рис. 2.1). В отдельных моделях некоторые составляющие могут отсутствовать или различаться по характеристикам, неизменным остается только общее 8-разрядное процессорное ядро (GPU, General Processing Unit). Кратко опишем наиболее важные компоненты, большинство из которых мы подробно будем рассматривать в дальнейшем.

Начнем с памяти. В структуре AVR имеются три разновидности памяти: flash-память программ, ОЗУ (SRAM) для временных данных и энергонезависимая память (EEPROM) для долговременного хранения констант и данных. Рассмотрим их по отдельности.

## Память программ

Объем встроенной flash-памяти программ в AVR-контроллерах составляет от 1 кбайта у ATtiny11 до 256 кбайт у ATmega2560. Первое число в наименовании модели соответствует величине этой памяти из ряда: 1, 2, 4, 8, 16, 32, 64, 128 и 256 кбайт. Память программ, как и любая другая flash-память, имеет страничную организацию (размер страницы, в зависимости от модели, составляет от 64 до 256 байт). Страница может программироваться только целиком. Число циклов перепрограммирования достигает 10 тыс.

С точки зрения программиста память программ можно считать построенной из отдельных ячеек — слов по два байта каждое. Устройство памяти программ (и только этой памяти) по двухбайтовым словам — очень важный момент, который нужно твердо усвоить. Такая организация обусловлена тем, что любая команда в AVR имеет длину ровно два байта. Исключения составляют команды `JMP`, `CALL` и некоторые другие (например, `LDS`), которые оперируют с 16-разрядными и более длинными адресами, длина этих команд равна четырем байтам и они применяются лишь в моделях с памятью программ объемом свыше 8 кбайт (подробнее см. главу 5). Во всех остальных случаях счетчик команд сдвигается при выполнении очередной команды на два байта (одно слово), поэтому необходимую емкость памяти легко подсчитать, зная число используемых команд. Абсолютные адреса в памяти про-

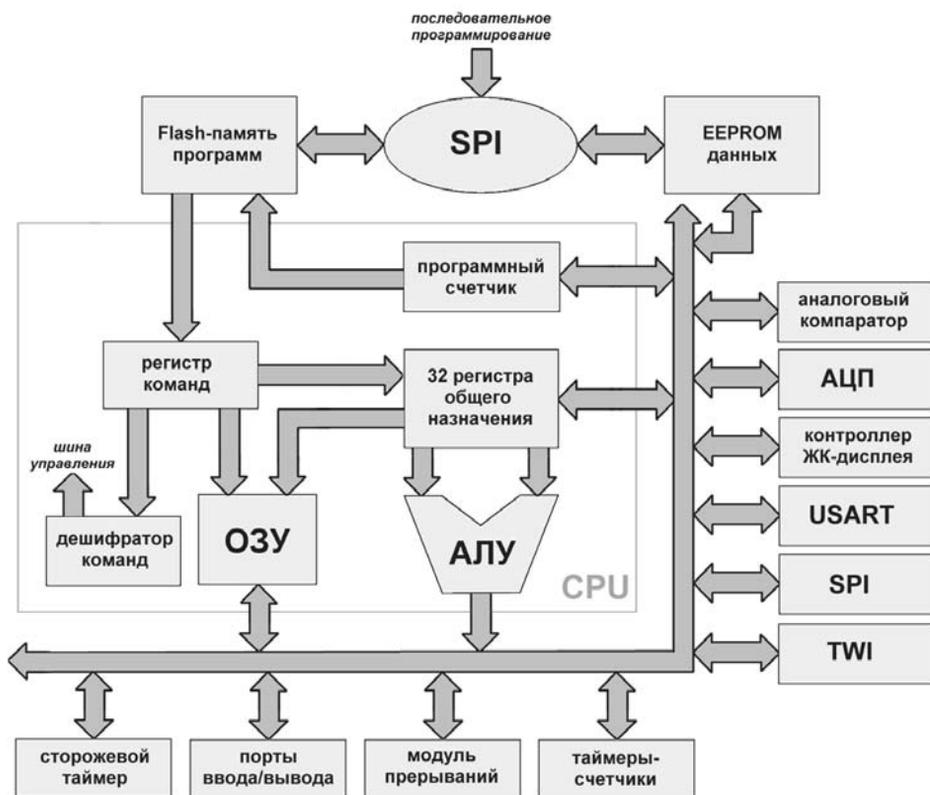


Рис. 2.1. Общая структурная схема микроконтроллеров AVR

грамм (указываемые, например, в таблицах векторов прерываний в техническом описании МК) также отсчитываются в словах.

### ЗАМЕТКИ НА ПОЛЯХ

Приведем пример интересного случая адресации, который представляет команда для чтения констант из памяти `LPM` (а также `ELPM` в МК с памятью программ 128 кбайт и более). Эта команда подразумевает чтение по *байтовому* адресу, указанному в двух старших `POH` (образующих так называемый регистр `Z`, см. *далее*). Однако чтобы не нарушать «чистоту» концепции организации памяти программ по словам, разработчики запутали этот простой вопрос, указав в описании, что при вызове команды `LPM` старшие 15 разрядов регистра `Z` адресуют *слово* в памяти, а младший разряд выбирает младший или старший байт (при равенстве разряда 0 или 1 соответственно) этого слова. Легко, однако, заметить, что байтовая и пословная организации памяти при таком подходе эквивалентны.

Последний адрес существующего объема памяти программ для конкретной модели обозначается константой `FLASHEND`. По умолчанию все контроллеры AVR всегда начинают выполнение программы с адреса `$0000`. Если в программе нет прерываний, то с этого адреса может начинаться прикладная программа. В противном случае по данному адресу располагается так называемая *таблица векторов прерываний*, подробнее о которой мы будем говорить в *главах 4 и 5*. Здесь укажем лишь,

что первым в этой таблице (по тому же адресу \$0000) всегда размещается вектор сброса RESET, который указывает на процедуру, выполняющуюся при сбросе МК (в том числе и при включении питания).

### ПРИМЕЧАНИЕ

В ассемблере AVR можно обозначать шестнадцатеричные числа в «паскалевском» стиле, предваряя их знаком \$, при этом стиль языка C (0x00) тоже действителен, а вот «интеловский» способ (00h) не работает. Подробнее об обозначениях чисел различных систем счисления в AVR-ассемблере см. главу 5.

В последних адресах памяти программ контроллеров семейства Mega может располагаться так называемый *загрузчик* — специальная программа, которая управляет загрузкой и выгрузкой прикладных программ из основного объема памяти. В этом случае положение вектора сброса и всей таблицы векторов прерываний (т. е. фактически начального адреса, с которого начинается выполнение программы) может быть изменено установкой специальных конфигурационных ячеек (см. главу 5).

## Память данных (ОЗУ, SRAM)

В отличие от памяти программ, адресное пространство памяти данных адресуется побайтно (а не пословно). Адресация полностью линейная, без какого-то деления на страницы, сегменты или банки, как это принято в некоторых других системах. Младшие МК семейства Tiny (включая Tiny1x и Tiny28) памяти данных, как таковой, не имеют, ограничиваясь лишь регистровым файлом (РОН) и регистрами ввода-вывода (РВВ). В других моделях объем встроенной SRAM колеблется от 128 байт в представителях семейства Tiny (например, у ATtiny2313) до 4–8 кбайт у старших моделей Mega.

Адресное пространство статической памяти данных (SRAM) условно делится на несколько областей, показанных на рис. 2.2. Темной заливкой выделена часть, относящаяся к собственно встроенной SRAM, до нее по порядку адресов расположено адресное пространство регистров (первые 32 байта занимает РОН, еще 64 — РВВ). Для старших моделей Mega со сложной структурой (например, ATmega128) 64-х регистров ввода-вывода может оказаться недостаточно, поэтому в них для дополнительных РВВ выделяется отдельное адресное пространство (от \$60 до максимально возможного в байтовой адресации значения \$FF, итого таких регистров может быть всего 160).

### ЗАМЕТКИ НА ПОЛЯХ

В архитектуре МК AVR понятие «ввода-вывода» употребляется в двух смыслах: во-первых, имеются «порты ввода-вывода» (I/O ports), которые мы рассмотрим в главе 3. Во-вторых, «регистры ввода-вывода» (РВВ) в структуре AVR называются регистры, которые обеспечивают доступ к дополнительным компонентам, внешним по отношению к GPU, за исключением ОЗУ (в том числе и к портам ввода-вывода). Такое подразделение приближает структуру МК AVR к привычной конфигурации персонального компьютера, где доступ к любым внешним по отношению к центральному процессору компонентам, кроме памяти, осуществляется через порты ввода-вывода.