

Введение

Интегральная технология за первые 20..30 лет своего развития достигла таких относительных темпов роста характеристик качества, которых не знала ни одна область человеческой деятельности (включая и такие бурно растущие, как авиация и космонавтика).

Появление МП как специального класса электронных цифровых приборов в начале 70-х обусловлено обозначившимся к этому времени противоречием между быстро развивавшимися потребностями в цифровых управляющих устройствах рынка товаров для конечного потребителя, с одной стороны, и тогдашними возможностями производства такого рода электроники, с другой. К тому времени цифровая техника в достаточной мере проявила свои потребительские качества и, как следствие, сформировался устойчивый спрос на различные цифровые контроллеры. Однако процесс их разработки и производства оставался весьма длительным и дорогостоящим. Очевидное решение проблемы - разработка прибора, обладающего универсальными возможностями и допускающего настройку на специфику выполняемой задачи, который можно было бы выпускать массовым тиражом и за счет этого добиться сравнительно низкой себестоимости. В 1971 году небольшая молодая компания Intel предложила такое устройство - 4-разрядную микросхему i4004 - первая БИС, которую с полным основанием можно отнести к классу МП.

Микропроцессорные решения в области РЭС в основном связаны с реализациями задач цифровой обработки сигналов. В этом смысле специфика настоящего курса напрямую корреспондируется с главным сегодняшним вектором движения информационных технологий. Этот вектор связан с телекоммуникационным транспортом через глобальные сети мультимедиа-продуктов и доминантой здесь являются алгоритмы и программно-аппаратная поддержка процессов сжатия данных, а эти инструменты в настоящее время во многом базируются на спектральных преобразованиях. Означенное обстоятельство делает курс «Микропроцессорные устройства РЭС» весьма актуальным не только в ракурсе специализации, но и в целом в общем горизонте современной техники.

В данном учебном пособии представлена первая часть курса «Микропроцессорные устройства РЭО», включающая в себя историю развития микропроцессорной техники, характеристики микропроцессоров, классификацию микропроцессоров по различным признакам, архитектуру микропроцессоров, способы ввода-вывода данных, а также сравнительно новое семейство микроконтроллеров AVR, которые в последние годы стали одними из самых востребованных на рынке цифровых устройств.

1. Общие сведения о микропроцессорах

1.1. История развития микропроцессоров

Микропроцессор (МП) - устройство, состоящее из одной или нескольких БИС, включающее в себя все необходимые средства для обработки и управления данными и рассчитанное на совместную работу с устройствами памяти и ввода-вывода информации.

Микропроцессор возник не на пустом месте, его появлению предшествовало десятилетие развития микроэлектроники, которая к началу 70-х годов добилась существенных успехов в создании интегральных микросхем. И все же выход i4004 стал революционным событием, приведшим к коренной ломке представлений о возможностях микроэлектронной и вычислительной техники. Вычислительная техника вступила в бурную микропроцессорную эру, которая характеризуется гонкой технологий и высокой скоростью совершенствования технического уровня изделий. Преимущества микропроцессоров были мгновенно оценены, и буквально в считанные месяцы к выпуску аналогичных БИС- приступили другие компании.

Вопрос о приоритете создания первого в мире микропроцессора уже к 1974 г. полностью прояснился. Вот как писал тогда об этом журнал "Electronics": "Intel определенно принадлежит честь использования идеи микропроцессора, и она была первой, кто выпустил их на рынок, хотя большие заслуги принадлежат также другим фирмам и отдельным лицам, которые тем или иным путем содействовали развитию техники больших интегральных схем.

Следует вспомнить компанию Viatron, которая в 1968 г. удивила мир, объявив о своем намерении создать систему обработки данных на основе собственного 8-разрядного микропроцессора, управляемого примитивной программой, записанной в ПЗУ. Однако эта компания встретила с серьезными финансовыми трудностями и спустя два года обанкротилась. Между тем, General Electric разработала однокристалльный базовый логический блок и вплотную подошла к созданию микропроцессора".

Уже в 1971 г. Intel не скрывала своих достижений. Она существовала всего четыре года и была известна микросхемами памяти, когда в шумной рекламной кампании провозгласила "новую эру в микроэлектронике", выпустив на рынок первый коммерческий микропроцессор.

Один кристалл i4004 обладал почти такими же вычислительными возможностями, как и ENTAC выпуска 1946 г. с его 18 000 электронных ламп. Микропроцессор был дешевым (около \$200) и имел крошечные размеры, что позволяло разработчикам создавать принципиально новые системы. Скептики предсказывали, что рынок однокристалльного процессора будет такой же крошечный, как и он сам, но действительность превзошла все ожидания.

Идея i4004 возникла у одного из основателей компании Intel Роберта Нойса (Robert Noyce). Он предложил использовать технологический процесс изготовления кристаллов памяти для создания логических чипов. Но какими именно должны быть эти логические устройства? Этот вопрос решил счастливый случай. В 1969 г. японская компания Busicom заказала Intel разработку комплекта из 12 специализированных чипов для настольных калькуляторов. Вместо этого Intel предложила строить калькуляторы на основе однокристалльного микропроцессора, который впоследствии приобрел известность как Intel 4004. Конструкторскую разработку выполнили Тэд Хофф (Тед Ной), Стэн Мэйзор (Stan Mazor) и Федерико Фэджин (Federico Faggin). Они позаимствовали многие решения из архитектуры больших ЭВМ. Однако чтобы разместить процессор на одном кристалле, им пришлось снизить его разрядность до четырех. Это позволило уменьшить количество транзисторов и упаковать процессор в 16-контактный корпус - самый большой в то время. Процессор мог выполнять 45 инструкций, и для программируемого калькулятора компании Busicom к комплекту MCS-4 достаточно было добавить четыре кристалла памяти по 256 байт. Процессор имел 2300 транзисторов, 4-разрядную шину данных и 12-разрядную шину адреса (шины мультиплексировались) и тактовую частоту 108 kHz. Это позволяло ему выполнять 0.06 млн. операций в секунду (MIPS).

Чип i4004 открыл дорогу более мощным и быстродействующим микропроцессорам следующего поколения, характерными признаками которого можно считать:

- переход на более прогрессивные технологии n-МОП и КМОП, позволившие повысить быстродействие МП до 2..2,5 МГц (200..500 тыс. операций RR), снизить потребление мощности (КМОП);

- значительные архитектурные отличия: расширение системы операций, использование широкого набора способов адресации (прямая, косвенная, относительная, безадресная, непосредственная), введены подсистемы прерываний и прямого доступа в память (ПДП), предусмотрен механизм универсального стека;

- структурные отличия: шины адреса и данных разделены, уменьшено число вспомогательных ИС и СИС.

Как только разработка i4004 поступила в серийное производство, Тэд Хофф и Стэн Мэйзор приступили к созданию его 8-разрядной версии - i8008. Это был первый микропроцессор, имевший систему прерываний и мультиплексирующую шину адрес-данные, но функционировал он неудовлетворительно. И уже в 1974 г. появился также 8-разрядный кристалл 8080. Он содержал 6000 транзисторов, выполнял более 250 команд, причем некоторые из них работали с парами регистров - с 16-разрядными данными. Его адресное пространство 64 КВ по тем

временам было огромным (современный Pentium Pro может адресоваться к 64 ТБ виртуальной памяти).

С выходом 8-разрядных чипов Intel открылись широчайшие перспективы для разработки дешевого программируемого оборудования. И события не заставили себя ждать. Вскоре Гари Килдэлл (Gary Kildall) из Digital Research создал для процессора i8080 операционную систему. Этот программный продукт значительно упростил процесс разработки и отладки прикладного программного обеспечения и, вместе с чипом i8080, фактически открыл невиданные ранее возможности создания доступных массовому потребителю компьютеров.

Первым персональным компьютером принято считать микроЭВМ Altair 8800 компании MITS, которая поступила в продажу в 1975 г. и уже была оснащена операционной системой CP/M. Машина выпускалась в виде набора "Сделай сам" или как законченное изделие и предназначалась для любителей сложной технической игрушки. В дальнейшем появился расширенный вариант устройства для бухгалтерских и инженерных расчетов.

Многие компании с середины 70-х годов приступили к производству личных, домашних, любительских микроЭВМ, но главный успех достался компании Apple, выпустившей в 1977 г. одноименную ПЭВМ, которая отличалась совершенным программным обеспечением, рассчитанным на самого неподготовленного пользователя. Дружественное программное обеспечение (friendly software) расширило круг потенциальных покупателей, ограниченный ранее любителями электроники и профессионалами.

Микропроцессорная гонка набирала все большее ускорение. Уже упоминавшийся Федерико Фэджин и Масатоши Шима (Masatoshi Shima) основали компанию Zilog, которая молниеносно наладила выпуск чипа Z80 - несколько усовершенствованного аналога 8080, совместимого и с ним, и с операционной системой CP/M. Микропроцессор Z80 имел встроенную программу регенерации динамического ОЗУ и вместе с CP/M составил "идеальную пару" для массового производства дешевых ПК.

Вскоре дебютировала Motorola со своим 6800-м, стартовали Texas Instruments, National Semiconductor и Fairchild, разработав свои собственные микропроцессоры; большинство из них использовалось для встраивания в "интеллектуальные" приборы.

Дальнейшее усовершенствование микропроцессоров связано со стремлением к увеличению быстродействия МПС и переходом на биполярные технологии - ТТЛ и ТТЛШ.

Это привело к значительным структурным изменениям в микропроцессорах по сравнению с предшествующими моделями:

- микропроцессоры выпускаются в виде секций со средствами межразрядных связей, позволяющими объединять в одну систему

произвольное число секций для достижения заданной разрядности. В состав секций включалось арифметико-логическое устройство, регистры общего назначения и некоторые элементы устройства управления;

- устройство управления вынесено на отдельный кристалл (группу кристаллов), общий для всех процессорных секций;

- за счет резерва внешних выводов (малая разрядность) предусмотрены отдельные шины ввода и вывода данных, адреса, причем данные от разных источников вводились по различным шинам;

- кристаллы управления реализуют устройство управления с программируемой логикой, что позволяет достаточно легко реализовать практически любую систему команд на фиксированной структуре операционного устройства.

На новый и чрезвычайно динамичный сектор компьютерного рынка обратила внимание корпорация IBM - гигант вычислительной индустрии. Выпустив в 1981 г. свой первый IBM PC, она сразу стала законодателем мод для производителей персоналок и поныне занимает лидирующее место на всемирном компьютерном рынке.

Усовершенствованный 16-разрядный чип появился в 1978 г. одновременно в "двух лицах" - 8086 и 8088. Первый чип содержал 29000 транзисторов, более развитую систему команд, имел 16-разрядные внешнюю и внутреннюю шины данных, 20-разрядную шину адреса и работал на частоте 5 MHz (некоторые версии имели частоту до 10 MHz). Его производительность в 10 раз превышала этот показатель своего предшественника (0.33 MIPS, но на 16-разрядной архитектуре).

Строго придерживаясь принципа программной совместимости, Intel сохранила 10-разрядный адресный регистр, присущий чипу i8080, но для расширения адресного пространства до 1 МВ добавила сегментный регистр, тем самым создав новую идеологию адресации, которая применяется и сегодня.

В 1980 году фирма Intel выпускает микроконтроллер i8048. Чуть позже в этом же году Intel выпускает следующий микроконтроллер i8051. Удачный набор периферийных устройств, возможность гибкого выбора внешней или внутренней программной памяти и приемлемая цена обеспечили этому микроконтроллеру успех на рынке. С точки зрения технологии микроконтроллер i8051 являлся для своего времени очень сложным изделием — в кристалле было использовано 128 тыс. транзисторов, что в 4 раза превышало количество транзисторов в 16-разрядном микропроцессоре i8086.

Процессор i8086 уже имел примитивный конвейер: блок интерфейса с шиной (Bus Interface Unit) поставлял поток инструкций исполнительному блоку (Execution Unit) из 6-байтовой очереди, так что операции выборки и исполнения осуществлялись одновременно. Это был истинно 16-разрядный процессор, в

отличие от следующей модели — i8088 (с математическим сопроцессором 8087), которая вышла спустя год.

Процессор i8088 совпадал со своим предшественником во всем, кроме того, что имел 8-разрядную внешнюю шину данных. Это позволяло строить недорогие системы, используя 8-разрядные наборы чипов серии i8085. Именно на процессор 8088 обратила, наконец, внимание корпорация IBM — флагман компьютерной индустрии. В октябре 1981 г. представлена рынку первая IBM PC, а уже к концу года было продано более 35 тыс. компьютеров.

Данные модели обрели мировую известность благодаря вычислительным системам класса Extended Technology - знаменитым PC XT компании IBM с прекрасным DOS-ом и незабываемой оболочкой Norton Commander. Архитектура 8086/8088 превратилась в промышленный стандарт для персональных компьютеров.

На рис. 1 показаны направления развития некоторых типов микропроцессорных систем за период 1975-1995г.

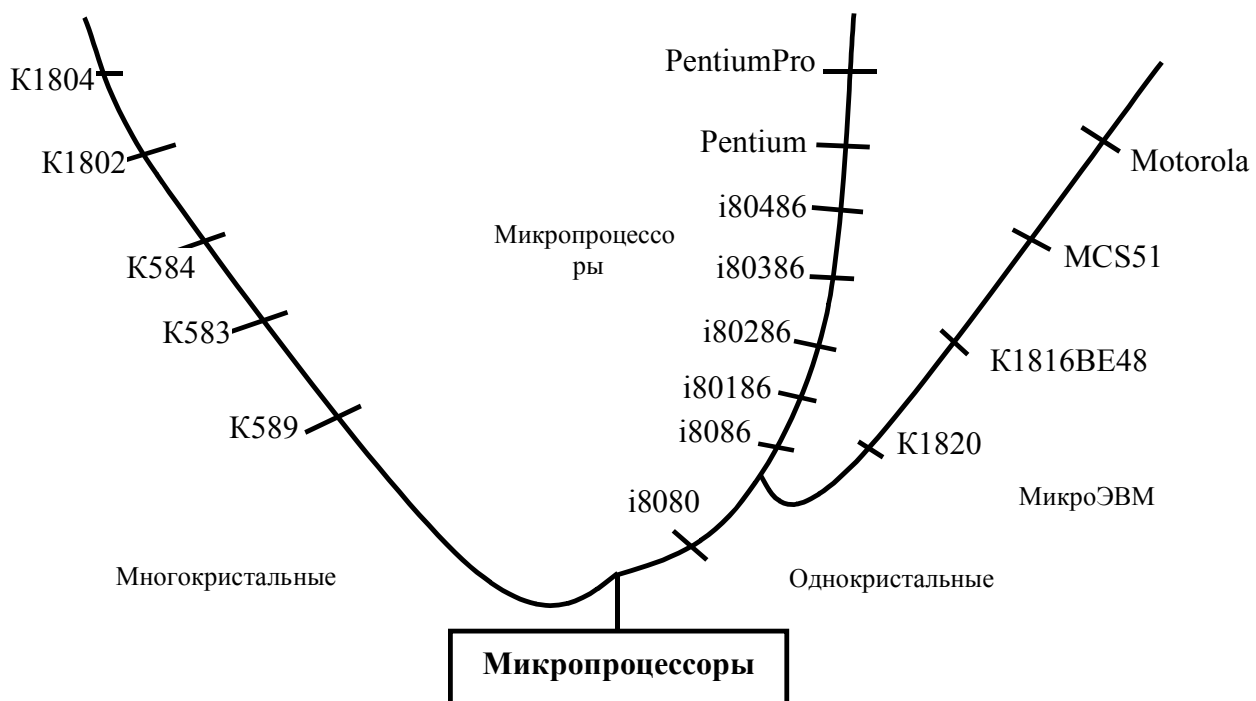


Рис.1. Последовательность развития микропроцессоров

Достигнутые к началу 70-х гг. успехи в области технологии интегральных микросхем и организации вычислительных устройств привели к появлению нового

класса приборов — микропроцессоров. Сегодня микропроцессорная техника — индустриальная отрасль со своей методологией и средствами проектирования.

Можно выделить следующие направления разработок, ориентированных на повышение производительности МП, которые велись с момента появления i8080:

- увеличение частоты и разрядности;

- конвейеризация этапов обработки команд - организация структуры МП в виде конвейера, звенья которого обрабатывают отдельные этапы выполнения команды, позволяет совмещать во времени выполнение нескольких последовательных команд;

- использование кэш-памяти - быстродействующей памяти небольшого объема непосредственно на кристалле МП, содержащей команды и данные, которые обрабатываются в текущий момент. Позволяет значительно уменьшить количество медленных операций обращения к внешней памяти;

- использование сопроцессоров - вычислителей, работающих параллельно с обрабатываемым ядром МП и ускоренно выполняющих специальные операции, например арифметические процедуры над числами, представленными в формате с плавающей запятой;

- использование RISC-процессоров - МП с уменьшенным набором команд. В уменьшенный набор входят наиболее часто выполняемые команды, обработка которых организована аппаратно, что дает заметный выигрыш в быстродействии.

К настоящему времени накоплен большой практический опыт проектирования микропроцессоров и микропроцессорных систем, область применения которых постоянно расширяется. На сегодняшний день существует более 200 модификаций микроконтроллеров, совместимых с i8051, выпускаемых двумя десятками компаний, и большое количество микроконтроллеров других типов. Популярностью у разработчиков пользуются 8-битные микроконтроллеры PIC фирмы Microchip Technology и AVR фирмы Atmel, 16-битные MSP430 фирмы TI, а также ARM, архитектуру которых разрабатывает фирма ARM и продаёт лицензии другим фирмам для их производства.

1.2. Характеристики микропроцессоров

К основным характеристикам микропроцессоров относятся:

- разрядность;

- быстродействие (тактовая частота);

- потребляемая мощность;

- технология (уровень логических сигналов);

- архитектурные особенности: система операций, способы адресации, наличие и организация подсистем прерываний и прямого доступа к памяти,

объем и организация памяти, конвейер операций, аппаратная поддержка системы виртуальной памяти и т.п.;

структурные особенности: количество и назначение шин (стандарт интерфейса), внутренняя структура;

- число источников питания;

- число больших интегральных схем (БИС) в комплекте.

1.3. Классификация микропроцессоров

По числу БИС в микропроцессорном комплекте различают микропроцессоры однокристалльные, многокристалльные и многокристалльные секционные.

Процессоры даже самых простых вычислительных машин имеют сложную функциональную структуру, содержат большое количество электронных элементов и множество разветвленных связей. Изменять структуру процессора необходимо так, чтобы полная принципиальная схема или ее части имели количество элементов и связей, совместимое с возможностями БИС. При этом микропроцессоры приобретают внутреннюю магистральную архитектуру, т. е. в них к единой внутренней информационной магистрали подключаются все основные функциональные блоки (арифметико-логический, рабочих регистров, стека, прерываний, интерфейса, управления и синхронизации и др.).

Для обоснования классификации микропроцессоров по числу БИС надо распределить все аппаратные блоки процессора между основными тремя функциональными частями: операционной, управляющей и интерфейсной. Сложность операционной и управляющей частей процессора определяется их разрядностью, системой команд и требованиями к системе прерываний; сложность интерфейсной части разрядностью и возможностями подключения других устройств (памяти, внешних устройств, датчиков и исполнительных механизмов и др.). Интерфейс процессора содержит несколько десятков информационных шин данных (ШД), адресов (ША) и управления (ШУ).

Однокристалльные микропроцессоры получаются при реализации всех аппаратных средств процессора в виде одной БИС или сверхбольшой интегральной схемы (СБИС). По мере увеличения степени интеграции элементов в кристалле и числа выводов корпуса параметры однокристалльных микропроцессоров улучшаются. Однако возможности однокристалльных микропроцессоров ограничены аппаратными ресурсами кристалла и корпуса. Для получения многокристалльного микропроцессора необходимо провести разбиение его логической структуры на функционально законченные части и реализовать их в виде БИС (СБИС). Функциональная законченность БИС многокристалльного микропроцессора означает, что его части выполняют заранее определенные функции и могут работать автономно.

На рис. 2 а показано функциональное разбиение структуры процессора при создании трехкристального микропроцессора (пунктирные линии), содержащего БИС операционного (ОП), БИС управляющего (УП) и БИС интерфейсного (ИП) процессоров.

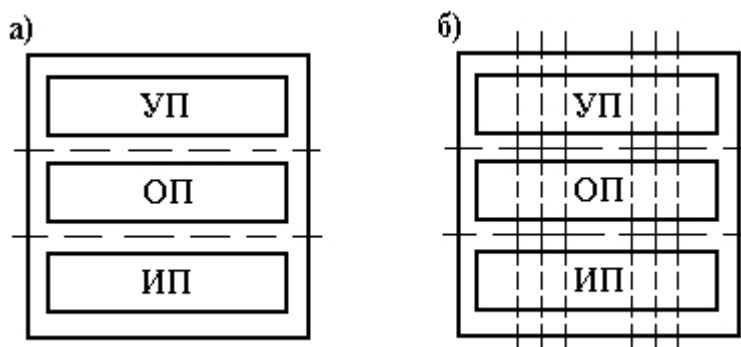


Рис. 2. Функциональная структура микропроцессора

Операционный процессор служит для обработки данных, управляющий процессор выполняет функции выборки, декодирования и вычисления адресов операндов и также генерирует последовательности микрокоманд. Автономность работы и большое быстродействие БИС УП позволяет выбирать команды из памяти с большей скоростью, чем скорость их исполнения БИС ОП. При этом в УП образуется очередь еще не исполненных команд, а также заранее подготавливаются те данные, которые потребуются ОП в следующих циклах работы. Такая опережающая выборка команд экономит время ОП на ожидание операндов, необходимых для выполнения команд программ. Интерфейсный процессор позволяет подключить память и периферийные средства к микропроцессору. Он по существу является сложным контроллером для устройств ввода/вывода информации. БИС ИП выполняет также функции канала прямого доступа к памяти.

Выбираемые из памяти команды распознаются и выполняются каждой частью микропроцессора автономно и поэтому может быть обеспечен режим одновременной работы всех БИС МП, т.е. конвейерный поточный режим исполнения последовательности команд программы (выполнение последовательности с небольшим временным сдвигом). Такой режим работы значительно повышает производительность микропроцессора.

Многокристальные секционные микропроцессоры получаются в том случае, когда в виде БИС реализуются части (секции) логической структуры процессора при функциональном разбиении ее вертикальными плоскостями (рис. 2 б). Для построения многоразрядных микропроцессоров при параллельном включении секций БИС в них добавляются средства "стыковки".

Для создания высокопроизводительных многоразрядных микропроцессоров требуется столь много аппаратных средств, не реализуемых в доступных БИС, что

может возникнуть необходимость еще и в функциональном разбиении структуры микропроцессора горизонтальными плоскостями. В результате рассмотренного функционального разделения структуры микропроцессора на функционально и конструктивно законченные части создаются условия реализации каждой из них в виде БИС'. Все они образуют комплект секционных БИС МП.

Таким образом, микропроцессорная секция - это БИС, предназначенная для обработки нескольких разрядов данных или выполнения определенных управляющих операций. Секционность БИС МП определяет возможность "наращивания" разрядности обрабатываемых данных или усложнения устройств управления микропроцессора при "параллельном" включении большего числа БИС.

Однокристалльные и трехкристалльные БИС МП, как правило, изготавливают на основе микроэлектронных технологий униполярных полупроводниковых приборов, а многокристалльные секционные БИС МП – на основе технологии биполярных полупроводниковых приборов. Использование многокристалльных микропроцессорных высокоскоростных биполярных БИС, имеющих функциональную законченность при малой физической разрядности обрабатываемых данных и монтируемых в корпус с большим числом выводов, позволяет организовать разветвление связи в процессоре, а также осуществить конвейерные принципы обработки информации для повышения его производительности.

По назначению различают универсальные и специализированные МП.

Универсальные микропроцессоры могут быть применены для решения широкого круга разнообразных задач. При этом их эффективная производительность слабо зависит от проблемной специфики решаемых задач. Специализация МП, т.е., его проблемная ориентация на ускоренное выполнение определенных функции позволяет резко увеличить эффективную производительность при решении только определенных задач.

Среди специализированных микропроцессоров можно выделить различные микроконтроллеры, ориентированные на выполнение сложных последовательностей логических операций, математические МП, предназначенные для повышения производительности при выполнении арифметических операций за счет, например, матричных методов их выполнения. МП для обработки данных в различных областях применений и т. д. С помощью специализированных МП можно эффективно решать новые сложные задачи параллельной обработки данных. Например, конволюция позволяет осуществить более сложную математическую обработку сигналов, чем широко используемые методы корреляции, которые в основном сводятся к сравнению всего двух серий данных (входных, передаваемых формой сигнала, и фиксированных опорных) и к определению их подобия. Конволюция дает возможность в реальном масштабе времени находить соответствие для сигналов изменяющейся формы путем

сравнения их с различными эталонными сигналами, что например может позволить эффективно выделить полезный сигнал на фоне шума.

Разработанные однокристалльные конвольверы используются в устройствах опознавания образов в тех случаях, когда возможности сбора данных превосходят способности системы обрабатывать эти данные.

По виду обрабатываемых входных сигналов различают цифровые и аналоговые микропроцессоры. Сами микропроцессоры являются цифровыми устройствами, однако, они могут иметь встроенные аналого-цифровые и цифро-аналоговые преобразователи. Поэтому входные аналоговые сигналы передаются в МП через преобразователь в цифровой форме, обрабатываются и после обратного преобразования в аналоговую форму поступают на выход. С архитектурной точки зрения такие микропроцессоры представляют собой аналоговые функциональные преобразователи сигналов и называются аналоговыми микропроцессорами. Они выполняют функции любой аналоговой схемы (например, производят генерацию колебаний, модуляцию, смещение, фильтрацию, кодирование и декодирование сигналов в реальном масштабе времени и т.д., заменяя сложные схемы, состоящие из операционных усилителей, катушек индуктивности, конденсаторов и т.д.). При этом применение аналогового микропроцессора значительно повышает точность обработки аналоговых сигналов и их генерацию, а также расширяет функциональные возможности за счет программной "настройки" цифровой части микропроцессора на различные алгоритмы обработки сигналов.

Обычно в составе однокристалльных аналоговых МП имеется несколько каналов аналого-цифрового и цифро-аналогового преобразования. В аналоговом микропроцессоре разрядность обрабатываемых данных достигает 24 бит и более, большое значение уделяется увеличению скорости выполнения арифметических операций.

Отличительная черта аналоговых микропроцессоров - способность к переработке большого объема числовых данных, т.е. к выполнению операций сложения и умножения с большой скоростью при необходимости даже за счет отказа от операций прерываний и переходов. Аналоговый сигнал, преобразованный в цифровую форму, обрабатывается в реальном масштабе времени и передается на выход обычно в аналоговой форме через цифро-аналоговый преобразователь. При этом согласно теореме Котельникова частота квантования аналогового сигнала должна вдвое превышать верхнюю частоту сигнала.

Сравнение цифровых микропроцессоров производится сопоставлением времени выполнения ими списков операций. Сравнение же аналоговых микропроцессоров производится по количеству эквивалентных звеньев аналого-цифровых фильтров рекурсивных фильтров второго порядка. Производительность аналогового микропроцессора определяется его способностью быстро выполнять операции

умножения: чем быстрее осуществляется умножение, тем больше эквивалентное количество звеньев фильтра в аналоговом преобразователе и тем более сложный алгоритм преобразования цифровых сигналов можно задавать в микропроцессоре.

Одним из направлений дальнейшего совершенствования аналоговых микропроцессоров является повышение их универсальности и гибкости. Поэтому вместе с повышением скорости обработки большого объема цифровых данных будут развиваться средства обеспечения развитых вычислительных процессов обработки цифровой информации за счет реализации аппаратных блоков прерывания программ и программных переходов.

По организации работы микропроцессоры делят на синхронные и асинхронные.

Синхронные микропроцессоры - микропроцессоры, в которых начало и конец выполнения операций задаются устройством управления (время выполнения операций в этом случае не зависит от вида выполняемых команд и величин операндов).

Асинхронные микропроцессоры позволяют начало выполнения каждой следующей операции определить по сигналу фактического окончания выполнения предыдущей операции. Для более эффективного использования каждого устройства микропроцессорной системы в состав асинхронно работающих устройств вводят электронные цепи, обеспечивающие автономное функционирование устройств. Закончив работу над какой-либо операцией, устройство вырабатывает сигнал запроса, означающий его готовность к выполнению следующей операции. При этом роль естественного распределителя работ принимает на себя память, которая в соответствии с заранее установленным приоритетом выполняет запросы остальных устройств по обеспечению их командной информацией и данными.

По организации структуры микропроцессорных систем (МПС) различают одно- и многомагистральные МПС.

В одномагистральных МПС все устройства имеют одинаковый интерфейс и подключены к единой информационной магистрали, по которой передаются коды данных, адресов и управляющих сигналов.

В многомагистральных МПС устройства группами подключаются к своей информационной магистрали. Это позволяет осуществить одновременную передачу информационных сигналов по нескольким (или всем) магистралям. Такая организация систем усложняет их конструкцию, но увеличивает производительность.

По количеству выполняемых программ различают одно- и многопрограммные микропроцессоры.

В однопрограммных микропроцессорах выполняется только одна программа. Переход к выполнению другой программы происходит после завершения текущей программы.

В много- или мультипрограммных микропроцессорах одновременно выполняется несколько (обычно несколько десятков) программ. Организация

мультипрограммной работы микропроцессорных управляющих систем позволяет осуществить контроль за состоянием и управлением большим числом источников или приемников информации.

1.4. Этапы разработки микропроцессоров

Микропроцессоры относятся к классу цифровых систем, поэтому методология их проектирования соответствует методологии проектирования цифровых систем при совместной разработке аппаратных и программных средств. Этапы проектирования микропроцессорных систем представлены на рис. 3.

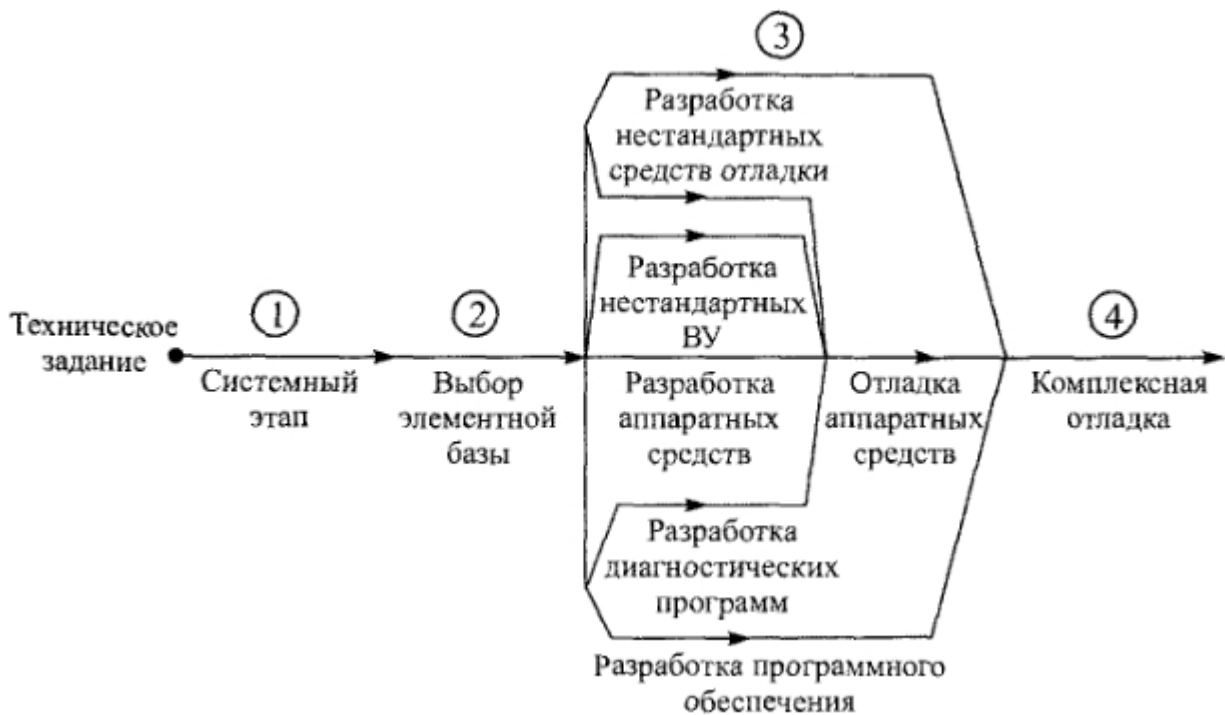


Рис. 3. Этапы разработки микропроцессорных систем

На **первом этапе проектирования** — системном — решаются общесистемные задачи разработки цифровых систем. В соответствии с функциональным назначением проектируемой системы определяют ее состав и структуру связей с внешней средой. В соответствии с техническим заданием рассчитывают характеристики представления, хранения и передачи обрабатываемой информации, т.е. способы кодирования и разрядность входных и выходных данных, скоростные характеристики обмена данными, протоколы обмена, временные характеристики информационных и управляющих сигналов,

характеристики защиты от ошибок. Затем для каждой функции обработки и управления разрабатывают математические методы преобразования информации и определяют количественные характеристики обработки данных. Формулируют требования к элементной базе, к ее разрядности, быстродействию, объемам хранимой информации, энергетическим и массогабаритным параметрам. Выбирают стандартные внешние устройства и формулируют требования к нестандартным внешним устройствам. Конкретный перечень решаемых задач зависит от области применения проектируемой системы, а их результаты во многом определяются опытом и интуицией разработчика-специалиста в области использования системы.

Второй этап проектирования — выбор элементной базы и разработка архитектуры МП. При этом учитывают требования, сформулированные на системном этапе: разрядность, быстродействие, объем памяти, класс решаемых задач, потребляемую мощность и др. Однако не менее важное значение имеют опыт разработчиков, наличие средств разработки МП.

Архитектуру МП разрабатывают с учетом характеристик выбранной элементной базы и средств разработки на ее основе. Определяют типы и форматы обрабатываемых данных, набор арифметических и логических операций и представление их результатов; организацию адресных пространств запоминающих устройств (ЗУ) и внешних устройств (ВУ), способы хранения информации и способы ее адресации, форматы команд, систему команд, характеристики системы прерываний, структуру внутрисистемного интерфейса, характеристики системы ввода/вывода.

Третий этап проектирования — разработка и отладка аппаратных средств и программного обеспечения проектируемых МП. При необходимости разрабатывают нестандартные средства отладки, внешние устройства и диагностические программы отладки. Все указанные виды работ можно выполнять параллельно.

Аппаратные средства МП проектируют на основе архитектуры, разработанной на предыдущем этапе, в соответствии с техническим заданием. При этом используют принципы модульного проектирования цифровой аппаратуры, выделяя функционально и конструктивно законченные модули, разрабатывают их внутрисистемный интерфейс и каждый модуль в отдельности. Для МП функциональными модулями обычно являются модули процессоров и его составных частей, ЗУ, устройства ввода/вывода. Конструктивным модулем является печатная плата. При разработке аппаратных средств МП используют системы автоматизации проектирования, типичные для разработки цифровой аппаратуры.

Для отладки аппаратных средств МП применяют традиционные для цифровой техники контрольно-измерительные устройства — осциллографы,

логические анализаторы и другие, а также специализированные средства — внутрисхемные эмуляторы.

Внутрисхемный эмулятор представляет собой специализированный МП на основе выбранной элементной базы, позволяющий имитировать работу всех частей разрабатываемых аппаратных средств МП в реальном времени, предоставляя в распоряжение отлаживаемой системы те ресурсы, которые отсутствуют в ней на данном этапе отладки. Управляет внутрисхемным эмулятором обычно универсальный МП, оснащенный средствами сопряжения эмулятора и специальным программным обеспечением управления отладкой. Внутрисхемный эмулятор можно реализовать на универсальном МП, который выполнен на выбранной для построения микропроцессорной системы элементной базе. Однако при этом требуется разработка нестандартных средств для подключения разрабатываемых аппаратных средств к МП.

Задача разработки аппаратных средств проектируемого МП существенно упрощается при использовании серийных средств разработки — одноплатных МП или комплектов модулей на выбранной элементной базе. В этом случае она сводится к выбору требуемого набора модулей и разработке средств сопряжения с ВУ.

Разработка нестандартных ВУ выполняется на основе требований, сформулированных к таким ВУ на системном этапе. Диагностические (тестовые) программы разрабатываются для проверки функционирования отдельных устройств микропроцессорной системы.

Принцип проверки работы МП с помощью тестовых программ состоит в том, что МП выполняет некоторые специально предназначенные для диагностирования устройства программы. Причем результаты, получаемые на отдельных этапах в процессе исполнения тестовых программ, заранее известны. Поэтому для выявления неисправности в программе оказывается достаточным предусмотреть выполнение операций, связанных с сопоставлением получаемых результатов с ожидаемыми, и если обнаруживается несовпадение указанных данных, то в программе происходит переход на ветвь выдачи соответствующей информации о наличии неисправности.

Тестовые программы строятся из отдельных подпрограмм, предназначенных для проверки работы отдельных блоков микропроцессорной системы. Вначале производится проверка микропроцессора. Если подпрограмма проверки этого блока не выявила неисправности, производится переход к следующим подпрограммам, предназначенным, например, для проверки оперативной и постоянной памяти. Затем могут проверяться программируемые интерфейсы ввода-вывода и т. д. Кроме того, если не обнаруживаются неисправности простейшими тестовыми программами малого объема, предназначенными, например, для обнаружения постоянных одиночных ошибок в разрядах узлов (типа

«залипания» в состоянии лог, 0 или лог. 1), то можно включить более сложные программы, выявляющие многократные ошибки, случайно возникающие ошибки (сбои) и т. д.

Тестовое диагностирование обычно проводится после пуска МПУ, но оно может проводиться и в процессе нормальной работы. В последнем случае по истечении определенного времени, отмечаемого таймером, либо при возникновении некоторых событий производится прерывание микропроцессора с переходом на выполнение хранящейся в постоянной памяти МПУ тестовой программы. Выполнение тестовой программы может также предусматриваться в режимах, когда микропроцессор не производит какой-либо полезной работы.

Программное обеспечение (ПО) включает в себя:

- разработку алгоритмов, реализующих функции обработки и управления, ориентированных на выбранную архитектуру микропроцессорной системы и язык программирования;
- составление и редактирование программ;
- трансляцию программ в машинные коды;
- компоновку программных модулей;
- отладку программ;
- запись в БИС ПЗУ или на внешний носитель для использования в системе и документирование.

Разработка алгоритмов состоит в нахождении численного решения для каждой функции с учетом архитектурных особенностей выбранных микропроцессорных средств. При разработке программ используют принципы модульности и проектирования "сверху вниз". Вначале разрабатывают иерархический состав модулей (головной, подчиненные) и определяют интерфейс между модулями (вызов модулей, передача данных между модулями), а затем — каждый программный модуль в отдельности.

Для составления программ используют язык Ассемблера, позволяющий получить минимальные объемы программ, что обеспечивает более высокое быстродействие и меньшие объемы ЗУ. Языки высокого уровня используют, если требования по быстродействию и объему ЗУ не критичны. Более высокая производительность программирования на языке высокого уровня обеспечивает более низкую стоимость и большую надежность программ. Важными факторами при выборе языка программирования является наличие у разработчика средств компиляции разрабатываемых программ в машинные коды МП (программ — компиляторов или Ассемблеров), опыта работы с языком и др.

Составляют программы с использованием редактора текстов, позволяющего оперативно вносить изменения в текст программы и представлять ее в форме, удобной для чтения. Компиляция программ позволяет получить

программу на машинном языке. Ее выполняют с помощью программ-ассемблеров или компиляторов.

При создании программ для новых микропроцессорных комплектов (МПК) следует применять программы, разработанные ранее для младших совместимых МПК. Если команды комплектов отличаются, то для их преобразования используют специальную программу-конвертор, обеспечивающую перекодировку команд из языка Ассемблера младшего МПК в язык Ассемблера старшего МПК, после этого возможна компиляция в машинные коды.

Объектные программы модулей формируются обычно в перемещаемом формате, т.е. они могут размещаться по произвольным адресам ЗУ. В таком виде их можно заносить в библиотеку специальной программой-библиотекарем и использовать в дальнейшем при компоновке общей программы. В библиотеке хранятся также стандартные подпрограммы, реализующие типовые функции управления, обработки данных, ввода/вывода.

Компоновку программ выполняет редактор связей, позволяющий объединять отдельные программы в общую путем согласования относительных адресов межмодульных ссылок. Программа-загрузчик размещает отдельные модули или полную программу по реальным адресам ЗУ, благодаря чему полученная абсолютная программа готова для исполнения. Отладка программ позволяет проверить правильность их работы без использования разрабатываемых аппаратных средств. Для этой цели применяют программную (имитатор) или аппаратную (эмулятор, отладочный модуль) модели, на которых выполняются разработанные программы с возможностью контроля результатов в произвольных контрольных точках программы.

В программной модели выполнение команд разрабатываемой программы моделируется в системе команд МП. Поэтому отладка ведется в относительном времени. Аппаратная модель реализуется на микропрограммируемых интегральных схемах или на БИС используемого МПК и позволяет вести отладку программ в реальном времени. Для отладки программ можно использовать также внутрисхемный эмулятор, оснащенный соответствующим интерфейсом и программой управления отладки (программой-отладчиком).

Отладку программ удобно вести в диалоговом режиме, позволяющем оперативно устанавливать контрольные точки в программе и получать информацию о состоянии программы в них (трассу отладки). Для отладки обычно используют программные или программно-аппаратные интерфейсы модели и программы управления отладкой.

Завершает разработку программного обеспечения микропроцессорной системы его запись на носители информации и документирование.

Четвертый этап проектирования МП — комплексная отладка — завершает процесс разработки микропроцессорной системы. На этом этапе

проверяют функционирование разработанной системы в реальном времени и в условиях, близких к реальным. Для этой цели используют испытательную и измерительную аппаратуру, в частности, специализированные тестеры, инженерные пульта управления и др. Наиболее совершенными и перспективными являются программируемые отладочные системы с возможностями имитации входных сигналов, сопряжения с разработанной системой, интерпретации результатов.

1.5. Общие сведения о программировании микропроцессоров

Команда - единица информации, указывающая МП, какую операцию необходимо выполнить. Поименованный набор команд составляет программу.

Существует два способа программирования:

1. Машинные коды - ввод кода каждой команды в соответствующие ячейки памяти.

2. Ассемблер - не что иное, как обычная подстановка вместо кодов команд их мнемонического условного наименования. Для разработки программы для микроконтроллера на ассемблере обычно требуется компьютер, а также две программы - текстовый редактор и собственно компилятор ассемблера. Компилятор - это программа, осуществляющая перевод тестовой программы на языке ассемблера в понятные микроконтроллеру коды. Текст самой программы создается в обычном редакторе, как обычный текстовый файл.

Форматом команды называется ее конструкция со спецификацией частей. Команда состоит из кода операции (КОП), определяющего функцию команды, и/или параметров, уточняющих эту функцию.

Пример программы на ассемблере:

```
LDI R16, $60
LDI R17, $7A
LDS R1, $0083
ADD R1, R16
ADD R1, R17
LOOP:
RJMP LOOP
```

В данной программе в первых двух командах осуществляется запись двух чисел в регистры общего назначения R16, R17, в третьей команде в регистр R1 записывается число из ячейки ОЗУ с номером 0083. Далее осуществляется попарно сложение записанных в регистры чисел. В последних двух строках

осуществляется «зацикливание» программы. Знак «\$» указывает на то, что данные записаны в шестнадцатеричной системе счисления.

Пример позволяет сделать несколько выводов: большинство команд связано с передачей информации из одной части системы в другую; некоторые команды служат только для этого или определяют местонахождение какой-либо информации. Поэтому очень важным для вычислительного устройства, каковым является МП, наличие надежных и гибких способов идентификации местонахождения информации, называемых режимами адресации. К числу основных режимов адресации относятся следующие.

Прямая адресация - адрес определяется как часть команды.

Регистровая адресация - операнд (число, над которым осуществляется операция) находится в регистре, а адрес регистра является частью команды.

Косвенная адресация - в ячейке, адрес которой является частью команды, находится адрес операнда.

Базовая адресация - адрес образуется сложением содержимого ячейки памяти или регистра с определенным числом, называемым смещением.

Инкрементация/декрементация - обращение к смежным или равномерно распределенным адресам.

Относительная адресация - адрес равен сумме числа и текущего содержимого РС.

Непосредственная адресация - операнд содержится непосредственно в команде.

Рассмотрим примеры основных видов программных операций.

1. Передача информации.

```
LDI R16, $60
MOV R1, R16
STS $0140, R1
```

2. Арифметические операции.

Пример. Сложить два числа из ячеек памяти 0409 и 0515, от суммы отнять содержимое 0090 и результат записать в 0100.

```
LDS R1, $0409
LDS R2, $0515
LDS R3, $0090
ADD R1, R2
SUB R1, R3
STS $0100, R1
LOOP:
RJMP LOOP
```

3. Логические операции.

Наиболее распространены операции И, ИЛИ, НЕ, искл. ИЛИ.

Лог. операции часто применяют для селективного изменения отдельных бит в байте. Эти действия осуществляются с помощью логических операций над нужным байтом и вторым байтом, называемом маской.

LDS R1, \$0409

LDS R2, \$0515

OR R1, R2

4. Переходы.

Обычно команды выбираются из последовательных ячеек памяти. Для управления ходом выполнения программы используются команды условного и безусловного переходов. Безусловные переходы заставляют программу совершить переход вне зависимости от условий работы МП, условные – дают возможность принимать решения. Пример команды безусловного перехода:

LDS R1, \$0409

LDS R2, \$0515

LDS R3, \$0090

ADD R1, R2

JMP \$6

5. Организация циклов.

Процедура повторяющегося выполнения одного и того же набора команд называется циклом. Цикл - один из основных программных объектов, реализующий возможность компьютерной техники выполнять огромные объемы рутинной вычислительной работы.

6. Подпрограммы.

В случае, когда один и тот же сегмент программы необходимо использовать в произвольных ее местах (к примеру вычисление синуса), этот сегмент удобно оформить в виде автономной программы.

Организованный таким образом автономный программный сегмент называется подпрограммой, переход к ней называется вызовом подпрограммы (рис. 4).

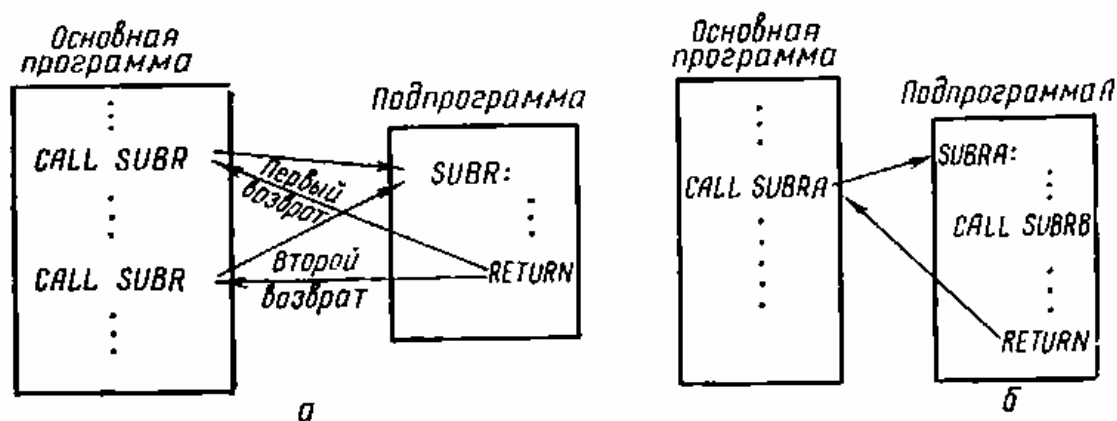


Рис. 4. Взаимодействие основной программы и подпрограммы

При вызове подпрограммы требуется ответить на несколько вопросов:

1. Где запомнить текущее содержимое РС, чтобы подпрограмма знала, куда возвращаться? - Автоматически запоминается в стеке при использовании команды вызова подпрограммы CALL
2. К какому адресу перейти? - Задается как параметр команды CALL
3. Как подпрограмма найдет адрес возврата? - С помощью команды RETURN
4. Как и где запомнить текущее содержимое РОНов, чтобы оно не разрушалось подпрограммой? - Команда CALL автоматически отправляет содержимое РОНов в стек.
5. Как сообщить подпрограмме адрес преобразуемых данных? - Параметры CALL.

Стек – это область памяти в ОЗУ с упрощенной адресацией, предназначена для хранения временных данных при вызове подпрограмм и организации прерываний.

1.6. Вопросы для самопроверки

1. Основные этапы эволюции МП.
2. Классификация МП по числу БИС в комплекте.
3. Классификация МП по назначению.
4. Классификация МП по виду обрабатываемых входных сигналов.
5. Отличие синхронных МП от асинхронных.
6. Основные характеристики МП.

7. Способы повышения производительности МП.
8. Перечислите этапы разработки МП.
9. Для чего применяется тестовое диагностирование?
10. Что такое программа?
11. Перечислите типовые программные процедуры.
12. Для чего предназначен стек?

2. Архитектура микропроцессоров

2.1. Обобщенная структура микропроцессора

Архитектура МП - общая конфигурация его компонент, главные их характеристики и способы соединения.

Понятие архитектуры микропроцессора отражает:

- его структуру, т.е. совокупность компонентов, составляющих микропроцессор, и связей между ними; для пользователя достаточно ограничиться регистровой моделью микропроцессора;
- способы представления и форматы данных;
- способы обращения ко всем программно-доступным для пользователя элементам структуры (адресация к регистрам, ячейкам постоянной и оперативной памяти, внешним устройствам);
- набор операций, выполняемых микропроцессором;
- характеристики управляющих слов и сигналов, вырабатываемых микропроцессором и поступающих в него извне;
- реакцию на внешние сигналы (система обработки прерываний и т.п.).

Существует несколько подходов к классификации микропроцессоров по типу архитектуры. Так, выделяют МП с CISC (Complete Instruction Set Computer) архитектурой, характеризуемой полным набором команд, и RISC (Reduce Instruction Set Computer) архитектурой, которая определяет систему с сокращенным набором команд одинакового формата, выполняемых за один такт МП.

По способу организации пространства памяти микропроцессорной системы различают два основных типа архитектур.

Организация, при которой для хранения программ и данных используется одно пространство памяти, называется **фон Неймановской архитектурой** (по имени математика, предложившего кодирование программ в формате, соответствующем формату данных). Программы и данные хранятся в едином пространстве, и нет никаких признаков, указывающих на тип информации в ячейке

памяти. Преимуществами такой архитектуры являются более простая внутренняя структура микропроцессора и меньшее количество управляющих сигналов.

Организация, при которой память программ CSEG (Code Segment) и память данных DSEG (Data Segment) разделены и имеют свои собственные адресные пространства и способы доступа к ним, называется **Гарвардской архитектурой** (по имени лаборатории Гарвардского Университета, предложившей ее). Такая архитектура является более сложной и требует дополнительных управляющих сигналов. Однако она позволяет осуществлять более гибкие манипуляции информацией, реализовывать компактно кодируемый набор машинных команд и, в ряде случаев, ускорять работу микропроцессора.

В настоящее время выпускаются микропроцессоры со смешанной архитектурой, в которых CSEG и DSEG имеют единое адресное пространство, однако различные механизмы доступа к ним.

На физическом уровне микропроцессор взаимодействует с памятью и системой ввода-вывода через единый набор системных шин - внутрисистемную магистраль. Она состоит из:

- шины данных DB (Data Bus), по которой производится обмен данными между ЦП, памятью и системой ВВ;
- шины адреса AB (Address Bus), используемой для передачи адресов ячеек памяти и портов ВВ, к которым осуществляется обращение;
- шины управления CB (Control Bus), по которой передаются управляющие сигналы, реализующие циклы обмена информацией и управляющие работой системы.

Этот же набор шин применяется для организации канала ПДП. Магистраль такого типа носит название демультимплексной или трехшинной с отдельными шинами адреса и данных.

В некоторых микропроцессорах с целью сокращения ширины физической магистрали вводят совмещенную шину адреса-данных AD (Address/Data Bus), по которой передаются как адреса так и данные. Этап передачи адресной информации отделен по времени от этапа передачи данных и стробируется специальным сигналом ALE (Address Latch Enable), который включен в состав CB. Данную магистраль обычно называют мультимплексной или двухшинной с совмещенными шинами адреса и данных.

Поскольку внутренняя архитектура МП определяется его местом в составе вычислительного устройства, рассмотрим обобщенную структуру микропроцессора (рис. 5) и определим функции его частей.

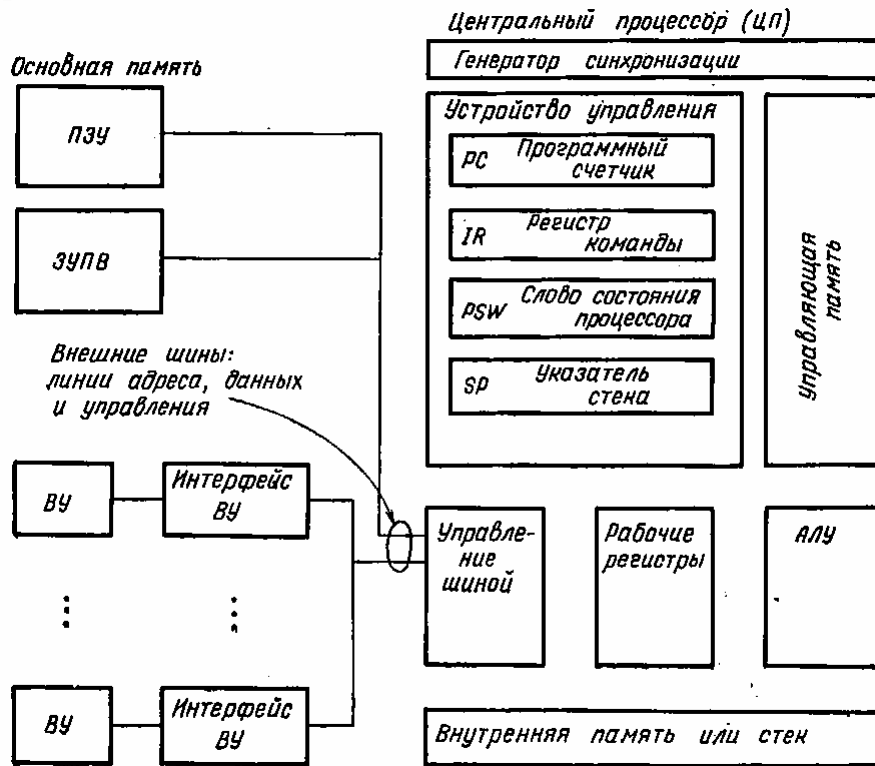


Рис. 5. Обобщенная структура микропроцессора

Устройство управления предназначено для выборки команд из ячеек внешней памяти, их интерпретации и выполнения. Обычно содержит следующие регистры:

- PC - программный счетчик
- IR - регистр команд
- PSW - регистр признаков
- SP - указатель стека

Управляющая память содержит последовательность микрокоманд, образующих команду. Может быть доступна или недоступна пользователю.

Рабочие регистры - ЗУ небольшого объема, предназначенное для хранения адресов и данных. В том случае, когда и адреса и данные хранятся в одних и тех же рабочих регистрах, эти регистры называются **РОНами** (Регистры Общего Назначения).

АЛУ - арифметико-логическое устройство, выполняет арифметические и логические операции над содержимым РОНов, ячеек памяти и РС.

Управление шиной - координирует взаимодействие центрального процессора с ВУ

Внутренняя память - память небольшого объема внутри центрального процессора. Основное отличие от РОНов - способ адресации.

Генератор синхронизации - формирует согласованные во времени последовательности импульсов для координации всех действий компьютера.

Оперативная память (ОЗУ, ЗУПВ) - предназначена для команд и данных, используемых в текущих операциях, обычно быстродействующая, энергозависимая и сравнительно небольшого объема.

Постоянная память (ПЗУ) - обычно память, допускающая только считывание или запись/считывание с относительно невысоким быстродействием, энергонезависимая. Используется для хранения общеупотребительных программ и данных.

Интерфейсы - устройства сопряжения ВУ с каналом обмена компьютера.

Кроме CSEG и DSEG практически все современные микропроцессоры имеют специально выделенное пространство данных небольшого объема, называемое набором программно-доступных регистров RSEG (Register Segment). В отличие от CSEG и DSEG регистры RSEG располагаются внутри ЦП в непосредственной близости от его АЛУ, что обеспечивает быстрый физический доступ к информации, хранящейся в них. В них, как правило, хранятся промежуточные результаты вычислений, часто используемые ЦП. Область RSEG может быть полностью изолирована от пространства данных DSEG, может частично пересекаться с ней, и может полностью являться подчастью DSEG. Внутренняя логическая организация RSEG очень разнообразна и играет важную роль при классификации архитектур микропроцессоров.

Регистры микропроцессора функционально неоднородны: одни служат для хранения данных или адресной информации, другие - для управления работой ЦП. В соответствии с этим все регистры можно разделить на регистры данных, указатели и регистры специального назначения. Регистры данных участвуют в арифметических и логических операциях в качестве источников операндов и приемников результата, адресные регистры или указатели используются для вычисления адресов данных и команд, расположенных в основной памяти. Специальные регистры служат для индикации текущего состояния ЦП и управления работой его составных частей. Способы использования того или иного вида регистров определяют конкретные особенности архитектуры микропроцессора.

Среди регистров данных часто в некоторых архитектурах выделяют один регистр, называемый аккумулятором А (Accumulator), с которым связывают большинство команд арифметической и логической обработки данных. Это означает, что арифметические и логические команды используют в качестве одного из своих операндов содержимое аккумулятора и сохраняют в нем результат операции. Ссылка на него производится неявно с помощью кода операции. При

этом нет необходимости в коде команды выделять специальную область для адресов операнда и результата. Такой тип архитектуры микропроцессора называется аккумуляторным. К недостаткам такой архитектуры можно отнести относительно низкое быстродействие, объясняемое тем, что аккумулятор является "узким местом", в которое каждый раз необходимо сначала занести операнд перед выполнением операции. Примером такой архитектуры могут служить микроконтроллеры семейства MCS-51 фирмы Intel.

Другим примером организации регистров данных являются т.н. "рабочие регистры" R0, R1 и т.д. В этом случае операнды и результаты арифметических и логических операций могут храниться не в одном, а в нескольких регистрах, что расширяет возможности по манипуляции данными. В отличие от аккумулятора рабочие регистры адресуются явно в коде команды. Такой тип архитектуры микропроцессора называется регистровым. Примером такой организации могут служить микропроцессоры семейства 80x86 фирмы Intel. В ряде микропроцессоров, предназначенных для работы в реальном масштабе времени, предусмотрены не один, а несколько наборов рабочих регистров. В каждый момент времени доступен лишь один из наборов регистров, выбор которого осуществляется записью соответствующей информации в определенный служебный регистр. Примером таких устройств могут служить микроконтроллеры семейства MCS-48 фирмы Intel.

Архитектура, при которой процессор способен использовать в качестве адресов операндов и результатов операции ячейки основной памяти, называется архитектурой типа "память - память". При этом исключаются временные затраты на перепись содержимого рабочих регистров при переходе от одной процедуры к другой. Однако при этом теряется быстрый доступ к промежуточным данным, т.к. они хранятся не во внутренних регистрах, а в DSEG. Решением этой проблемы может служить размещение части DSEG на одном кристалле с ЦП и использование в качестве рабочих областей этого внутреннего сегмента ОЗУ. Примером такой организации могут служить микроконтроллеры семейства MCS-96 фирмы Intel.

Практически во всех современных микропроцессорах выделяют отдельную область памяти под так называемый "стек", используемый, в общем случае, для передачи параметров процедурам и сохранения адресов возврата из них. Стек может быть расположен внутри микропроцессора или вне его. Он может занимать часть адресного пространства DSEG или RSEG, а может быть расположен и отдельно от них. В последнем случае говорят о т.н. "аппаратном стеке". Передача функций аккумулятора вершине стека приводит к "стековой архитектуре". Стековая организация дает возможность использовать безадресные команды, код которых имеет наименьшую длину. Безадресные команды оперируют данными, находящимися на вершине стека и непосредственно под ней. При выполнении операции исходные операнды извлекаются из стека, а результат передается не

вершину стека. Стековая архитектура обладает высокой вычислительной эффективностью. Существует специальный язык высокого уровня FORTH, построенный на основе безадресных команд. Такая архитектура используется в специализированных процессорах высокой производительности и, в частности в RISC-процессорах.

Служебные регистры, расположенные внутри микропроцессора, предназначены для различных функций управления его работой и индикации состояния его составных частей. Их состав и организация зависят от конкретной архитектуры процессора и различаются в каждом конкретном случае. Наиболее часто встречающимися регистрами специальных функций являются “программный счетчик” PC (Program Counter), “указатель стека” SP (Stack Pointer) и “слово состояния программы” PSW (Program Status Word). Программный счетчик PC в каждый конкретный момент времени содержит адрес команды, следующей в CSEG за той, которая в данный момент выполняется. Указатель стека SP хранит текущий адрес вершины стека. Слово состояния программы PSW содержит набор текущих признаков результата выполнения операции. С каждым признаком результата связывается одноразрядная переменная-флажок, соответствующая определенному биту PSW. К типовым флажкам-признакам относятся:

- CF (Carry Flag) - флажок переноса из старшего разряда АЛУ. Равен 1, если в результате выполнения арифметической операции или операции сдвига произошел перенос из старшего разряда результата;

- ZF (Zero Flag) - флажок признака нуля. Равен 1, если результат операции равен 0;

- SF (Sign Flag) - флажок знака результата. Дублирует знаковый разряд результата операции;

- AF (Auxiliary Carry Flag) - флажок дополнительного переноса. Равен 1, если в результате выполнения арифметической операции или операции сдвига произошел перенос из младшей тетрады результата в старшую. Часто используется в двоично-десятичной арифметике;

- OF (Overflow Flag) - флажок переполнения. Равен 1, если в результате выполнения арифметической операции произошло переполнение разрядной сетки результата;

- PF (Parity Flag) - флажок четности. Равен 1, если число 1 в результате операции нечетно и наоборот.

- IF (Interrupt Flag) - флажок разрешения прерывания. Индицирует, разрешены ли прерывания в системе.

Конкретные флаги используются программой для анализа результата предшествующей команды и принятия решения о дальнейшем ходе выполнения программы. Специальные регистры могут занимать часть адресного пространства DSEG или RSEG, а могут быть расположены и отдельно от них.

Разрядностью микропроцессорной системы принято считать количество бит информации, которое ее ЦП может обработать с помощью одной команды. Разрядность микропроцессора определяется разрядностью его АЛУ, внутренних регистров данных и внешней шины данных. На сегодняшний день существуют 8-, 16-, 32- и 64-разрядные микропроцессоры. Для того, чтобы обрабатывать информацию с разрядностью большей, чем разрядность микропроцессора необходимо реализовывать специальные алгоритмы вычислений с повышенной разрядностью. Эти алгоритмы требуют дополнительного времени для своего выполнения. Поэтому повышение разрядности микропроцессора при заданной разрядности вычислений напрямую связано с увеличением быстродействия системы.

Существуют микропроцессоры, архитектура которых адаптирована для выполнения вычислений определенного рода. К числу таких процессоров относятся “процессоры цифровой обработки сигналов” DSP (Digital Signal Processor). Их архитектура имеет особенности, позволяющие им с наибольшей производительностью осуществлять алгоритмы рекуррентной обработки данных, которые используются во многих задачах, требующих их выполнения в масштабе “реального времени”, таких как аудио- и видео-кодирование, регулирование, цифровая фильтрация, цифровая связь и т.п. Все эти процессоры построены, как правило, по Гарвардской архитектуре. Современные DSP имеют отдельные шины адреса/данных для CSEG и DSEG, что позволяет им с помощью одной команды осуществить доступ к различным видам памяти и произвести несколько операций над данными. Основной особенностью DSP является то, что кроме обыкновенного АЛУ, которое присутствует во всех процессорах, они имеют еще несколько вычислительных устройств. К числу таких устройств в первую очередь относится “умножитель-аккумулятор” MAU (Multiple-Accumulator Unit), способный с помощью одной команды умножить два многоразрядных числа и сложить результат удвоенной разрядности с результатом предыдущей команды. Подобная операция “умножения-сложения” используется во всех рекуррентных алгоритмах. Наличие MAU в сочетании с вышеуказанными особенностями организации шин процессора позволяет DSP за одну команду полностью выполнить один шаг рекуррентного алгоритма и подготовить исходные данные для следующего шага. Другим дополнительным вычислительным устройством является “многоразрядный регистр сдвига” S (Shifter), способный выполнять операции сдвига над числами, разрядность которых превышает разрядность АЛУ. Совместная работа этих вычислительных устройств позволяет достичь на выполнении рекуррентных алгоритмов вычислительной производительности, несравнимой с любыми другими процессорами.

2.2. Логическая структура микропроцессора

Логическая структура микропроцессора, т. е. конфигурация составляющих микропроцессор логических схем и связей между ними, определяется функциональным назначением. Именно структура задает состав логических блоков микропроцессора и то, как эти блоки должны быть связаны между собой, чтобы полностью отвечать архитектурным требованиям. Срабатывание электронных блоков микропроцессора в определенной последовательности приводит к выполнению заданных архитектурой микропроцессора функций, т. е. к реализации вычислительных алгоритмов. Одни и те же функции можно выполнить в микропроцессорах со структурой, отличающейся набором, количеством и порядком срабатывания логических блоков. Различные структуры микропроцессоров, как правило, обеспечивают их различные возможности, в том числе и различную скорость обработки данных. Логические блоки микропроцессора с развитой архитектурой представлены на рис. 6.

На рис. 6 приняты следующие обозначения:

БУПК - блок управления последовательностью команд;

БУВО - блок управления выполнением операций;

БУФКА - блок управления формированием кодов адресов;

БУВП - блок управления виртуальной памятью;

БЗП - блок защиты памяти; БУПП - блок управления прерыванием работы процессора;

БУВВ - блок управления вводом выводом;

РСОЗУ - регистровое сверхоперативное запоминающее устройство;

АЛБ - арифметико-логический блок;

БДА - блок дополнительной арифметики;

БС - блок синхронизации.



Рис. 6. Логическая структура микропроцессора

При проектировании логической структуры микропроцессоров необходимо рассмотреть:

- номенклатуру электронных блоков, необходимую и достаточную для реализации архитектурных требований;
- способы и средства реализации связей между электронными блоками;
- методы отбора если не оптимальных, то наиболее рациональных вариантов логических структур из возможного числа структур с отличающимся составом блоков и конфигурацией связей между ними.

При проектировании микропроцессора приводятся в соответствие внутренняя сложность кристалла и количество выводов корпуса. Относительный рост числа элементов по мере развития микроэлектронной технологии во много раз превышает относительное увеличение числа выводов корпуса, поэтому проектирование БИС в виде конечного автомата, а не в виде набора схем, реализующих некоторый набор логических переключательных функций и схем памяти, дает возможность получить функционально законченные блоки и устройства МП.

Использование микропроцессорных комплектов БИС позволяет создать МП для широких областей применения вследствие программной адаптации МП к конкретной области применения: изменяя программу работы микропроцессора, изменяют функции информационно-управляющей системы. Поэтому за счет составления программы работы МП в конкретных условиях работы определенной системы можно получить оптимальные характеристики последней.

Если уровень только программной "настройки" МП не позволит получить эффективную систему, доступен следующий уровень проектирования - микропрограммный. За счет изменения содержимого ПЗУ или программируемой логической матрицы (ПЛМ) можно "настроиться" на более специфичные черты системы обработки информации. В этом случае частично за счет изменения микропрограмм затрагивается аппаратный уровень системы. Техничко-экономические последствия здесь связаны лишь с ограниченным вмешательством в технологию изготовления управляющих блоков МП.

Изменение аппаратного уровня информационно-управляющей микропроцессорной системы, включающего в себя функциональные БИС комплекта, одновременно с конкретизацией микропрограммного и программного уровней позволяет наилучшим образом удовлетворить требованиям, предъявляемым к системе.

Решение задач управления в конкретной системе чисто аппаратными средствами (аппаратная логика) дает выигрыш в быстродействии, однако приводит к сложностям при модификации системы. Микропроцессорное решение (программная логика) является более медленным, но более гибким решением, позволяющим развивать и модифицировать систему. Изменение технических требований к информационно-управляющей микропроцессорной системе ведет лишь к необходимости перепрограммирования работы микропроцессора. Именно это качество обеспечивает высокую логическую гибкость микропроцессоров, определяет возможность их широкого использования, а значит и крупносерийного производства.

2.3. Организация ввода-вывода

Передача данных между МП и внешними устройствами (ВУ) называется вводом-выводом (ВВ). Для организации ВВ используются 3 основных способа: программный ВВ, ВВ по прерываниям и прямой доступ к памяти. Первый способ характеризуется тем, что инициирование и управление ВВ осуществляется программой, выполняемой процессором, а внешние устройства играют сравнительно пассивную роль и сигнализируют только о своем состоянии, в частности, о готовности к операциям ввода/вывода. Во втором режиме ВВ иницируется не процессором, а внешним устройством, генерирующим специальный сигнал прерывания. Реагируя на этот сигнал готовности устройства к передаче данных, процессор передает управление подпрограмме обслуживания устройства, вызвавшего прерывание. Действия, выполняемые этой подпрограммой, определяются пользователем, а непосредственными

операциями ВВ управляет процессор. Наконец, в режиме прямого доступа к памяти, который используется, когда пропускной способности процессора недостаточно, действия процессора приостанавливаются, он отключается от системной шины и не участвует в передачах данных между основной памятью и быстродействующим ВУ. Заметим, что во всех вышеуказанных случаях основные действия, выполняемые на системной магистрали МП подчиняются двум основным принципам.

1. В процессе взаимодействия любых двух устройств МП одно из них обязательно выполняет активную, управляющую роль и является заказчиком. Второе оказывается управляемым, исполнителем. Чаще всего заказчиком является процессор.

2. Другим важным принципом, заложенным в структуру интерфейса, является принцип квитирования (запроса - ответа): каждый управляющий сигнал, посланный заказчиком, подтверждается сигналом исполнителя. При отсутствии ответного сигнала исполнителя в течение заданного интервала времени формируется тайм-аут, заказчиком фиксируется ошибка обмена и прекращается данная операция.

Рассмотрим подробнее данные способы ВВ.

2.3.1. Программный ввод-вывод

Программный ВВ - самый простой способ, требует минимума дополнительной логики, выполняется полностью под управлением МП. В ВУ выделяются 2 регистра - регистр готовности и буферный регистр. МП обращается сначала к регистру готовности и, обнаружив там признак наличия таковой, начинает обмен через буферный регистр (рис. 7)

Основное достоинство программного ВВ - его простота, но при этом МП тратит время на ожидании готовности (доля времени полезной работы может составлять менее 1%).

2.3.2. Ввод-вывод по прерываниям

Одной из разновидностей программно-управляемого обмена данными с ВУ в МП является обмен с прерыванием программы, отличающийся от асинхронного программного ВВ тем, что переход к выполнению команд, физически реализующих обмен данными, осуществляется с помощью специальных аппаратных средств. Команды обмена данными в этом случае

выделяют в отдельный программный модуль - подпрограмму обработки прерывания.

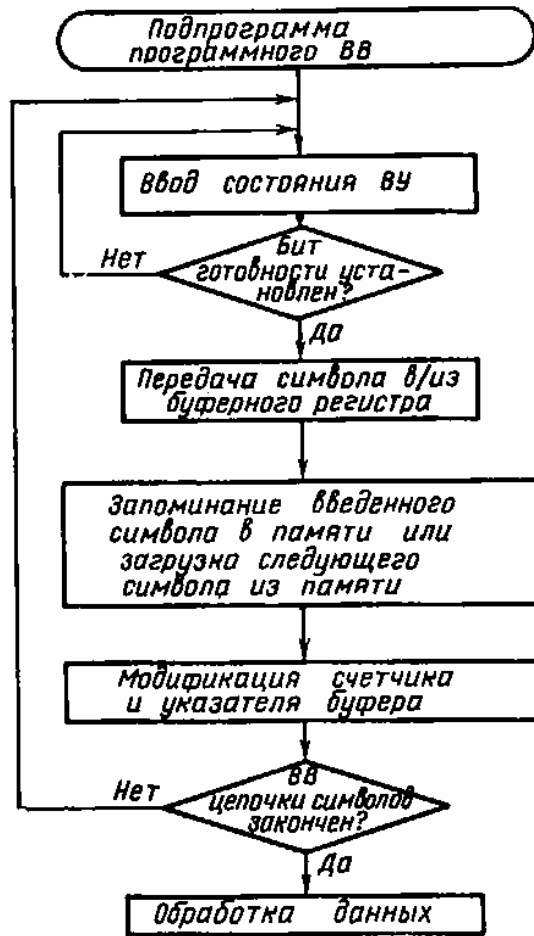


Рис. 7. Программный ввод-вывод

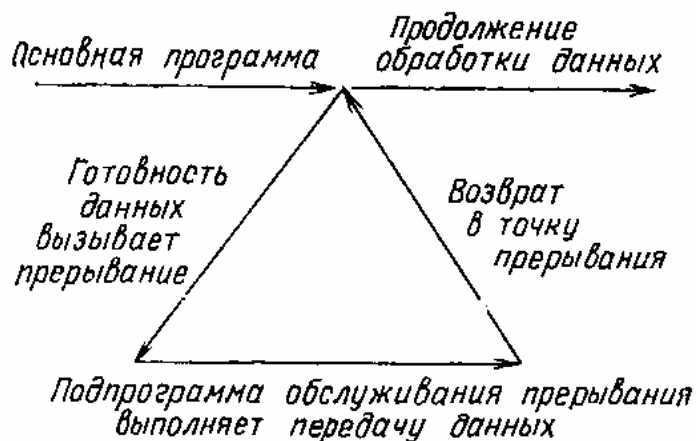


Рис. 8. Ввод-вывод по прерыванию

Работу МП можно разделить во времени между 2-мя независимыми программами - программа заднего плана обработки данных и программа ВВ переднего плана (рис. 8). При этом вторая программа производит прерывание первой.

Задачей аппаратных средств обработки прерывания в МП является приостановка выполнения одной программы (ее еще называют основной программой) и передача управления подпрограмме обработки прерывания. Действия, выполняемые при этом процессором, как правило, те же, что и при обращении к подпрограмме. Только при обращении к подпрограмме они иницируются командой, а при обработке прерывания - управляющим сигналом от ВУ, который называют "Запрос на прерывание" или "Требование прерывания".

Эта важная особенность обмена с прерыванием программы позволяет организовать обмен данными с ВУ в произвольные моменты времени, не зависящие от программы, выполняемой в МП. Таким образом, появляется возможность обмена данными с ВУ в реальном масштабе времени, определяемом внешней по отношению к МП средой. Обмен с прерыванием программы существенным образом экономит время процессора, затрачиваемое на обмен. Это происходит за счет того, что исчезает необходимость в организации программных циклов ожидания готовности ВУ, на выполнение которых тратится значительное время, особенно при обмене с медленными ВУ.

Прерывание программы по требованию ВУ не должно оказывать на прерванную программу никакого влияния, кроме увеличения времени ее выполнения за счет приостановки на время выполнения подпрограммы обработки прерывания. Поскольку для выполнения подпрограммы обработки прерывания используются различные регистры процессора (счетчик команд, регистр состояния и т.д.), то информацию, содержащуюся в них в момент прерывания, необходимо сохранить для последующего возврата в прерванную программу.

Обычно задача сохранения содержимого счетчика команд и регистра состояния процессора возлагается на аппаратные средства обработки прерывания. Сохранение содержимого других регистров процессора, используемых в подпрограмме обработки прерывания, производится непосредственно в подпрограмме. Отсюда следует достаточно очевидный факт: чем больший объем информации о прерванной программе сохраняется программным путем, тем больше время реакции МП на сигнал прерывания, и наоборот. Предпочтительными с точки зрения повышения производительности МП (сокращения времени выполнения подпрограмм обработки, а

следовательно, и основной программы) являются уменьшение числа команд, обеспечивающих сохранение информации о прерванной программе, и реализация этих функций аппаратными средствами.

Формирование сигналов прерываний - запросов ВУ на обслуживание происходит в контроллерах соответствующих ВУ. В простейших случаях в качестве сигнала прерывания может использоваться сигнал "Готовность ВУ", поступающий из контроллера ВУ в системный интерфейс МП. Однако такое простое решение обладает существенным недостатком - процессор не имеет возможности управлять прерываниями, т. е. разрешать или запрещать их для отдельных ВУ.

Для решения этой проблемы регистр состояния и управления контроллера ВУ дополняют еще одним разрядом - "Разрешение прерывания". Запись 1 или 0 в разряд "Разрешение прерывания" производится программным путем по одной из линий шины данных системного интерфейса. Управляющий сигнал системного интерфейса "Запрос на прерывание" формируется с помощью схемы совпадения только при наличии единиц в разрядах "Готовность ВУ" и "Разрешение прерывания" регистра состояния и управления контроллера.

Аналогичным путем решается проблема управления прерываниями в МП в целом. Для этого в регистре состояния процессора выделяется разряд, содержимое которого определяет, разрешены или запрещены прерывания от внешних устройств. Значение этого разряда может устанавливаться программным путем.

В МП обычно используется одноуровневая система прерываний, т. е. сигналы "Запрос на прерывание" от всех ВУ поступают на один вход процессора. Поэтому возникает проблема идентификации ВУ запросившего обслуживание, и реализации заданной очередности (приоритета) обслуживания ВУ при одновременном поступлении нескольких сигналов прерывания. Существуют два основных способа идентификации ВУ, запросивших обслуживания:

- программный опрос регистров состояния (разряд "Готовность ВУ") контроллеров всех ВУ;
- использование векторов прерывания.

Организация прерываний с программным опросом готовности предполагает наличие в памяти МП единой подпрограммы обслуживания прерываний от всех внешних устройств.

Обслуживание ВУ с помощью единой подпрограммы обработки прерываний производится следующим образом. В конце последнего машинного цикла выполнения очередной команды основной программы процессор проверяет наличие требования прерывания от ВУ. Если сигнал прерывания есть

и в процессоре прерывание разрешено, то процессор переключается на выполнение подпрограммы обработки прерываний.

После сохранения содержимого регистров процессора, используемых в подпрограмме, начинается последовательный опрос регистров состояния контроллеров всех ВУ, работающих в режиме прерывания. Как только подпрограмма обнаружит готовое к обмену ВУ, сразу выполняются действия по его обслуживанию. Завершается подпрограмма обработки прерывания после опроса готовности всех ВУ и восстановления содержимого регистров процессора.

2.3.3. Прямой доступ к памяти

Одним из способов обмена данными с ВУ является обмен в режиме прямого доступа к памяти (ПДП). В этом режиме обмен данными между ВУ и основной памятью МП происходит без участия процессора (рис. 9). Обменом в режиме ПДП управляет не программа, выполняемая процессором, а электронные схемы, внешние по отношению к процессору. Обычно схемы, управляющие обменом в режиме ПДП, размещаются в специальном контроллере, который называется контроллером прямого доступа к памяти.

Обмен данными в режиме ПДП позволяет использовать в МП быстродействующие внешние запоминающие устройства, такие, например, как накопители на жестких магнитных дисках, поскольку ПДП может обеспечить время обмена одним байтом данных между памятью и ВЗУ, равное циклу обращения к памяти.

Для реализации режима прямого доступа к памяти необходимо обеспечить непосредственную связь контроллера ПДП и памяти МП. Для этой цели можно было бы использовать специально выделенные шины адреса и данных, связывающие контроллер ПДП с основной памятью. Но такое решение нельзя признать оптимальным, так как это приведет к значительному усложнению МП в целом, особенно при подключении нескольких ВУ. В целях сокращения количества линий в шинах МП контроллер ПДП подключается к памяти посредством шин адреса и данных системного интерфейса. При этом возникает проблема совместного использования шин системного интерфейса процессором и контроллером ПДП. Можно выделить два основных способа ее решения: реализация обмена в режиме ПДП с "захватом цикла" и в режиме ПДП с блокировкой процессора.

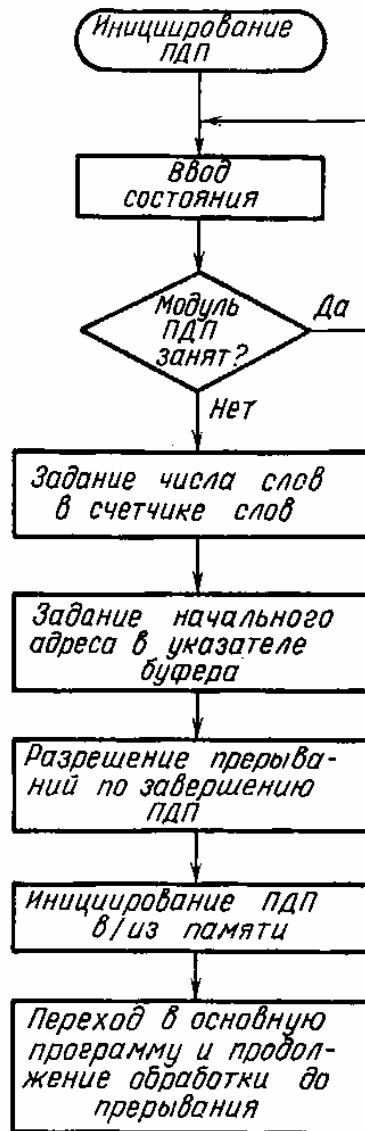


Рис. 9. Прямой доступ к памяти

Существуют две разновидности прямого доступа к памяти с "захватом цикла". Наиболее простой способ организации ПДП состоит в том, что для обмена используются те машинные циклы процессора, в которых он не обменивается данными с памятью. В такие циклы контроллер ПДП может обмениваться данными с памятью, не мешая работе процессора. Однако возникает необходимость выделения таких циклов, чтобы не произошло временного перекрытия обмена ПДП с операциями обмена, инициируемыми процессором. В некоторых процессорах формируется специальный управляющий сигнал, указывающий циклы, в которых процессор не обращается

к системному интерфейсу. При использовании других процессоров для выделения таких циклов необходимо применение в контроллерах ПДП специальных селектирующих схем, что усложняет их конструкцию. Применение рассмотренного способа организации ПДП не снижает производительности МП, но при этом обмен в режиме ПДП возможен только в случайные моменты времени одиночными байтами или словами.

Более распространенным является ПДП с "захватом цикла" и принудительным отключением процессора от шин системного интерфейса. Для реализации такого режима ПДП системный интерфейс МП дополняется двумя линиями для передачи управляющих сигналов "Требование прямого доступа к памяти" (ТПДП) и "Предоставление прямого доступа к памяти" (ППДП).

Управляющий сигнал ТПДП формируется контроллером прямого доступа к памяти. Процессор, получив этот сигнал, приостанавливает выполнение очередной команды, не дожидаясь ее завершения, выдает на системный интерфейс управляющий сигнал ППДП и отключается от шин системного интерфейса. С этого момента все шины системного интерфейса управляются контроллером ПДП. Контроллер ПДП, используя шины системного интерфейса, осуществляет обмен одним байтом или словом данных с памятью МП и затем, сняв сигнал ТПДП возвращает управление системным интерфейсом процессору. Как только контроллер ПДП будет готов к обмену следующим байтом, он вновь "захватывает" цикл процессора и т.д. В промежутках между сигналами ТПДП процессор продолжает выполнять команды программы. Тем самым выполнение программы замедляется, но в меньшей степени, чем при обмене в режиме прерываний.

Применение в МП обмена данными с ВУ в режиме ПДП всегда требует предварительной подготовки, а именно: для каждого ВУ необходимо выделить область памяти, используемую при обмене, и указать ее размер, т.е. количество записываемых в память или читаемых из памяти байт (слов) информации. Следовательно, контроллер ПДП должен обязательно иметь в своем составе регистр адреса и счетчик байт (слов). Перед началом обмена с ВУ в режиме ПДП процессор должен выполнить программу загрузки. Эта программа обеспечивает запись в указанные регистры контроллера ПДП начального адреса выделенной ВУ памяти и ее размера в байтах или словах в зависимости от того, какими порциями информации ведется обмен. Сказанное не относится к начальной загрузке программ в память в режиме ПДП. В этом случае содержимое регистра адреса и счетчика байт слов устанавливается переключателями или перемычками непосредственно на плате контроллера.

2.4. Запоминающие устройства микропроцессоров

2.4.1. Виды и параметры запоминающих устройств

Запоминающие устройства (ЗУ) служат для хранения информации и обмена ею с другими цифровыми устройствами. Микросхемы и системы памяти постоянно совершенствуются. В наиболее развитой иерархии памяти можно выделить следующие уровни:

- **регистровые ЗУ**, находящиеся в составе процессора или других устройств, благодаря которым уменьшается число обращений к другим уровням памяти, реализованным вне процессора и требующим большего времени для операций обмена информацией;

- **кэш-память**, служащая для хранения копий информации, используемой в текущих операциях обмена. Высокое быстродействие кэш-памяти повышает производительность МП;

- **основная память (оперативная, постоянная)**, работающая в режиме непосредственного обмена с процессором и по возможности согласованная с ним по быстродействию. Исполняемый в текущий момент фрагмент программы обязательно находится в основной памяти;

- **специализированные виды памяти**, характерные для некоторых специфических архитектур (например, видеопамять и др.);

- **внешняя память**, хранящая большие объемы информации. Данная память обычно реализуется на основе устройств с подвижным носителем информации (магнитные и оптические диски, внешние носители информации, выполненные по технологии Flash-памяти и т.д.).

Рассмотрим наиболее важные параметры ЗУ.

Информационная емкость — максимально возможный объем хранимой информации. Выражается в битах или словах (в частности, в байтах). Бит хранится запоминающим элементом (ЗЭ), а слово — запоминающей ячейкой (ЗЯ), т. е. группой ЗЭ, к которым возможно лишь одновременное обращение. Добавление к единице измерения множителя "К" (кило) означает умножение на $2^{10} = 1024$, а множителя "М" (мега) — умножение на $2^{20} = 1048576$.

Быстродействие (производительность) ЗУ оценивают временами считывания, записи и длительностями циклов чтения/записи. Время считывания — интервал между моментами появления сигнала чтения и слова на выходе ЗУ. Время записи — интервал после появления сигнала записи, достаточный для установления ЗЯ в состояние, задаваемое входным словом. Минимально допустимый интервал между последовательными чтениями или записями образует соответствующий цикл.

Важнейшие параметры ЗУ находятся в противоречии. Так, например, большая информационная емкость не сочетается с высоким быстродействием, а быстродействие, в свою очередь не сочетается с низкой стоимостью.

Для классификации ЗУ важнейшим признаком является способ доступа к данным. В данном учебном пособии ограничимся рассмотрением адресных ЗУ, которые в данный момент широко распространены.

Адресные ЗУ делятся на RAM (Random Access Memory) и ROM ((Read Only Memory). Русские синонимы термина RAM: ОЗУ (оперативные ЗУ) или ЗУПВ (ЗУ с произвольной выборкой). Оперативные ЗУ хранят данные, участвующие в обмене при исполнении текущей программы, которые могут быть изменены в произвольный момент времени. Это энергозависимая память, данные пропадают после отключения питания.

В ROM (русский эквивалент — ПЗУ) содержимое либо вообще не изменяется, либо изменяется, но редко и в специальном режиме. Для рабочего режима это «память только для чтения». В ПЗУ данные сохраняются после отключения питания.

2.4.2. Оперативные запоминающие устройства

ОЗУ подразделяются на статические и динамические. В первом варианте запоминающими элементами являются триггеры, сохраняющие свое состояние, когда схема находится под питанием и нет новой записи данных. Во втором варианте данные хранятся в виде зарядов конденсаторов, образуемых элементами МОП структур. Саморазряд конденсаторов ведет к разрушению данных, поэтому они должны периодически (каждые несколько миллисекунд) регенерироваться.

Регенерация данных в динамических ЗУ осуществляется с помощью специальных контроллеров. Разработаны также ЗУ с динамическими запоминающими элементами, имеющие внутреннюю встроенную систему регенерации, у которых внешнее поведение относительно управляющих сигналов становится аналогичным повелению статических ЗУ. Такие ЗУ называют квазистатическими.

Статические ОЗУ называются SRAM (Static RAM), а динамические — DRAM (Dynamic RAM).

Динамические ОЗУ характеризуются наибольшей информационной емкостью и невысокой стоимостью, поэтому именно они используются как основная память. Статические ЗУ в 4...5 раз дороже динамических и приблизительно во столько же раз меньше по информационной емкости.

Их достоинством является высокое быстродействие, а типичная область использования — схемы кэш-памяти.

Статические и динамические ОЗУ чаще всего используются в различных микроконтроллерах. В компьютерах с большими объемами вычислений часто используется такой вид ОЗУ, как синхронная динамическая память с произвольным доступом SDRAM (Synchronous Dynamic Random Access Memory).

Первый стандарт SDRAM с появлением последующих стандартов стал именоваться SDR (Single Data Rate — в отличие от Double Data Rate). За один такт принималась одна управляющая команда и передавалось одно слово данных. На смену памяти типа SDRAM пришла синхронная динамическая память с произвольным доступом и удвоенной скоростью передачи данных DDR SDRAM (Double Data Rate Synchronous Dynamic Random Access Memory). На смену памяти типа DDR SDRAM пришла DDR2 SDRAM, основное отличие от DDR — вдвое большая частота работы шины, по которой данные передаются в буфер микросхемы памяти. Следующее поколение данного вида памяти — это DDR3 SDRAM. Преимущества по сравнению с DDR2:

- более высокая полоса пропускания (до 2400 МГц);
- сниженное тепловыделение (результат уменьшения напряжения питания);
- меньшее энергопотребление и улучшенное энергосбережение.

2.4.3. Постоянные запоминающие устройства

Очень часто в различных применениях требуется хранение информации, которая не изменяется в процессе эксплуатации устройства – программы в микроконтроллерах, начальные загрузчики (BIOS) в компьютерах, таблицы коэффициентов цифровых фильтров в сигнальных процессорах и др.

В ПЗУ типа **ROM (М)** запись информации производится при помощи последней операции производства микросхемы - металлизации. Металлизация производится при помощи маски, поэтому такие ПЗУ получили название масочных (ПЗУМ).

Программирование масочного ПЗУ производится на заводе изготовителе, что очень неудобно для мелких и средних серий производства, не говоря уже о стадии разработки устройства. Естественно, что для крупносерийного производства масочные ПЗУ являются самым дешевым видом ПЗУ, и поэтому широко применяются в настоящее время.

Для мелких и средних серий производства радиоаппаратуры были разработаны микросхемы, которые можно программировать в специальных

устройствах - программаторах. В этих ПЗУ постоянное соединение проводников в запоминающей матрице заменяется плавкими перемычками, изготовленными из поликристаллического кремния. При производстве ПЗУ изготавливаются все перемычки, что эквивалентно записи во все ячейки памяти ПЗУ логических единиц. В процессе программирования ПЗУ на выводы питания и выходы микросхемы подаётся повышенное питание. При этом, если на выход ПЗУ подаётся напряжение питания (логическая единица), то через перемычку ток протекать не будет и перемычка останется неповрежденной. Если же на выход ПЗУ подать низкий уровень напряжения (присоединить к корпусу), то через перемычку запоминающей матрицы будет протекать ток, который испарит ее и при последующем считывании информации из этой ячейки ПЗУ будет считываться логический ноль. Такие микросхемы называются **программируемыми ПЗУ (ППЗУ) или PROM**. Запись информации в них производится однократно без возможности перезаписи информации. В качестве примера ППЗУ можно назвать микросхемы 155PE3, 556PT4, 556PT8 и другие.

Программируемые ПЗУ оказались очень удобны при мелкосерийном и среднесерийном производстве. Однако при разработке радиоэлектронных устройств часто приходится менять записываемую в ПЗУ программу. ППЗУ при этом невозможно использовать повторно, поэтому раз записанное ПЗУ при ошибочной или промежуточной программе приходится выкидывать, что естественно повышает стоимость разработки аппаратуры. Для устранения этого недостатка был разработан еще один вид ПЗУ, который мог бы стираться и программироваться заново.

ПЗУ с ультрафиолетовым стиранием строится на основе запоминающей матрицы, построенной на ячейках памяти, представляющей собой МОП транзистор, в котором затвор выполняется из поликристаллического кремния. Затем в процессе изготовления микросхемы этот затвор окисляется и в результате он будет окружен оксидом кремния - диэлектриком с прекрасными изолирующими свойствами. В ячейке при полностью стертом ПЗУ, заряда в плавающем затворе нет, и поэтому транзистор ток не проводит. При программировании ПЗУ, на второй затвор, находящийся над плавающим затвором, подаётся высокое напряжение и в плавающий затвор за счет туннельного эффекта индуцируются заряды. После снятия программирующего напряжения индуцированный заряд остаётся на плавающем затворе и, следовательно, транзистор остаётся в проводящем состоянии. Заряд на плавающем затворе подобной ячейки может храниться десятки лет.

Структурная схема описанного постоянного запоминающего устройства не отличается от описанного ранее масочного ПЗУ. Единственное отличие - вместо плавкой перемычки используется описанная выше ячейка. Такой вид

ПЗУ называется репрограммируемыми постоянными запоминающими устройствами (РПЗУ) или EPROM. В РПЗУ стирание ранее записанной информации осуществляется ультрафиолетовым излучением. Для того, чтобы этот свет мог беспрепятственно проходить к полупроводниковому кристаллу, в корпус микросхемы ПЗУ встраивается окошко из кварцевого стекла. При облучении микросхемы РПЗУ изолирующие свойства оксида кремния теряются, накопленный заряд из плавающего затвора стекает в объем полупроводника, и транзистор запоминающей ячейки переходит в закрытое состояние. Время стирания микросхемы РПЗУ 10-30 мин. Количество циклов записи - стирания микросхем EPROM находится в диапазоне от 10 до 100 раз, после чего микросхема РПЗУ выходит из строя. Это связано с разрушающим воздействием ультрафиолетового излучения на оксид кремния. В качестве примера микросхем EPROM можно назвать микросхемы 573 серии российского производства, микросхемы серий 27сXXX зарубежного производства.

Так как корпуса с кварцевым окошком очень дороги и малое количество циклов записи-стирания привели к поиску способов стирания информации из РПЗУ электрическим способом. На этом пути встретилось много трудностей, которые к настоящему времени практически решены. Сейчас достаточно широко распространены микросхемы с электрическим стиранием информации. В качестве запоминающей ячейки в них используются такие же ячейки, как и в РПЗУ, но они стираются электрическим потенциалом, поэтому количество циклов записи - стирания для этих микросхем достигает 1000000 раз. Время стирания ячейки памяти в таких ПЗУ уменьшается до 10 мс. Схема управления для электрически стираемых программируемых ПЗУ получилась сложная, поэтому наметилось два направления развития этих микросхем:

- ЭСППЗУ (EEPROM – Electrically Erasable Programmable ROM) – электрически стираемое программируемое постоянное запоминающее устройство;
- FLASH-ПЗУ.

Электрически стираемые ППЗУ (EEPROM) дороже и меньше по объему, но зато позволяют перезаписывать каждую ячейку памяти отдельно. В результате эти микросхемы обладают максимальным количеством циклов записи - стирания. Область применения электрически стираемых ПЗУ - хранение данных, которые не должны стираться при выключении питания. К таким микросхемам относятся отечественные микросхемы 573PP3, 558PP3 и зарубежные микросхемы EEPROM серии 28сXX.

В последнее время наметилась тенденция уменьшения габаритов ЭСПЗУ за счет уменьшения количества внешних выводов микросхем. Для этого адрес и данные передаются в микросхему и из микросхемы через последовательный порт. При этом используются два вида последовательных портов - SPI порт и I2C порт (микросхемы 93сXX и 24сXX серий соответственно). Зарубежной серии 24сXX соответствует отечественная серия микросхем 558РРХ.

Память типа **FLASH** по запоминающему элементу подобна памяти типа EEPROM, но имеет структурные и технологические особенности, позволяющие выделить ее в отдельный вид. Часто память такого типа используется в микроконтроллерах для хранения программ.

Флэш-память строится на транзисторных элементах памяти с "плавающим" затвором, что обеспечивает высокую плотность хранения информации. Существуют различные технологии построения базовых элементов флэш-памяти, разработанные ее основными производителями. Эти технологии отличаются количеством слоев, методами стирания и записи данных, а также структурной организацией, что отражается в их названии. Наиболее широко известны NOR и NAND типы флэш-памяти, запоминающие транзисторы в которых подключены к разрядным шинам соответственно параллельно и последовательно. Первый тип имеет относительно большие размеры ячеек и быстрый произвольный доступ, что позволяет выполнять программы непосредственно из этой памяти. Второй тип имеет меньшие размеры ячеек и быстрый последовательный доступ.

Среди главных достоинств этой памяти можно назвать следующие:

- сравнительно длительное хранение информации (десятки лет);
- сравнительно небольшие размеры;
- высокая надежность хранения данных, в том числе устойчивость к механическим нагрузкам.

Основные недостатки флэш-памяти:

- сравнительно невысокая скорость передачи данных (в сравнении с динамической оперативной памятью);
- незначительный объем (по сравнению с жесткими дисками);
- ограничение по количеству циклов перезаписи (хотя эта цифра в современных разработках очень высока – более миллиона циклов).

Необходимо отметить, что надежность и быстродействие флэш-памяти постоянно увеличиваются, что способствует более частому применению флэш-памяти в микроконтроллерах в качестве ПЗУ.

2.5. Вопросы для самопроверки

1. Основные узлы МП.
2. Отличие РОНов от рабочих регистров.
3. Назначение программного счетчика.
4. Что такое RISC архитектура?
5. Основные виды ВВ.
6. Недостатки программного ВВ.
7. Что такое DSP процессоры?
8. Какие существуют ЗУ?
9. Основные параметры ЗУ.
10. Достоинства и недостатки статических и динамических ОЗУ.
11. Виды ПЗУ.
12. Достоинства флэш-памяти и области ее применения.

3. Микроконтроллеры семейства AVR

3.1. Общие сведения о AVR микроконтроллерах

Идея разработки нового RISC-ядра принадлежит двум студентам Norwegian University of Science and Technology (NTNU) из норвежского города Тронхейма (Trondheim) Альфу Богену (Alf-Egil Bogen) и Вегарду Воллену (Vegard Wollen). В 1995 году Боген и Воллен решили предложить американской корпорации Atmel, которая была известна своими чипами с Flash-памятью, выпускать новый 8-битный RISC-микроконтроллер и снабдить его Flash-памятью программ на кристалле.

Идея была одобрена Atmel Corp. и было принято решение незамедлительно инвестировать в данную разработку. В конце 1996 года был выпущен опытный микроконтроллер AT90S1200, а во второй половине 1997-го корпорация Atmel приступила к серийному производству нового семейства микроконтроллеров, к их рекламной и технической поддержке.

Новое ядро было запатентовано и получило название AVR.

Система команд и внутреннее устройство чипов AVR разрабатывались совместно с фирмой IAR Systems - производителем компиляторов языков программирования C/C++, что обеспечило уникальные характеристики этих микроконтроллеров. В результате для AVR стало возможным получать высокую плотность кода при использовании языков высокого уровня, практически не теряя в производительности по сравнению с программами, написанными на низкоуровневом языке Ассемблера.

Микроконтроллеры AVR имеют гарвардскую архитектуру (программа и данные находятся в разных адресных пространствах) и систему команд, близкую к идеологии RISC. Процессор AVR имеет 32 8-битных регистра общего назначения, объединённых в регистровый файл.

Система команд микроконтроллеров AVR весьма развита и насчитывает в различных моделях от 90 до 133 различных инструкций. Большинство команд занимает только 1 ячейку памяти (16 бит). Большинство команд выполняется за 1 такт.

Всё множество команд микроконтроллеров AVR можно разбить на несколько групп:

- команды логических операций;
- команды арифметических операций и команды сдвига;
- команды операции с битами;
- команды пересылки данных;
- команды передачи управления;
- команды управления системой.

Управление периферийными устройствами осуществляется через адресное пространство данных. Для удобства существуют «сокращённые команды» IN/OUT.

Наибольшее распространение получили следующие семейства микроконтроллеров AVR: tiny, mega, XMEGA.

Рассмотрим преимущество микроконтроллеров семейства AVR над микроконтроллерами PIC, которые сейчас достаточно распространены.

Прежде всего, микроконтроллеры AVR имеют более совершенную архитектуру и могут выполнять команды в каждом такте (в отличие от PIC, которым для выполнения команды требуется четыре такта). Поэтому при той же тактовой частоте микроконтроллеры AVR работают в 4 раза быстрее. Кроме того, они имеют 32 рабочих регистра (в отличие от одного единственного, имеющегося в PIC) и почти в 3 раза больше команд. Благодаря этому программы для AVR практически всегда будут короче аналогичных программ для PIC. Однако, несмотря на то что в документации указывается от 90 до 120 команд (в зависимости от модели), многие из них дублируют друг друга и, по моим подсчетам, из всех команд действительно уникальными является не более 50.

А вот к так называемым регистрам специальных функций микроконтроллеров PIC (которые в AVR называются регистрами ввода/вывода) разрешен прямой доступ (можно писать непосредственно в порты), что в микроконтроллерах AVR не допускается. Однако это не такой уж большой недостаток, и в целом программы для AVR являются более эффективными. Все микроконтроллеры AVR имеют FLASH-память программ, что позволяет

осуществлять их многократное перепрограммирование. И, наконец, в связи с тем, что различные модели микроконтроллеров PIC разрабатывались на протяжении многих лет, у них имеется ряд досадных проблем с совместимостью, которых в микроконтроллерах AVR до сих пор удавалось избегать.

3.2. Архитектура AVR микроконтроллеров

3.2.1. Обобщенная структурная схема

Микроконтроллеры семейства AVR имеют единую базовую структуру. Обобщенная структурная схема микроконтроллера представлена на рис. 10.

В состав микроконтроллера входят:

- генератор тактового сигнала (GCK);
- процессор (CPU);
- постоянное запоминающее устройство для хранения программы, выполненное по технологии Flash, (FlashROM);
- оперативное запоминающее устройство статического типа для хранения данных (SRAM);
- постоянное запоминающее устройство для хранения данных, выполненное по технологии EEPROM (EEPROM);
- набор периферийных устройств для ввода и вывода данных и управляющих сигналов и выполнения других функций.

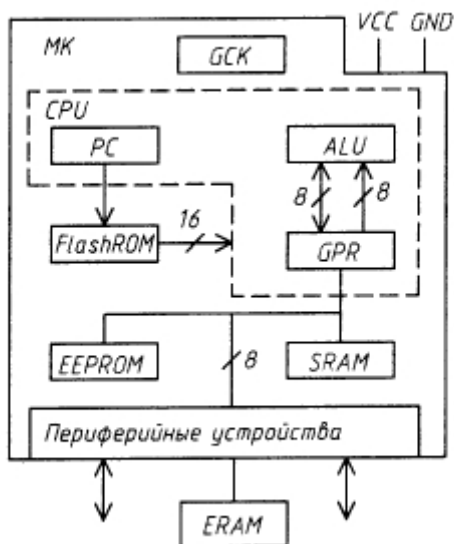


Рис. 10. Обобщенная структурная схема микроконтроллеров AVR

В микроконтроллерах типа t11, t12, t15, 1200 и t28 запоминающее устройство SRAM отсутствует. В микроконтроллерах типа t11 и t28 отсутствует также запоминающее устройство EEPROM.

К микроконтроллерам типа 8515 и m103 может быть подключено внешнее запоминающее устройство для хранения данных (ERAM). Команды программы хранятся только во внутреннем запоминающем устройстве FlashROM.

Выходы VCC и GND предназначены для подключения источника напряжения питания микроконтроллера. Уровень напряжения всех сигналов в микроконтроллере отсчитывается относительно уровня на нише GND, принимаемого за 0В. В состав процессора (CPU) входят:

- счетчик команд (PC);
- арифметико-логическое устройство (ALU);
- блок регистров общего назначения (GPR - General Purpose Registers) и другие элементы, не показанные на схеме рис. 10.

Кроме регистров общего назначения в микроконтроллере имеются регистры специальных функций, которые в семействе AVR называются регистрами ввода-вывода (I/O Registers, IOR). С участием данных регистров осуществляются:

- управление работой микроконтроллера и отдельных его устройств;
- определение состояния микроконтроллера и отдельных его устройств;
- ввод данных в микроконтроллер и отдельные его устройства и вывод данных;
- другие функции.

Для нумерации регистров ввода-вывода используются номера от 0 до 63 (от \$00 до \$3F, где S - указатель шестнадцатеричного кода). Каждому регистру присвоено имя, связанное с выполняемой этим регистром функцией. Микроконтроллеры разных типов имеют разный состав регистров ввода-вывода, при этом регистры с одинаковыми номерами могут иметь разные имена.

Работа некоторых устройств микроконтроллера зависит от состояния дополнительных однобитовых запоминающих элементов — установочных битов (Fuse Bits). Исходные значения установочных битов записываются на заводе-изготовителе. Значение установочного бита может быть изменено только при программировании микроконтроллера.

3.2.2. Генератор тактового сигнала

Микроконтроллеры семейства AVR являются устройствами синхронного типа. Действия, выполняемые в микроконтроллере, привязаны к импульсам тактового сигнала. Микроконтроллеры имеют полностью статическую структуру и могут работать при тактовой частоте от 0 Гц. В качестве генератора тактового сигнала (GCK) используются:

- внутренний генератор с внешним кварцевым или керамическим резонатором (XTAL);
- внутренний RC-генератор (IRC);
- внутренний генератор с внешней RC-цепочкой (ERC);
- внешний генератор (EXT).

У микроконтроллеров, имеющих внутренний генератор с внешним резонатором (XTAL), резонатор подключается к выводам XTAL1 и XTAL2, которые через конденсаторы малой емкости (20-30 пФ) соединяются с шиной GND. Тактовая частота определяется рабочей частотой резонатора. У микроконтроллера типа 128 при нулевом значении установочного бита INTSCAP подключение выводов XTAL1 и XTAL2 к шине GND выполняется через внутренние конденсаторы емкостью 50 пФ.

У микроконтроллеров типа t11 и t12 в качестве выводов XTAL1 и XTAL2 используются выводы PB3 и PB4.

Внешний генератор (EXT) подключается к выводу XTAL1, при этом вывод XTAL2 остается неподключенным. У микроконтроллера типа 2343, не имеющего выводов XTAL, внешний генератор подключается к выводу PB3.

В генераторах с RC-цепочкой тактовая частота определяется параметрами цепочки, но изменяется в значительных пределах при изменении напряжения питания микроконтроллера.

В микроконтроллерах типа 2343 и 1200 внутренний RC-генератор (IRC) используется при нулевом значении установочного бита RCEN.

В микроконтроллерах типа t11, t28, m163 выбор генератора для работы определяется комбинацией значений установочных битов CKSEL. У микроконтроллеров типа t11 таких битов три (CKSEL2 - 0), у микроконтроллеров остальных типов - четыре (CKSEL3 - 0).

В микроконтроллерах, имеющих внутренний генератор с внешней RC-цепочкой (ERC), резистор (3-100 кОм) подключается между выводом XTAL1 и шиной VCC, а конденсатор (не менее 20 пФ) - между выводом XTAL1 и шиной GND.

3.2.3. Процессор

Процессор (CPU) формирует адрес очередной команды, выбирает команду из памяти и организует ее выполнение. Код команды имеет формат: "слово" (16 бит) или "два слова".

В состав процессора, кроме счетчика команд (PC), арифметико-логического устройства (ALU) и блока регистров общего назначения (GPR), входят:

- регистр состояния микроконтроллера SREG;
- регистр-указатель стека SP или SPL и SPH и другие элементы, далее не рассматриваемые.

В счетчике команд адрес очередной команды формируется путем добавления 1 к числу, код которого хранится в счетчике команд. При пуске и перезапуске микроконтроллера в счетчик команд заносится код числа 0 и первая команда выбирается из FlashROM по адресу 0.

В арифметико-логическом устройстве (ALU) выполняются арифметические и логические операции. Операнды поступают из регистров общего назначения (GPR). При выполнении одноместных операций результат записывается в регистр, из которого поступил операнд. При выполнении двухместных операций результат записывается в регистр, из которого поступил первый операнд.

Блок регистров общего назначения (GPR) содержит 32 восьмиразрядных регистра, которым присвоены имена R0, R1, ..., R31. В некоторых операциях в ALU могут участвовать лишь регистры со старшими номерами (от R16 до R31). Регистры с именами от R24 до R31 могут образовывать пары, используемые для хранения слов, при этом регистр с четным номером хранит младший байт, а регистр с нечетным номером - старший байт.

Паре регистров R26, R27 присвоено имя X, паре регистров R28, R29 — имя Y, паре регистров R30, R31 - имя Z. Эти пары регистров используются для хранения адресов при обращениях к памяти с косвенной адресацией.

Регистр состояния микроконтроллера SREG (№ \$3F) содержит восемь разрядов (SREG7, SREG6, ..., SREG0).

Разряд SREG, 7 (I) используется для разрешения/запрещения прерываний. При I = 0 все прерывания запрещены. При I = 1 любое прерывание может быть разрешено.

Разряд SREG6 (T) используется для хранения бита при выполнении операции с битами. Остальные разряды регистра SREG используются для хранения признаков результатов арифметических и логических операций, выполняемых в ALU.

Регистр-указатель стека SP(№ \$3D) хранит и формирует адрес при обращении к стеку типа LIFO. В микроконтроллерах типа t11, t12, t15, 1200 и t28 в качестве стека используется специальное запоминающее устройство (аппаратный стек). Этот стек используется только для хранения адресов возврата при прерываниях и обращениях к подпрограммам. В системе команд отсутствуют команды обращения к стеку.

В микроконтроллерах AVR других типов в качестве стека используется выделяемая пользователем область в SRAM. В системе команд есть команды для обращения к стеку. Запись байтов в стек выполняется в порядке уменьшения адресов в SRAM. При пуске и перезапуске микроконтроллера в регистр-указатель стека заносится код числа 0. Для нормальной работы стека в регистр-указатель необходимо занести другой начальный адрес. В микроконтроллерах с большой емкостью SRAM регистр-указатель состоит из двух регистров SPL и SPH (№№ \$3D и \$3E).

3.2.4. Запоминающее устройство FlashROM

Постоянное запоминающее устройство FlashROM предназначено для хранения кодов команд программы и констант. Ячейка памяти содержит 16 разрядов. В ней могут храниться код команды формата "слово", половина кода команды формата "два слова" или коды двух констант. При чтении кодов команд адрес в FlashROM поступает из счетчика команд. При чтении констант адрес поступает из пары Z регистров общего назначения.

Запись кодов в FlashROM выполняется в процессе программирования побайтно. В микроконтроллерах с большим числом выводов (20 и более) байт может вводиться параллельно или последовательно. В микроконтроллерах с малым числом выводов (8) байт вводится последовательно.

Программирование может выполняться с использованием дополнительного источника напряжения +12 В или без использования дополнительного источника. Последовательное программирование без использования дополнительного источника напряжения производится с использованием трех выводов микроконтроллера и может выполняться после установки микроконтроллера в аппаратуре, где ему предстоит работать.

В микроконтроллере типа m163 имеется возможность вводить и записывать в FlashROM коды в процессе работы микроконтроллера. Эта процедура (Self-programming) выполняется под управлением специальной программы (Boot Loader), которая составляется программистом и записывается в специальную секцию FlashROM. При выполнении этой программы используется регистр SPMCR (№ \$37).

3.2.5. Запоминающее устройство SRAM

Оперативное запоминающее устройство статического типа SRAM предназначено для хранения данных, получаемых в процессе работы микроконтроллера. При выключении напряжения питания микроконтроллера данные в SRAM теряются.

Ячейка памяти содержит 8 разрядов. Адрес байта при обращении к SRAM может быть указан в коде команды с обращением к SRAM (прямая адресация) или предварительно записан в пару регистров X, Y или Z (косвенная адресация). Обращение к SRAM может выполняться с использованием адреса, хранящегося в регистре-указателе стека.

Байт для записи в SRAM поступает из регистра общего назначения. Байт, считанный из SRAM, поступает в регистр общего назначения.

В адресное пространство SRAM, кроме адресов, по которым выполняется обращение к ячейкам памяти SRAM, включены 32 адреса для обращения к регистрам общего назначения (адреса от \$00 до \$1F) и 64 адреса для обращения к регистрам ввода-вывода (адреса от \$20 до \$5F).

Первой ячейке в SRAM соответствует адрес \$60. Адрес для обращения к регистру общего назначения по команде обращения к SRAM совпадает с номером регистра. Адрес для обращения к регистру ввода-вывода по команде обращения к SRAM равен его номеру, увеличенному на число 32 (\$20).

В микроконтроллерах, допускающих подключение внешнего запоминающего устройства ERAM, адреса для обращения к ячейкам ERAM включены в адресное пространство SRAM и расположены после старшего адреса SRAM.

На рис. 11 в качестве примера показано распределение адресов в адресном пространстве SRAM между регистрами общего назначения (GPR), регистрами ввода-вывода (IOR), ячейками памяти в SRAM и ячейками памяти в ERAM у микроконтроллера типа 8515.

Запись в стек выполняется в порядке убывания адресов. В качестве исходного адреса для обращений к стеку, который заносится в регистр-указатель стека в начале программы, целесообразно использовать старший адрес SRAM.

3.2.6. Запоминающее устройство EEPROM

Постоянное запоминающее устройство EEPROM предназначено для хранения данных, записанных при программировании микроконтроллера и получаемых в процессе выполнения программы. При выключении напряжения питания данные сохраняются. Ячейка памяти содержит 8 разрядов.

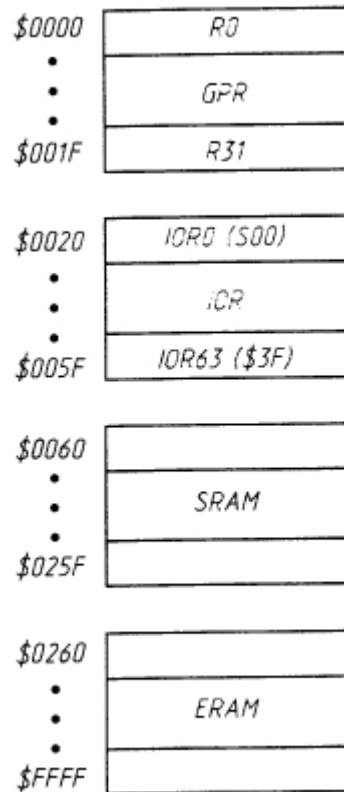


Рис. 11. Распределение адресного пространства

EEPROM имеет обособленное адресное пространство. При обращении к EEPROM адрес записывается в регистр адреса EEAR (№ \$1E). В микроконтроллерах типа 8515, 8535, m163 и m103 регистр адреса содержит два восьмиразрядных регистра - EEARL и EEARN (№№ \$1K и \$1F). Байт, предназначенный для записи, заносится в регистр данных EEDR (№ \$1D). Байт, получаемый при чтении, поступает в этот же регистр. Для управления процедурами записи и чтения используется регистр управления EECR (№ \$1C).

Для записи байта в EEPROM необходимо:

- записать адрес в регистр адреса;
- записать байт в регистр данных;
- установить в единичное состояние разряд EEMWE регистра EECR;
- при EEMWE = 1 установить в единичное состояние разряд EEWЕ регистра EECR.

Процедура записи выполняется в зависимости от величины напряжения питания за 2,5—4 мс. При завершении записи разряд EEWЕ регистра EECR аппаратно сбрасывается в нулевое состояние.

Разряд EEMWE сохраняет единичное состояние в течение 4-х тактов после установки и аппаратно сбрасывается в нулевое состояние.

В микроконтроллерах типа t12, t15, t133, 8535, m163 и m103 при нулевом состоянии разряда EEWЕ формируется запрос прерывания EЕ RDY. Прерывание по данному запросу разрешено при единичном состоянии разряда EERIE регистра EECR.

Для чтения байта из EEPROM необходимо:

- записать адрес в регистр адреса;

- установить в единичное состояние разряд EERE регистра EECR.

Считанный байт поступает в регистр данных. Разряд EERE регистра EECR аппаратно сбрасывается в нулевое состояние.

3.3. Микроконтроллер AVR ATmega128

В качестве примера микроконтроллера семейства AVR рассмотрим микроконтроллер ATmega128 (рис. 12).

ATmega128 – маломощный 8-разр. КМОП микроконтроллер, основанный на расширенной AVR RISC-архитектуре. За счет выполнения большинства инструкций за один машинный цикл ATmega128 достигает высокой производительности по сравнению с микроконтроллерами других семейств, что позволяет проектировщикам систем оптимизировать соотношение энергопотребления и быстродействия.

Ядро AVR сочетает богатый набор инструкций с 32 универсальными рабочими регистрами. Все 32 регистра непосредственно подключены к арифметико-логическому устройству (АЛУ), который позволяет указать два различных регистра в одной инструкции и выполнить ее за один цикл. Данная архитектура обладает большей эффективностью кода за счет достижения производительности в 10 раз выше по сравнению с обычными CISC-микроконтроллерами.

Микроконтроллер производится по технологии высокоплотной энергонезависимой памяти компании Atmel. Встроенная внутрисистемная программируемая флэш-память позволяет перепрограммировать память программ непосредственно внутри системы через последовательный интерфейс SPI с помощью простого программатора или с помощью автономной программы в загрузочном секторе. Загрузочная программа может использовать любой интерфейс для загрузки прикладной программы во флэш-память. Программа в загрузочном секторе продолжает работу в процессе обновления прикладной секции флэш-памяти, тем самым поддерживая двухоперационность: чтение во время записи. За счет сочетания 8-разр. RISC

Структура микроконтроллера ATmega128 включает следующие функциональные блоки:

- 8-разрядное арифметическо-логическое устройство (АЛУ);
- внутреннюю flash-память программ объемом 128 Кбайт с возможностью внутрисистемного программирования через последовательный интерфейс;
- 32 регистра общего назначения;
- внутреннюю EEPROM память данных объемом 4 Кбайт;
- внутреннее ОЗУ данных объемом 4 Кбайт;
- 6 параллельных 8-разрядных портов;
- 4 программируемых таймера-счетчика;
- 10-разрядный 8-канальный АЦП и аналоговый компаратор;
- последовательные интерфейсы UART0, UART0, TWI и SPI;
- блоки прерывания и управления (включая сторожевой таймер).

Рассмотрим назначение различных портов и управляющих сигналов микроконтроллера.

Port A (PA7..PA). 8-разрядный двунаправленный порт. К выводам порта могут быть подключены встроенные нагрузочные резисторы (отдельно к каждому разряду). Выходные буферы обеспечивают ток 20 мА и способность прямо управлять светодиодным индикатором. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, ток будет вытекать только при подключенных встроенных нагрузочных резисторах. Порт А при наличии внешней памяти данных используется для организации мультиплексируемой шины адреса/данных.

Port B (PB7..PB0). 8-разрядный двунаправленный порт со встроенными нагрузочными резисторами. Выходные буферы обеспечивают ток 20 мА. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, ток будет вытекать только при подключенных встроенных нагрузочных резисторах. Порт В используется также при реализации специальных функций.

Port C (PC7..PC0). Порт С является 8-разрядным выходным портом. Выходные буферы обеспечивают ток 20 мА. Порт С при наличии внешней памяти данных используется для организации шины адреса.

Port D (PD7..PD0). 8-разрядный двунаправленный порт со встроенными нагрузочными резисторами. Выходные буферы обеспечивают ток 20 мА. При использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, ток будет вытекать только при подключенных встроенных нагрузочных резисторах. Порт D используется также при реализации специальных функций.

Port E (PE7..PE0). 8-разрядный двунаправленный порт со встроенными нагрузочными резисторами. Выходные буферы обеспечивают ток 20 мА. При

использовании выводов порта в качестве входов и установке внешнего сигнала в низкое состояние, вытекающий через них ток обеспечивается только при подключенных встроенных нагрузочных резисторах. Порт E используется также при реализации специальных функций.

Port F (PF7..PF0). 8-разрядный входной порт. Входы порта используются также как аналоговые входы аналого-цифрового преобразователя.

RESET. Вход сброса. Для выполнения сброса необходимо удерживать низкий уровень на входе более 50 нс.

XTAL1, XTAL2. Вход и выход инвертирующего усилителя генератора тактовой частоты.

TOSC1, TOSC2. Вход и выход инвертирующего усилителя генератора таймера/счетчика.

WR, RD. Стробы записи и чтения внешней памяти данных.

ALE. Строб разрешения фиксации адреса внешней памяти. Строб ALE используется для фиксации младшего байта адреса с выводов AD0-AD7 в защелке адреса в течение первого цикла обращения. В течение второго цикла обращения выходы AD0-AD7 используются для передачи данных.

AVCC. Напряжение питания аналого-цифрового преобразователя. Вывод подсоединяется к VCC через низкочастотный фильтр.

AREF. Вход опорного напряжения для аналого-цифрового преобразователя. На этот вывод подается напряжение в диапазоне между AGND и AVCC.

AGND. Это вывод должен быть подсоединен к отдельной аналоговой земле, если она есть на плате. В ином случае вывод подсоединяется к общей земле.

PEN. Вывод разрешения программирования через последовательный интерфейс. При удержании сигнала на этом выводе на низком уровне после включения питания, прибор переходит в режим программирования по последовательному каналу.

VCC, GND. Напряжение питания и земля.

Высокие характеристики семейства AVR обеспечиваются следующими особенностями архитектуры:

- в качестве памяти программ используется внутренняя flash-память. Она организована в виде матрицы 16-разрядных ячеек и может загружаться программатором, либо через порт SPI;

- система команд включает 133 инструкций;

- 16-разрядные память программ и шина команд вместе с одноуровневым конвейером позволяют выполнить большинство инструкций за один такт синхрогенератора (50 нс при частоте FOSC=20 МГц);

– память данных имеет 8-разрядную организацию. Младшие 32 адреса пространства занимают регистры общего назначения, далее следуют 64 адреса регистров ввода-вывода, затем внутреннее ОЗУ данных объемом до 4096 ячеек. Возможно применение внешнего ОЗУ данных объемом до 60 Кбайт;

– внутренняя энергонезависимая память типа EEPROM объемом до 4 Кбайт представляет собой самостоятельную матрицу, обращение к которой осуществляется через специальные регистры ввода-вывода.

Тридцать два регистра общего назначения включены в сквозное адресное пространство ОЗУ данных и занимают младшие адреса. Хотя физически регистры выделены из памяти данных, такая организация обеспечивает гибкость в работе. Файл регистров общего назначения прямо связан с АЛУ, каждый из регистров способен работать как аккумулятор. Большинство команд выполняются за один такт, при этом из регистров файла могут быть выбраны два операнда, выполнена операция и результат возвращен в регистровый файл. Старшие шесть регистров файла могут использоваться как три 16-разрядных регистра, и выполнять роль, например, указателей при косвенной адресации.

Пространство памяти ввода-вывода содержит 64 адреса с непосредственной адресацией или может адресоваться как память данных, следующая за регистрами по адресам - F. Кроме того, ATmega128 имеет пространство расширенного ввода-вывода по адресам - \$FF в статическом ОЗУ, для доступа к которому могут использоваться только процедуры ST/STS/STD и LD/LDS/LDD.

Высокопроизводительное АЛУ AVR-микроконтроллеров работает в непосредственной связи со всеми 32 универсальными рабочими регистрами. АЛУ позволяет выполнить за один машинный цикл операцию между двумя регистрами или между регистром и константой. Операции АЛУ могут быть классифицированы на три группы: арифметические, логические и битовые. Кроме того, архитектурой ATmega128 поддерживаются операции умножения со знаком и без знака и дробным форматом. После выполнения арифметической операции регистр статуса обновляется для отображения результата выполнения операции.

Для ветвления программы поддерживаются инструкции условных и безусловных переходов и вызовов процедур, позволяющих непосредственно адресоваться в пределах адресного пространства. Большинство инструкций представляют собой одно 16-разр. слово. Каждый адрес памяти программ содержит 16- или 32-разр. инструкцию. Флэш-память программ разделена на две секции: секция программы начальной загрузки и секция прикладной программы. Обе секции имеют отдельные биты защиты от записи и чтения/записи. Инструкция SPM (запись в секцию прикладной программы) должна использоваться только внутри секции программы начальной загрузки.

При генерации прерывания и вызове подпрограмм адрес возврата из программного счетчика записывается в стек. Стек эффективно распределен в статическом ОЗУ памяти данных и, следовательно, размер стека ограничен общим размером статического ОЗУ и используемым его объемом. В любой программе сразу после сброса должна быть выполнена инициализация указателя стека (SP) (т.е. перед выполнением процедур обработки прерываний или вызовом подпрограмм). Указатель стека – SP – доступен на чтение и запись в пространстве ввода-вывода. Доступ к статическому ОЗУ данных может быть легко осуществлен через 5 различных режимов адресации архитектуры AVR. Стек обычно используется для хранения временных данных, для хранения локальных переменных и для хранения адресов возврата при прерываниях и вызовах подпрограмм. Регистр указателя стека указывает на вершину стека. Обратите внимание на организацию стека, который направляется от старших в более младшие позиции статического ОЗУ. Это означает, что команда помещения в стек PUSH уменьшает значение указателя стека.

Указатель стека указывает на область стека в статическом ОЗУ данных, где размещены стеки прерываний и подпрограммы. Данная область стека в статическом ОЗУ памяти данных должна быть определена программно до вызова любой процедуры или разрешения прерываний. Указатель стека однократно декрементируется при помещении данных в стек инструкцией PUSH и дважды декрементируется при помещении в стек адреса возврата при вызове подпрограмм или прерываниях. Указатель стека однократно инкрементируется при извлечении данных из стека инструкцией POP и дважды инкрементируется при извлечении адреса возврата при выполнении инструкции выхода из подпрограммы RET или выхода из процедуры обработки прерываний RETI.

Гибкий модуль прерываний содержит свои управляющие регистры в пространстве ввода-вывода и имеет дополнительный бит общего разрешения работы системы прерываний в регистре статуса. У всех прерываний имеется свой вектор прерывания в соответствии с таблицей векторов прерываний. Прерывания имеют приоритет в соответствии с позицией их вектора. Прерывания с меньшим адресом прерывания имеют более высокий приоритет.

AVR-микроконтроллеры поддерживают несколько различных источников прерываний. Все прерывания, а также сброс имеют свой индивидуальный вектор в памяти программ. Для каждого прерывания имеется собственный бит разрешения. Кроме того, имеется возможность общего разрешения работы прерываний с помощью управления соответствующим битом в статусном регистре. В зависимости от значения программного счетчика прерывания могут быть автоматически отключены, если запрограммировать биты защиты

загрузочного сектора VLB02 или VLB12. Данная функция улучшает защиту программы.

Имеется два основных типа прерываний.

Первый тип прерываний активизируется событием, которое приводит к установке флага прерываний. Для данных прерываний программный счетчик изменяется на соответствующий вектор прерывания для выполнения процедуры его обработки и затем аппаратно очищает флаг прерывания. Флаги прерывания также сбрасываются путем записи лог.1 в соответствующий разряд. Если возникает условие прерывания, но данное прерывание запрещено, то флаг устанавливается и запоминается до разрешения этого прерывания или сбрасывается программно. Аналогично, если возникает одно и более условий прерываний при сброшенном флаге общего разрешения прерываний, то соответствующий флаг устанавливается и запоминается до возобновления работы прерываний, а затем прерывания будут выполнены в соответствии с приоритетом.

Второй тип прерываний активизируется сразу после выполнения условия прерывания. Данные прерывания не обязательно имеют флаги прерываний. Если условие прерывания исчезает до его разрешения, то данный запрос игнорируется.

После выхода из прерывания AVR-микроконтроллер возвращается к выполнению основной программы и выполняет еще одну инструкцию до обслуживания любого из отложенных прерываний.

Обратите внимание, что регистр статуса автоматически не запоминается при вызове процедуры обработки прерывания и не восстанавливается при выходе из этой процедуры. Данные действия необходимо выполнить программно.

3.4. Вопросы для самопроверки

1. Достоинства микропроцессоров семейства AVR.
2. Назначение регистров X, Y, Z.
3. Из каких двух регистров состоит указатель стека SP?
4. Организация адресного пространства микроконтроллера.
5. Назначение FlashROM.

СПИСОК ЛИТЕРАТУРЫ

1. Балашов Е. П., Григорьев В. Л., Петров Г. А. Микро- и миниЭВМ. Л.: – Энергоатомиздат. 1984. – 376 с.
2. Микропроцессоры: В 3-х кн. / под ред. Преснухина. М.: Высшая школа. 1986.
3. Токхайм Р. Микропроцессоры: Курс и упражнения / Пер. с англ. под ред. Грасевича. – М.: Энергоатомиздат. 1987. – 338 с.
4. Майоров С. А., Кириллов В. В., Приблудла А. А. Введение в микроЭВМ. – Л.: Машиностроение. 1988. – 303 с.
5. Горбунов А.Л. Микропроцессорные устройства РЭС: Конспект лекций. – М.:МГТУ ГА, 1999. – 24 с.
6. Смирнов В.Е. Микропроцессорные системы. Принципы построения микропроцессорных систем. – М.:МГТУ ГА, 2004. ч.1. – 93 с.
7. Гребнев В.В. Микроконтроллеры семейства AVR фирмы Atmel – М.: ИП РадиоСофт. 2002. – 186 с.
8. Евстифеев А.В. Микроконтроллеры AVR семейства Tini и Mega фирмы «ATMEL» – М.: Издательский дом «Додэка-XXI», 2004. – 560 с.

Введение	3
1. Общие сведения о микропроцессорах	4
1.1. История развития микропроцессоров	4
1.2. Характеристики микропроцессоров	9
1.3. Классификация микропроцессоров	10
1.4. Этапы разработки микропроцессоров	15
1.5. Общие сведения о программировании микропроцессоров	20
1.6. Вопросы для самопроверки	24
2. Архитектура микропроцессоров	24
2.1. Обобщенная структура микропроцессора	24
2.2. Логическая структура микропроцессора	31
2.3. Организация ввода-вывода	33
2.3.1. Программный ввод-вывод	34
2.3.2. Ввод-вывод по прерываниям	34
2.3.3. Прямой доступ к памяти	38
2.4. Запоминающие устройства микропроцессоров	41
2.4.1. Виды и параметры запоминающих устройств	41
2.4.2. Оперативные запоминающие устройства	42
2.4.3. Постоянные запоминающие устройства	43
2.5. Вопросы для самопроверки	47
3. Микроконтроллеры семейства AVR	47
3.1. Общие сведения о AVR микроконтроллерах	47
3.2. Архитектура AVR микроконтроллеров	49
3.2.1. Обобщенная структурная схема	49
3.2.2. Генератор тактового сигнала	51
3.2.3. Процессор	52
3.2.4. Запоминающее устройство FlashROM	53
3.2.5. Запоминающее устройство SRAM	54
3.2.6. Запоминающее устройство EEPROM	54
3.3. Микроконтроллер AVR ATmega128	56
3.4. Вопросы для самопроверки	62
Список литературы	63