**1 Краткие исторические сведения по развитию и применению электронных вычислительных машин, микропроцессорных устройств и систем.**

Микропроцессоры – это логические устройства, оказывающие заметное влияние на нашу жизнь. Микропроцессоры можно обнаружить в карманных калькуляторах, в кассовых аппаратах магазинов, в бытовых и научных приборах, в медицинском и офисном оборудовании, в мобильных телефонах,

в детских игрушках – и это далеко не полный перечень их применения. Более того, каждый день находятся новые приложения и разрабатываются новые изделия на основе микропроцессоров. Можно констатировать, что в настоящее время практически ни одна разработка радиоэлектронного

оборудования не проводится без применения микропроцессоров. Их потенциальное воздействие на нашу жизнь почти не поддается воображению.

**С 1951 года, когда был создан первый производимый промышленностью компьютер** (Univac I, США), электронные вычислительные машины (ЭВМ) оказывают сильное влияние на наше общество и образ жизни. Возникла новая отрасль промышленности. Такие термины, как «цифровые вычисления», «логическое проектирование» и «программирование», стали научными и инженерными понятиями. Однако широта этих понятий часто приводила к расхождению интересов. Например, интересы одной группы специалистов лежали в использовании ЭВМ и программировании (область программного обеспечения), другой – в создании ЭВМ (область аппаратуры). Хотя такое деление по интересам было, по-видимому, оправданным в отношении больших вычислительных машин, но с момента появления ***миникомпьютеров в 1965*** году проблемы, с которыми сталкивались прикладные программисты и конструкторы машин, стали переплетаться теснее. **Миникомпьютеры** уже не были всецело предназначены для обработки данных и решения вычислительных задач; зачастую они входили как составные части в системы, требовавшие быстрого принятия решения,— ***системы реального времени****.*

**С появлением в 1971 году микропроцессоров** расхождение интересов еще более уменьшилось. Началась эра программируемой логики. В эту эру понятия программирования и принципы проектирования логических схем сблизились настолько, что их взаимопроникновение потребовало и от ученых, и от инженеров полного понимания как принципов программирования, так и аппаратуры. Только при этом условии можно было успешно использовать все заложенные в микропроцессорах возможности.

*Компьютер –* термин, используемый в англоязычной среде, для наименования электронной вычислительной машины (ЭВМ). *ЭВМ* – универсальное устройство для автоматической обработки

информации, представленной в форме цифровых кодов. *Процессор –* основной функциональный блок компьютера, непосредственно осуществляющий процесс обработки цифровых данных и

программное управление этим процессом. *Микропроцессор* – процессор, выполненный по технологии больших интегральных схем (БИС-технологии) на одном кристалле в одном корпусе.

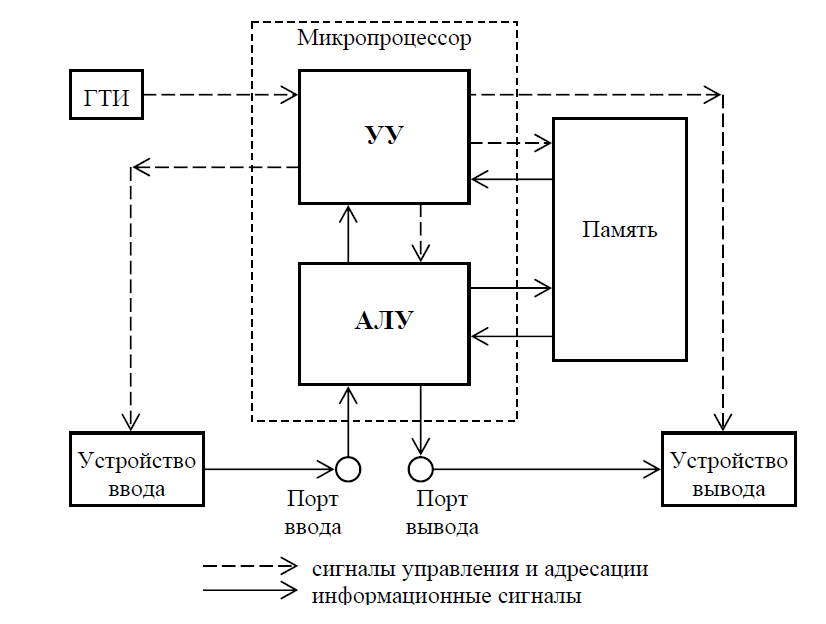
*Микрокомпьютер* – компьютер, построенный на базе микропроцессора. *Микроконтроллер* – микрокомпьютер, предназначенный для работы в качестве встроенного блока управления аппарата или прибора под управлением не изменяемой в процессе работы программы. *Однокристальный микроконтроллер* – микроконтроллер, выполненный на одном кристалле в одном корпусе.

Микропроцессор является продуктом достижений технологии полупроводников, позволяющей создавать большие интегральные схемы (БИС) с числом транзисторов на одной кремниевой подложке (на одном «кристалле») в десятки и сотни тысяч. Это называют *высокой степенью интеграции.* Собственно говоря, высокая степень интеграции и привела к созданию микропроцессоров. Как будет показано далее, в конструкцию микропроцессора заложена большая гибкость. Но сам по себе он не может решить ту или иную конкретную задачу. Чтобы решить задачу, его нужно соединить с другими устройствами и запрограммировать. В их число обычно входят память и устройства ввода/вывода. *Некоторая совокупность соединенных друг с другом системных устройств, включающая микропроцессор, память и устройства ввода/вывода, нацеленная на выполнение некоторой четко определенной функции, в зависимости от ее сложности и называется* ***микропроцессорным устройством,*** или ***микропроцессорной системой****.* Микрокомпьютеры обладают всеми свойствами обычных ЭВМ, однако замечательная их особенность состоит в относительно низкой стоимости и малых размерах. Именно этому они обязаны своей популярностью и успехом.

Большие ЭВМ и миникомпьютеры обладают, конечно, большей вычислительной мощностью, но не для всех приложений эта мощность оказывается необходимой. Более того, стоимость больших компьютеров и миникомпьютеров часто не позволяет включать их в системы, где они могли бы с успехом применяться. Микропроцессоры открывают возможность для применения программируемых устройств в тех логических системах, для которых фактор стоимости оказывается важнее, чем скорость и разнообразие вычислений.

**2 Типовая структура микропроцессорного устройства (микрокомпьютера),назначение функциональных блоков, общие сведения о его функционировании**

Типовая компьютерная система включает шесть функциональных блоков: устройство ввода, память, арифметико-логическое устройство (АЛУ), устройство управления (УУ), устройство вывода, генератор тактовых импульсов (ГТИ).

****

**Устройство ввода** получает из внешнего мира данные и команды, которые поступают в память.

**Устройство вывода** получает вычисленные результаты и передает их человеку- оператору или другой системе.

Запоминание больших объемов информации происходит в **памяти**, или, точнее, в запоминающем устройстве (ЗУ). Этот функциональный блок компьютера подразделяется на подблоки, называемые *регистрами,* каждый из которых способен хранить одно машинное слово. Машины, спроектированные в соответствии с концепцией разделения памяти на два вида (программа находится всегда в одной памяти, а данные – в другой, что позволяет одновременно обращаться и к памяти программы и данных), называют машинами *гарвардского* типа. В соответствии со второй концепцией различие между программной памятью и памятью для данных не проводится - это машины *фон-Неймановского,* или *принстонского* типа.

Обработка данных осуществляется главным образом в **арифметико-логическом устройстве**. Эта обработка включает как арифметические, так и логические операции.

**Устройство управления** управляет работой компьютера. Оно автоматически, последовательно по одной, выбирает команды из памяти, декодирует каждую из них и генерирует необходимые для ее выполнения сигналы.

Синхронизация работы отдельных блоков компьютера осуществляется с помощью ***генератора тактовых импульсов* (ГТИ).**

**3 Память микрокомпьютера. Компьютеры гарвардского и фон-Нейманского**

**типов.**

Запоминание больших объемов информации происходит в памяти, или, точнее, в запоминающем устройстве (ЗУ). Этот функциональный блок компьютера подразделяется на подблоки, называемые ***регистрами****,* каждый из которых способен хранить одно машинное слово. Каждый такой регистр, или ячейка памяти, имеет свой адрес. ***Адрес*** – это просто целое число, однозначно идентифицирующее ячейку. Слово, хранящееся в ячейке, называют ***содержимым*** этой ячейки.

Итак, как данные, так и программа хранятся в памяти. Это важное обстоятельство приводит к двум основным концепциям проектирования компьютеров. Первая заключается в том, что компьютер имеет два отдельных и четко различающихся вида памяти. Программа находится всегда в одной

памяти, а данные – в другой, что позволяет одновременно обращаться и к памяти программы и данных. Эта возможность значительно повышает быстродействие системы. Машины, спроектированные в соответствии с концепцией разделения памяти на два вида, называют машинами ***гарвардского*** типа. В соответствии со второй концепцией различие между программной

памятью и памятью для данных не проводится. Такие компьютеры называют машинами ***фон-Неймановского,*** или ***принстонского*** типа. В них программа может размещаться в любом месте общей памяти, и задача программиста — следить за тем, чтобы данные и программа обрабатывались по-разному. Преимущество второй концепции – в возможности трактовать программу как данные, что позволяет компьютеру изменять свои собственные программы. Существуют микропроцессоры, спроектированные в соответствии, как с первой, так и со второй концепцией.

Благодаря низкой стоимости микропроцессоры часто предназначаются для решения одной конкретной задачи. Большие универсальные ЭВМ постоянно перепрограммируются и поэтому могут решать задачи широкого спектра. Микропроцессорам, специализированным для одного конкретного приложения, такая гибкость не нужна. Однажды написанная и отлаженная программа в дальнейшем обычно не изменяется. Поэтому такие микрокомпьютеры часто имеют два вида памяти: память, из которой возможно только считывание (ROM – read only memory), или постоянная память, и память со считыванием и записью (RWM – read/write memory). Изменить информацию,

однажды записанную в постоянную память, сложно. Память этого типа благодаря своей низкой стоимости используется для хранения программ и постоянных данных; изменяющаяся информация хранится в памяти со считыванием и записью.

Память со считыванием и записью принято называть памятью с произвольной выборкой (RAM – random-access memory), несмотря на то что и память только со считыванием также обладает произвольностью выборки. Термин «произвольная выборка», или «произвольный доступ», соответствует тому факту, что обращение к любой ячейке выполняется за одно и то же время.

В качестве русских эквивалентов сокращениям RAM, RWM и ROM часто используются: ЗУПВ – запоминающее устройство с произвольной выборкой, ОЗУ – оперативное запоминающее устройство и ПЗУ – постоянное запоминающее устройство.

Следует обратить внимание на то, что из наличия постоянной памяти не следует, что машина относится к гарвардскому типу. В машине фон- Неймановского типа ячейки ROM (только для считывания) могут появляться в памяти где угодно. Главное то, что микропроцессору «безразлично», откуда он получает информацию – из ROM или RAM.

**4 Арифметико-логическое устройство.**

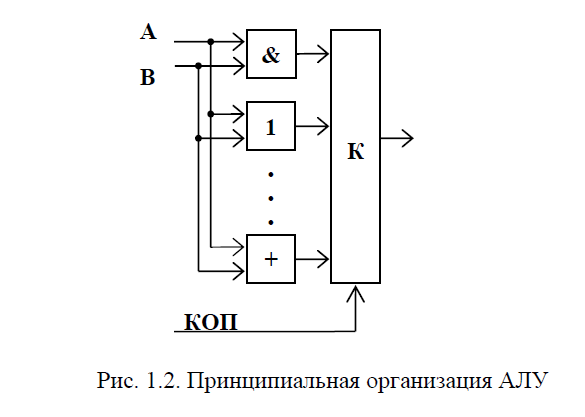
Обработка данных осуществляется главным образом в арифметико-логическом устройстве. Эта обработка включает как арифметические, так и логические операции. Встроенные операции чрезвычайно элементарны. Полный перечень возможных встроенных операций для любого процессора задается его системой команд. Более сложные математические действия должны выполняться с помощью программ, пользующихся встроенными операциями.

В качестве модели для пояснения принципа работы АЛУ рассмотрим схему, показанную на рис. 1.2.

На вход АЛУ поступают данные **А** и **В,** над которыми необходимо произвести заданное действие. Над этими данными одновременно производятся все возможные (перечисленные в системе команд) логические и арифметические операции, результаты которых поступают на вход коммутатора (К). На коммутатор также подается код операции (КОП) команды, выполняющейся в данный момент, который указывает, какой из результатов должен быть передан на выход АЛУ. Обычно в состав АЛУ входит ряд регистров, предназначенных для хранения операндов на время их обработки.

Главный регистр в АЛУ называется ***аккумулятором****.* В нем, как правило, находится один из операндов перед выполнением операции, и в него же помещается ее результат.

В состав АЛУ входит также регистр флагов. ***Флаги*** или ***флажки*** *–* это просто биты, содержащие информацию, характеризующую состояние микропроцессора, которое важно для выбора дальнейшего пути вычислений. Например, может существовать флажок, указывающий на нулевой результат операции. Программист может воспользоваться проверкой этого флажка для принятия решения: если некоторая операция дала нулевой результат, то будет выполнена одна последовательность команд, в противном случае — другая. Наиболее часто используются три флага: **С** – переполнения, **Z** – нулевого результата, **DC** – десятичного переноса.



Флажковые биты, характеризующие результаты операций или каких-либо проверок, часто размещаются вместе с другой важной информацией о состоянии машины в специальном регистре, называемом ***словом состояния программы*** (PSW – program status word).

**5 Устройство управления.**

Устройство управления управляет работой компьютера. Оно автоматически, последовательно по одной, выбирает команды из памяти, декодирует каждую из них и генерирует необходимые для ее выполнения сигналы. Для того чтобы получить команду из памяти, устройство управления прежде всего должно знать ее адрес. Обычно команды выбираются из последовательных ячеек памяти, и их адреса указываются ***программным счетчиком*** (ПС)*,* находящимся в устройстве управления.

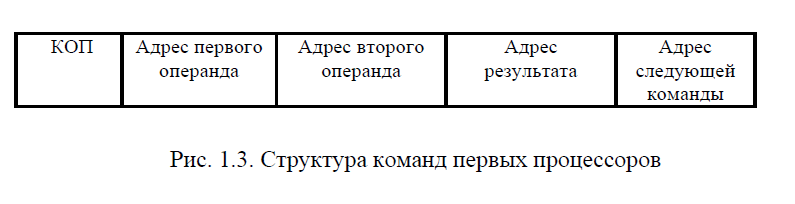
В момент включения питания автоматически формируется сигнал СБРОС, которым обнуляется программный счетчик и это обеспечивает выборку команды из нулевой ячейки памяти. После извлечения первой команды к счетчику добавляется единица, и он указывает уже на следующую

ячейку с командой и т.д. При необходимости можно счетчику в нужный момент по отдельному каналу навязать адрес, который будет указывать не на следующую ячейку, а какую-то иную. Благодаря этому можно организовать ветвление в программе.

Далее, чтобы иметь возможность декодировать и выполнить текущую команду, ее нужно где-то запомнить. Этой цели в устройстве управления служит ***регистр команды****.* Для того чтобы быть правильно проинтерпретированной устройством управления, команда должна иметь определенную ***структуру***, т.е. содержать определенные логические части, называемые полями, каждая из которых несет определенную информацию. Взаимное расположение полей в команде с указанием длин полей в битах называют ***форматом команды****.* У микропроцессоров разных типов форматы команд различны. Однако есть информация, которая должна присутствовать в команде в любом случае. Наиболее важное значение имеет ***код операции*** (КОП) и в некоторых командах адрес. ***Код операции*** – это совокупность двоичных цифр, которые однозначно определяют операцию, выполняемую в процессе интерпретации команды. ***Адресная часть*** команды (если она присутствует) указывает на ячейки (например в памяти), к которым нужно обратиться, выполняя команду. Например, если выполняется операция сложения, адресная часть команды может указывать на ячейку, где находится второе слагаемое. Но адресный код не обязателен – в некоторых командах он может и отсутствовать. Структура команд первых процессоров была более громоздкой (рис. 1.3). В команде указывались адреса обоих операндов, адрес, куда записывался

результат и адрес следующей команды. Для извлечения такой команды приходилось несколько раз обращаться к памяти, что естественно приводило к значительному снижению быстродействия. За счет введения программного счетчика избавились от части, в которой указывался адрес следующей

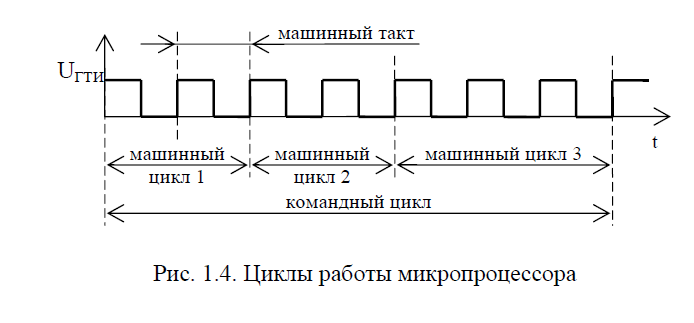
команды. Так как один из операндов, как правило, после выполнения некоторого действия не нужен, то стали записывать результат на его место, что позволило исключить еще одно поле. С введением аккумулятора в АЛУ приняли, что один из операндов находится в нем, и отпала необходимость указывать адрес этого операнда. Таким образом, в процессе эволюции компьютеров сократили длины команд и за счет этого увеличили их быстродействие.

****

Важно различать понятия «адрес ячейки памяти» и «содержимое ячейки памяти». В командах определенного формата существует адресная часть. Это числовой указатель ячейки, связанной с операндом. Однако команда сама находится в памяти и, следовательно, имеет связанный с ней адрес. Как правило, этот адрес не совпадает с адресной частью в самой команде.

Следующая функция устройства управления – это синхронизация работы отдельных блоков компьютера. Она осуществляется с помощью ***генератора тактовых импульсов*** (ГТИ)*,* или ***тактового генератора****.* Обработка команды занимает несколько периодов тактового генератора. Вообще говоря, выполнение команды можно разделить на три фазы (машинных цикла): *извлечение кода операции; первичная дешифрация кода команды для определения длины ее адресной части и извлечение из памяти адресной части команды; вторичная дешифрация для определения операции и выполнение команды.*

Выполнение каждого из названных циклов требует нескольких периодов тактового генератора (машинных тактов) (рис. 1.4). Совокупное время, требуемое для выборки, декодирования и выполнения команды, образует ***командный цикл****,* или ***цикл выполнения команды****.*

****