## §2. АРХИТЕКТУРА ЭВМ

### 2.1. О ПОНЯТИИ «АРХИТЕКТУРА ЭВМ»

Термин «архитектура» используется в популярной литературе по вычислительной технике достаточно часто, однако определение этого понятия и его содержание могут у разных авторов достаточно различаться. Разберемся в этом вопросе более тщательно.

Начать целесообразно с происхождения термина. Слово «архитектура» в изначальном своем смысле используется в градостроении. Будучи достаточно сложной структурой, современный город состоит из районов, площадей, улиц, домов и т.п., расположенных определенным образом. Жителей города обычно мало интересует, как выглядит конкретный дом и из каких материалов он построен. Зато очень важно знать район, где этот дом расположен, улицы, ведущие к нему, и транспорт, пользуясь которым можно сократить время в пути.

Для того, чтобы ориентироваться в хитросплетении улиц и площадей, в любом городе существует исторически сложившаяся система названий, а также определенная нумерация домов. Наличие общепринятой адресации позволяет однозначно определить положение любого строения и в случае необходимости быстро отыскать его. Именно на существовании такой адресной системы построена работа почты. Во многих случаях расположение улиц и присвоение им имен носит беспорядочный характер. В то же время бывает, что эта деятельность тщательно продумана и является продолжением общей планировки города, т.е. фактически частью его архитектуры. Классическим примером может служить известная система взаимно-перпендикулярных улиц (авеню и стриты) города Нью-Йорка. Помимо чисто практической, архитектура города может иметь еще и художественную ценность (что обычно больше интересует приезжих). Но этот аспект понятия «архитектура» вряд ли переносим на вычислительную технику.

Используя аналогию с градостроительством, естественно понимать под архитектурой ЭВМ ту совокупность их характеристик, которая необходима почьзователю. Это, прежде всего, основные устройства и блоки ЭВМ, а также структура связей между ними. И действительно, если заглянуть, например, в «Толковый словарь по вычислительным системам», мы прочтем там. что термин «архитектура ЭВМ используется для описания принципа действия, конфигурации и взаимного соединения основных логических узлов ЭВМ (вследствие чего термин «архитектура» оказывается ближе к обыденному значению этого слова)».

Однако описание внутренней структуры ЭВМ вовсе не является самоцелью: с точки зрения архитектуры представляют интерес лишь те связи и принципы, которые являются наиболее общими, присущими многим конкретным реализацням вычислительных машин. Часто говорят даже о семействах ЭВМ. т.е. группах моделей, совместимых между собой. В пределах одного семейства основные принципы устройства и функционирования машин одинаковы, хотя отдельные модели могут существенно различаться по производительности, стоимости и другим параметрам. Ярким примером могут служить различные модификации компьютеров PDP фирмы DEC (более известные нашим пользователям по отечественным аналогам - серии ДВК), семейство MSX-машин. к которому принадлежит широко распространенная YAMAHA, а также заполонившие мир IBM-совместимые персональные компьютеры.

Именно то общее, что есть в строении ЭВМ, и относят к понятию архитектуры. Важно отметить, что целью такой общности в конечном счете служит вполне понятное стремление: все машины одного семейства, независимо от их конкретного устройства и фирмы-производителя, должны быть способны выполнять одну и ту же программу (на практике из-за постоянного роста вычислительной мощности техники чаще используется менее жесткий принцип совместимости снизу вверх: все программы данной модели выполнимы на более старших). Отсюда неизбежно следует вывод, что с точки зрения архитектуры важны не все сведения о построении ЭВМ, а только те, которые могут как-то использоваться при программировании и «пользовательской» работе с ЭВМ. Равно как максимально подробная архитектура города не нуждается в описании марок кирпичей, из которых построены дома, и растворов, которыми эти кирпичи скреплены, так и архитектура ЭВМ не содержит описания электронных схем, других деталей реализации, «невидимых» для пользователя (например, внутреннего ускорителя доступа к памяти).

Ниже приводится перечень тех наиболее общих принципов построения ЭВМ, которые относятся к архитектуре:

• структура памяти ЭВМ;

• способы доступа к памяти и внешним устройствам;

• возможность изменения конфигурации компьютера;

• система команд;

• форматы данных;

• организация интерфейса.

Суммируя все вышеизложенное, получаем следующее определение архитектуры:

*«Архитектура - это наиболее общие принципы построения ЭВМ, реализующие*

*программное управление работой и взаимодействием основных ее функциональных*

*узлов».*

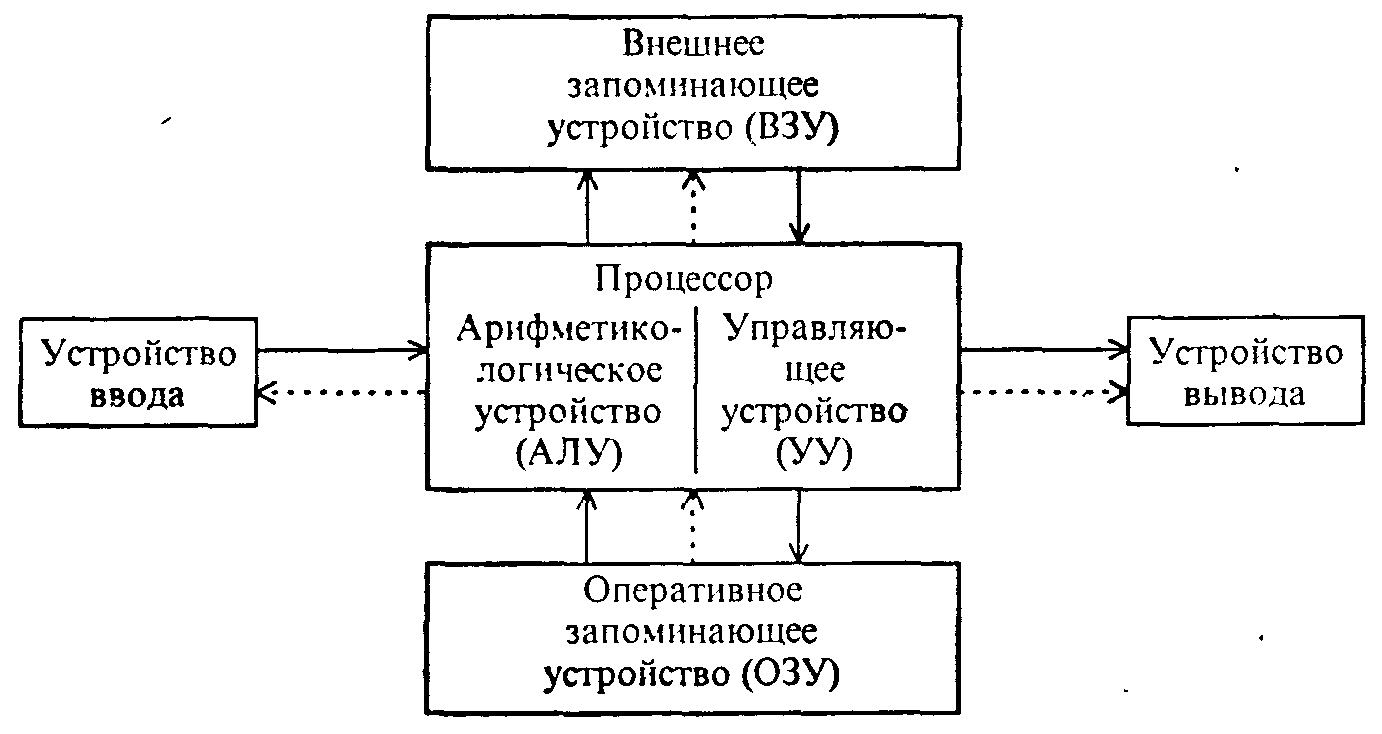
### 1.2. КЛАССИЧЕСКАЯ АРХИТЕКТУРА ЭВМ II ПРИНЦИПЫ ФОН НЕЙМАНА

Основы учения об архитектуре вычислительных машин заложил выдающийся американский математик Джон фон Нейман. Он подключился к созданию первой в мире ламповой ЭВМ ENIAC в 1944 г., когда ее конструкция была уже выбрана. В процессе работы во время многочисленных дискуссий со своими коллегами Г.Голдстайном и А.Берксом фон Нейман высказал идею принципиально новой ЭВМ. В 1946 г. ученые изложили свои принципы построения вычислительных машин в ставшей классической статье «Предварительное рассмотрение логической конструкции электронно-вычислительного устройства». С тех пор прошло полвека, но выдвинутые в ней положения сохраняют актуальность и сегодня.

В статье убедительно обосновывается использование двоичной системы для представления чисел (нелишне напомнить, что ранее все вычислительные машины хранили обрабатываемые числа в десятичном виде). Авторы убедительно продемонстрировали преимущества двоичной системы для технической реализации, удобство и простоту выполнения в ней арифметических и логических операций. В дальнейшем ЭВМ стали обрабатывать и нечисловые виды информации - текстовую, графическую, звуковую и другие, но двоичное кодирование данных по-прежнему составляет информационную основу любого современного компьютера.

Еще одной поистине революционной идеей, значение которой трудно переоценить, является предложенный Нейманом принцип «хранимой программы». Первоначально программа задавалась путем установки перемычек на специальной коммутационной панели. Это было весьма трудоемким занятием: например, для изменения программы машины ENIAC требовалось несколько дней (в то время как собственно расчет не мог продолжаться более нескольких минут - выходили из строя лампы). Нейман первым догадался, что программа может также храниться в виде набора нулей и единиц, причем в той же самой памяти, что и обрабатываемые ею числа. Отсутствие принципиальной разницы между программой и данными дало возможность ЭВМ самой формировать для себя программу в соответствии с результатами вычислений.

Фон Нейман не только выдвинул основополагающие принципы логического устройства ЭВМ, но и предложил ее структуру, которая воспроизводилась в течение первых двух поколений ЭВМ. Основными блоками по Нейману являются устройство управления (УУ) и арифметико-логическое устройство (АЛУ) (обычно объединяемые в центральный процессор), память, внешняя память, устройства ввода и вывода. Схема устройства такой ЭВМ представлена на рис. 4.10. Следует отметить, что внешняя память отличается от устройств ввода и вывода тем, что данные в нее заносятся в виде, удобном компьютеру, но недоступном для непосредственного



*Рис. 4.10.* Архитектура ЭВМ, построенной на принципах фон Неймана. Сплошные линии со стрелками указывают направление потоков информации, пунктирные-управляющих сигналов от процессора к остальными узлам ЭВМ

восприятия человеком. Так, накопитель на магнитных дисках относится к внешней памяти, а клавиатура - устройство ввода, дисплей и печать - устройства вывода.

Устройство управления и арифметико-логическое устройство в современных компьютерах объединены в один блок - процессор, являющийся преобразователем информации, поступающей из памяти и внешних устройств (сюда относятся выборка команд из памяти, кодирование и декодирование, выполнение различных, в том числе и арифметических, операций, согласование работы узлов компьютера). Более детально функции процессора будут обсуждаться ниже.

Память (ЗУ) хранит информацию (данные) и программы. Запоминающее устройство у современных компьютеров «многоярусно» и включает оперативное запоминающее устройство (ОЗУ), хранящее ту информацию, с которой компьютер работает непосредственно в данное время (исполняемая программа, часть необходимых для нее данных, некоторые управляющие программы), и внешние запоминающие устройства (ВЗУ) гораздо большей емкости, чем ОЗУ. но с существенно более медленным доступом (и значительно меньшей стоимостью в расчете на 1 байт хранимой информации). На ОЗУ и ВЗУ классификация устройств памяти не заканчивается - определенные функции выполняют и СОЗУ (сверхоперативное запоминающее устройство), и ПЗУ (постоянное запоминающее устройство), и другие подвиды компьютерной памяти.

В построенной по описанной схеме ЭВМ происходит последовательное считывание команд из памяти и их выполнение. Номер (адрес) очередной ячейки памяти. из которой будет извлечена следующая команда программы, указывается специальным устройством - счетчиком команд в УУ. Его наличие также является одним из характерных признаков рассматриваемой архитектуры.

Разработанные фон Нейманом основы архитектуры вычислительных устройств оказались настолько фундаментальными, что получили в литературе название «фон-неймановской архитектуры». Подавляющее большинство вычислительных машин на сегодняшний день - фон-неймановские машины. Исключение составляют лишь отдельные разновидности систем для параллельных вычислений, в которых отсутствует счетчик команд, не реализована классическая концепция переменной и имеются другие существенные принципиальные отличия от классической модели (примерами могут служить потоковая и редукционная вычислительные машины).

По-видимому, значительное отклонение от фон-неймановской архитектуры произойдет в результате развития идеи машин пятого поколения, в основе обработки информации в которых лежат не вычисления, а логические выводы.

### 2.3. СОВЕРШЕНСТВОВАНИЕ И РАЗВИТИЕ ВНУТРЕННЕЙ СТРУКТУРЫ ЭВМ

В предыдущем разделе была описана классическая структура ЭВМ, соответствующая вычислительным машинам первого и второго поколений. Естественно, что в результате бурного развития технологии производства средств вычислительной техники такая структура не могла не претерпеть определенных прогрессивных изменений.

Как отмечалось выше, появление третьего поколения ЭВМ было обусловлено переходом от транзисторов к интегральным микросхемам. Значительные успехи в миниатюризации электронных схем не просто способствовали уменьшению размеров базовых функциональных узлов ЭВМ, но и создали предпосылки для существенного роста быстродействия процессора. Возникло существенное противоречие между высокой скоростью обработки информации внутри машины и медленной работой устройств ввода-вывода, в большинстве своем содержащих механически движущиеся части. Процессор, руководивший работой внешних устройств, значительную часть времени был бы вынужден простаивать в ожиданииинформации «извнешнего мира», что существенно снижало бы эффективность работы всей ЭВМ в целом. Для решения этой проблемы возникла тенденция к освобождению центрального процессора от функций обмена и к передаче их специальным электронным схемам управления работой внешних устройств.Такие схемы имели различные названия: каналы обмена, процессоры ввода-вывода, периферийные процессоры. Последнее время все чаще используется термин «контроллер внешнего устройства» (или просто контроллер).

Наличие интеллектуальных контроллеров внешних устройств стало важной отличительной чертой машин третьего и четвертого поколений.

Контроллер можно рассматривать как специализированный процессор, управляющий работой «вверенного ему» внешнего устройства по специальным встроенным программам обмена. Такой процессор имеет собственную систему команд. Например, контроллер накопителя на гибких магнитных дисках (дисковода) умеет позиционировать головку на нужную дорожку диска, читать или записывать сектор, форматировать дорожку и т.п. Результаты выполнения каждой операции заносятся во внутренние регистры памяти контроллера и могут быть в дальнейшем прочитаны центральным процессором.

Таким образом, наличие интеллектуальных внешних устройств может существенно изменять идеологию обмена. Центральный процессор при необходимости произвести обмен выдает задание на его осуществление контроллеру. Дальнейший обмен информацией может протекать под руководством контроллера без участия центрального процессора. Последний получает возможность «заниматься своим делом», т.е. выполнять программу дальше (если по данной задаче до завершения обмена ничего сделать нельзя, то можно в это время решать другую).

Перейдем теперь к обсуждению вопроса о внутренней структуре ЭВМ, содержащей интеллектуальные контроллеры, изображенной на рис. 4.11. Из рисунка видно, что для связи между отдельными функциональными узлами ЭВМ используется общая шина (часто ее называют магистралью). Шина состоит из трех частей:

• шина данных, по которой передается информация;

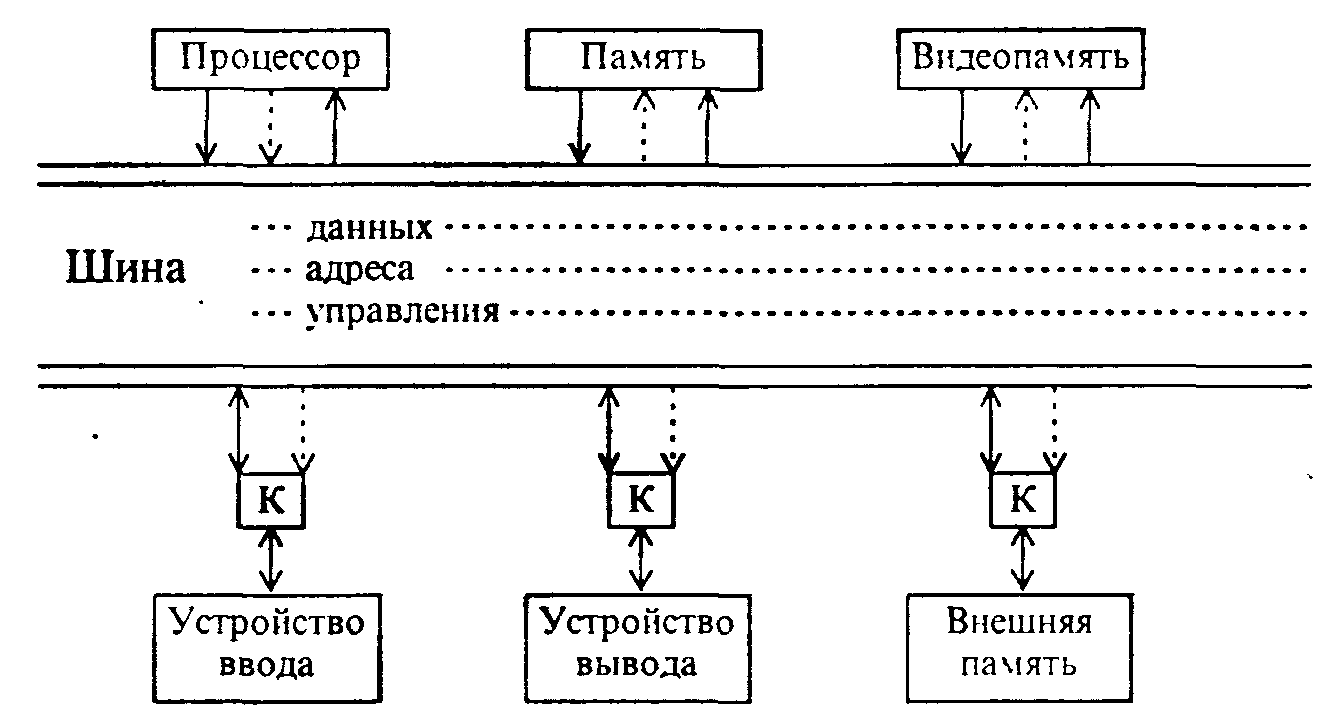
• шина адреса, определяющая, куда передаются данные;

• шина управления, регулирующая процесс обмена информацией.

Отметим, что существуют модели компьютеров, у которых шины данных и адреса для экономии объединены. У таких машин сначала на шину выставляется адрес, а затем через некоторое время данные; для какой именно цели используется шина в данный момент, определяется сигналами на шине управления.

Описанную схему легко пополнять новыми устройствами - это свойство называют открытостью архитектуры. Для пользователя открытая архитектура означает возможность свободно выбирать состав внешних устройств для своего компьютера, т.е. конфигурировать его в зависимости от круга решаемых задач.

На рис. 4.11 представлен новый по сравнению с рис. 4.10 вид памяти - видео-ОЗУ (видеопамять). Его появление связано с разработкой особого устройства вывода - дисплея. Основной частью дисплея служит электронно-лучевая трубка, которая отображает информацию примерно так же, как это происходит в телевизоре (к некоторым дешевым домашним моделям компьютеров просто подключается обычный телевизор). Очевидно, что дисплей, не имея механически движущихся частей, является «очень быстрым» устройством отображения информации. Поэтому для ЭВМ третьего и четвертого поколений он является неотъемлемой частью (хотя впервые дисплей был реализован на некоторых ЭВМ второго поколения, например, на «МИР-2» - очень интересной во многих отношениях отечественной разработке).



*Рис. 4.11.* Шинная архитектура ЭВМ

Для получения на экране монитора стабильной картинки ее надо где-то хранить. Для этого и существует видеопамять. Сначала содержимое видеопамяти формируется компьютером, а затем контроллер дисплея выводит изображение на экран. Объем видеопамяти существенно зависит от характера информации (текстовая или графическая) и от числа цветов изображения. Конструктивно она может быть выполнена как обычное ОЗУ или содержаться непосредственно в контроллере дисплея (именно поэтому на рис. 4.11 она показана пунктиром).

Остановимся еще на одной важной особенности структуры современных ЭВМ. Поскольку процессор теперь перестал быть центром конструкции, стало возможным реализовывать прямые связи между устройствами ЭВМ. На практике чаще всего используют передачу данных из внешних устройств в ОЗУ и наоборот. Режим, при котором внешнее устройство обменивается непосредственно с ОЗУ без участия центрального процессора, называется прямым доступом к памяти (ПДП). Для его реализации необходим специальный контроллер. Подчеркнем, что режим ПДП в машинах первого и второго поколений не существовал. Поэтому встречающаяся иногда схема ЭВМ, на которой данные из устройств ввода напрямую поступают в ОЗУ, не соответствует действительности: данные при отсутствии контроллера ПДП всегда сначала принимаются во внутренние регистры процессора и лишь затем в память.

При описании магистральной структуры мы упрощенно предполагали, что все устройства взаимодействуют через общую шину. С точки зрения архитектуры этого вполне достаточно. Упомянем все же, что на практике такая структура применяется только для ЭВМ с небольшим числом внешних устройств. При увеличении потоков информации между устройствами ЭВМ единственная магистраль перегружается, что существенно тормозит работу компьютера. Поэтому в состав ЭВМ могут вводиться одна или несколько дополнительных шин. Например, одна шина может использоваться для обмена с памятью, вторая -для связи с «быстрыми», а третья - с «медленными» внешними устройствами. Отметим, что высокоскоростная шина данных ОЗУ обязательно требуется при наличии режима ПДП.

Завершая обсуждение особенностей внутренней структуры современных ЭВМ, укажем несколько характерных тенденций в ее развитии. Во-первых, постоянно расширяется и совершенствуется набор внешних устройств, что приводит, как описывалось выше, к усложнению системы связей между узлами ЭВМ. Во-вторых, вычислительные машины перестают быть однопроцессорными. Помимо центрального, в компьютере могут быть специализированные процессоры для вычисления с плавающей запятой (так называемые математические сопроцессоры), видеопроцессоры для ускорения вывода информации на экран дисплея и т.п. Развитие методов параллельных вычислений также вызывает к жизни вычислительные системы достаточно сложной структуры, в которых одна операция выполняется сразу несколькими процессорами. В-третьих, наметившееся стремление иметь быстродействующие машины не только для вычислений, но и для логического анализа информации, также может привести в ближайшиегоды к серьезному пересмотру традиционной фон-неймановской архитектуры.

Еще одной особенностью развития современных ЭВМ является все ускоряющееся возрастание роли межкомпьютерных коммуникаций. Все большее количество компьютеров объединяются в сети и обрабатывают имеющуюся информацию совместно.

Таким образом, внутренняя структура вычислительной техники постоянно совершенствовалась и будет совершенствоваться. Вместе с тем, на данный момент подавляющее большинство существующих ЭВМ, несмотря на имеющиеся различия, по-прежнему состоит из одинаковых узлов и основано на общих принципах фон-неймановской архитектуры.

### 2.4. ОСНОВНОЙ ЦИКЛ РАБОТЫ ЭВМ

В данном разделе коротко рассмотрена последовательность действий при выполнении команды в ЭВМ. Можно утверждать, что рабочий цикл в общем виде одинаков для всех фон-неймановских машин.

Как уже отмечалось в *п.2.2,* важной составной частью фон-неймановской архитектуры является счетчик адреса команд. Этот специальный внутренний регистр процессора всегда указывает на ячейку памяти, в которой хранится следующая команда программы. При включении питания или при нажатии на кнопку сброса (начальной установки) в счетчик аппаратно заносится стартовый адрес находящейся в ПЗУ программы инициализации всех устройств и начальной загрузки. Дальнейшее функционирование компьютера определяется программой. Таким образом, вся деятельность ЭВМ - это непрерывное выполнение тех или иных программ, причем программы эти могут в свою очередь загружать новые программы и т.д.

Каждая программа состоит из отдельных машинных команд. Каждая машинная команда, в свою очередь, делится на ряд элементарных унифицированных составных частей, которые принято называть тактами. В зависимости от сложности команды она может быть реализована за разное число тактов. Например, пересылка информации из одного внутреннего регистра процессора в другой выполняется за несколько тактов, а для перемножения двух целых чисел их требуется на порядок больше. Существенное удлинение команды происходит, если обрабатываемые данные еще не находятся внутри процессора и их приходится считывать из ОЗУ.

При выполнении каждой команды ЭВМ проделывает определенные стандартные действия:

1) согласно содержимому счетчика адреса команд, считывается очередная команда программы (ее код обычно заносится на хранение в специальный регистр УУ, который носит название регистра команд);

2) счетчик команд автоматически изменяется так, чтобы в нем содержался адрес следующей команды (в простейшем случае для этой цели достаточно к текущему значению счетчика прибавить некоторую константу, определяющуюся длиной команды);

3) считанная в регистр команд операция расшифровывается, извлекаются необходимые данные и над ними выполняются требуемые действия.

Затем во всех случаях, за исключением команды останова или наступления прерывания (см. ниже в п. 3.5), все описанные действия циклически повторяются.

После выборки команды останова ЭВМ прекращает обработку программы. Для выхода из этого состояния требуется либо запрос от внешних устройств, либо перезапуск машины.

Рассмотренный основной алгоритм работы ЭВМ позволяет шаг за шагом выполнить хранящуюся в ОЗУ линейную программу. Если же требуется изменить порядок вычислений для реализации развилки или цикла, достаточно в счетчик команд занести требуемый адрес (именно так происходит условный или безусловный переход).

В компьютерах на базе микропроцессоров INTEL 80286 и более поздних моделей для ускорения основного цикла выполнения команды используется метод конвейеризации (иногда применяется термин «опережающая выборка»). Идея состоит в том, что несколько внутренних устройств процессора работают параллельно: одно считывает команду, другое дешифрует операцию, третье вычисляет адреса используемых операндов и т.д. В результате по окончании команды чаще всего оказывается, что следующая уже выбрана из ОЗУ, дешифрована и подготовлена к исполнению. Отметим, что в случае нарушения естественного порядка выполнения команд в программе (например, при безусловном переходе) опережающая выборка оказывается напрасной и конвейер очищается. Следующая за переходом команда выполняется дольше, так как, чтобы конвейер «заработал на полную мощность», необходимо его предварительно заполнить. Иными словами, в конвейерной машине время выполнения программы может зависеть не только от составляющих ее команд, но и от их взаимного расположения.

### 2.5. СИСТЕМА КОМАНД ЭВМ И СПОСОБЫ ОБРАЩЕНИЯ К ДАННЫМ

Важной составной частью архитектуры ЭВМ является система команд. Несмотря на большое число разновидностей ЭВМ, на самом низком («машинном») уровне они имеют много общего. Система команд любой ЭВМ обязательно содержит следующие группы команд обработки информации.

1. Команды передачи данных (перепись), копирующие информациюиз одного места в другое.

2. Арифметические операции, которым фактически обязана своим названием вычислительная техника. Конечно, доля вычислительных действий в современном компьютере заметно уменьшилась, но они по-прежнему играют в программах важную роль. Отметим, что к основным арифметическим действиям обычно относятся сложение и вычитание (последнее в конечном счете чаще всего тем или иным способом также сводится к сложению). Что касается умножения и деления, то они во многих ЭВМ выполняются по специальным программам.

3. Логические операции, позволяющие компьютеру анализировать обрабатываемую информацию. Простейшими примерами могут служить сравнение, а также известные логические операции И, ИЛИ, НЕ (инверсия). Кроме того к ним часто добавляются анализ отдельных битов кода, их сброс и установка.

4. Сдвиги двоичного кода влево и вправо. Для доказательства важности этой группы команд достаточно вспомнить правило умножения столбиком: каждое последующее произведение записывается в такой схеме со сдвигом на одну цифру влево. В некоторых частных случаях умножение и деление вообще может быть заменено сдвигом (вспомните, что дописав или убрав ноль справа, т.е. фактически осуществляя сдвиг десятичного числа, можно увеличить или уменьшить его в 10 раз).

5. Команды ввода и вывода информации для обмена с внешними устройствами. В некоторых ЭВМ внешние устройства являются специальными служебными адресами памяти, поэтому ввод и вывод осуществляется с помощью команд переписи.

6. Команды управления, реализующие нелинейные алгоритмы. Сюда прежде всего следует отнести условный и безусловный переход, а также команды обращения к подпрограмме (переход с возвратом). Некоторые ЭВМ имеют специальные команды для организации циклов, но это не обязательно: цикл может быть сведен к той или иной комбинации условного и безусловного переходов. Часто к этой же группе команд относят немногочисленные операции по управлению процессором -типа «останов» или НОП («нет операции»). Иногда их выделяют в особую группу.

С ростом сложности устройства процессора увеличивается и число команд, анализирующих состояние управляющих битов и воздействующих на них. Здесь для примера можно назвать биты режима работы процессора и биты управления механизмами прерываний от внешних устройств.

В последнее время все большую роль в наборе команд играют команды для преобразования из одного формата данных в другой (например, из 8-битного в 16-битный и т.п.), которые заметно упрощают обработку данных разного типа, но в принципе могут быть заменены последовательностью из нескольких более простых команд.

Рассматриваясистему команд, нельзя не упомянуть о двух современных взаимно конкурирующих направлениях в ее построении: компьютер с полным набором команд CISC (Complex Instruction Set Computer) и с ограниченным набором - RISC (Reduced Instruction Set Computer). Разделение возникло из-за того, что основную часть времени компьютеру приходится выполнять небольшую часть из своего набора команд, остальные же используются эпизодически (в одной из популярных статей это в шутку сформулировано в виде следующей наглядной аналогии: «20% населения выпивают 80% пива»). Таким образом, если существенно ограничить набор операций до наиболее простых и коротких, зато тщательно оптимизировать их, получится достаточно эффективная и быстродействующая RISC-машина. Правда за скорость придется платить необходимостью программной реализации «отброшенных» команд, но часто эта плата бывает оправданной: например, для научных расчетов или машинной графики быстродействие существенно важнее проблем программирования. Подробнее вопросы, связанные с системой команд современных микропроцессоров, будут рассмотрены ниже в этой главе.

Подводя итог, еще раз подчеркнем, что основной набор команд довольно слабо изменился в ходе бурной эволюции ЭВМ. В то же время способы указания адреса расположения информации в памяти претерпели значительное изменение и заслуживают особого рассмотрения.

Команда ЭВМ обычно состоит из двух частей - операционной и адресной. Операционная часть (иначе она еще называется кодом операции - КОП) указывает, какое действие необходимо выполнить с информацией. Адресная часть описывает, где используемая информация хранится. У нескольких немногочисленных команд управления работой машины адресная часть может отсутствовать, например, в команде останова; операционная часть имеется, всегда.

Код операции можно представить себе как некоторый условный номер в общем списке системы команд. В основном этот список построен в соответствии с определенными внутренними закономерностями, хотя они не всегда очевидны.

Адресная часть обладает значительно большим разнообразием и ее следует рассмотреть подробнее.

Прежде всего отметим, что команды могут быть одно-, двух- и трехадресные в зависимости от числа участвующих в них операндов.

Первые ЭВМ имели наиболее простую и наглядную трехадресную систему команд. Например: взять числа из адресов памяти А1 и А2, сложить их и сумму поместить в адрес A3. Если для операции требовалось меньшее число адресов, то лишние просто не использовались. Скажем, в операции переписи указывались лишь ячейки источника и приемника информации А1 и A3, а содержимое А2 не имело никакого значения.

Трехадресная команда легко расшифровывалась и была удобна в использовании, но с ростом объемов ОЗУ ее длина становилась непомерно большой. Действительно, длина команды складывается из длины трех адресов и кода операции. Отсюда следует, например, что для скромного ОЗУ из 1024 ячеек только для записи адресной части одной команды требуется 3\*10 = 30 двоичных разрядов, что для технической реализации не очень удобно. Поэтому появились двухадресные машины, длина команды в которых сокращалась за счет исключения адреса записи результата. В таких ЭВМ результат операции оставался в специальном регистре (сумматоре) и был пригоден для использования в последующих вычислениях. В некоторых машинах результат записывался вместо одного из операндов.

Дальнейшее упрощение команды привело к созданию одноадресных машин. Рассмотрим систему команд такой ЭВМ на конкретном простом примере. Пусть надо сложить числа, хранящиеся в ячейках с адресами ОЗУ А1 и А2, а сумму поместить в ячейку с адресом A3. Для решения этой задачи одноадресной машине потребуется выполнить три команды:

• извлечь содержимое ячейки А1 в сумматор;

• сложить сумматор с числом из А2;

• записать результат из сумматора в A3.

Может показаться, что одноадресной машине для решения задачи потребуется втрое больше команд, чем трехадресной. На самом деле это не всегда так. Попробуйте самостоятельно спланировать программу вычисления выражения А5 = (А1 + А2)\*АЗ/А4 и вы обнаружите, что потребуется три трехадресных команды и всего пять одноадресных. Таким образом, одноадресная машина в чем-то даже эффективнее, так как она не производит ненужной записи в память промежуточных результатов.

Ради полноты изложения следует сказать о возможности реализации безадресной (нуль-адресной) машины, использующей особый способ организации памяти -стек. Понимание принципов устройства такой машины потребовало бы некоторых достаточно подробных разъяснений. Сейчас безадресные ЭВМ практически не применяются. Поэтому ограничимся лишь упоминанием того факта, что устроенная подобным образом система команд лежала в основе некоторых программируемых микрокалькуляторов (например, типа «БЗ-21» и «БЗ-34» и им подобных).

До сих пор в описании структуры машинной команды мы пользовались интуитивным понятием об адресе информации. Рассмотрим теперь вопрос об адресации элементов ОЗУ более подробно и строго. Наиболее просто была организована память в ЭВМ первых двух поколений. Она состояла из отдельных ячеек, содержимое каждой из которых считывалось или записывалось как единое целое. Каждая ячейка памяти имела свой номер, который и получил название адреса. Очевидно, что адреса соседних ячеек ОЗУ являются последовательными целыми числами, т.е. отличаются на единицу. В рассматриваемых ЭВМ использовались данные только одного типа (вещественные числа), причем их длина равнялась длине машинной команды и совпадала с разрядностью памяти и всех остальных устройств машины. Для примера укажем, что ячейка типичной ЭВМ второго поколения состояла из 36 двоичных разрядов.

Очень часто программа предназначалась для обработки по одним и тем же формулам определенного количества содержимого последовательно расположенных ячеек (в языках высокого уровня такого рода структуры получили впоследствии название массивов). В ЭВМ первых двух поколении были предусмотрены особые механизмы циклической обработки массивов информации. С этой целью в машинных командах помимо обычных адресов можно было использовать модифицируемые, у которых специальный управляющий бит был установлен в единицу. К помеченным таким образом модифицируемым адресам при выполнении команды прибавлялось значение из специальных индексных ячеек. Меняя содержимое индексных ячеек, можно было получать доступ к различным элементам массива. Особо подчеркнем, что формирование результирующего адреса осуществлялось в УУ в момент исполнения команды, поэтому исходная команда в ОЗУ сохранялась без изменений.

Описанный механизм модификации адресов существенно упрощал написание циклических программ, таких как нахождение суммы последовательных ячеек ОЗУ, копирование отдельных участков памяти и т.п.

В ЭВМ третьего поколения идеология построения памяти существенно изменилась: минимальная порция информации для обмена с ОЗУ была установлена равной 8 двоичных разрядов, т.е. один байт. Стало возможным обрабатывать несколько типов данных: символы текста (1 байт), целые числа (2 байта), вещественные числа обычной или двойной точности (4 или 8 байт соответственно). В связи с этим была введена новая условная единица измерения информации - машинное слово. Оно равнялось 4 байтам и соответствовало длине стандартного вещественного числа. Все объемы информации начали измеряться в единицах, кратных слову: двойное слово, полуслово и т.п. Естественно, что адрес (номер ячейки ОЗУ) в машинах с байтовой организацией стал относится к отдельному байту; байты памяти имеют возрастающие на единицу номера. Слово состоит из нескольких последовательно расположенных байтов. В качестве адреса слова удобно принимать адрес одного из образующих его байтов (обычно используется младший байт, имеющий наименьший номер). Таким образом, адреса слов меняются уже не через единицу; их приращение зависит от длины машинного слова в байтах и равняется четырем.

Размер машинного слова был, по-видимому, выбран исходя из форматов обрабатываемой информации, а не в связи с разрядностью каких-либо устройств. Для подтверждения этого приведем несколько фактов о типичных ЭВМ третьего поколения из семейства ЕС. Арифметико-логическое устройство модели «ЕС-1022» имело 16 двоичных разрядов, «ЕС-1033» - 32 разряда, а «ЕС-1050» - 64 разряда. В то же время за одно обращение к оперативной памяти в «ЕС-1022» и «ЕС-1033» выбиралось 4 байта, в «ЕС-1050» - 8 байт (а в «ЕС-1045» - 16 байт). Таким образом, разнообразие цифр свидетельствует, что 32 разряда (4 байта) не являлись каким-то технически выделенным объемом информации.

В машинах третьего поколения появились иеще несколько особенностей: разная длина команд в зависимости от способа адресации данных, наличие специальной сверхоперативной регистровой памяти, вычисление эффективного адреса ОЗУ как суммы нескольких регистров и т.п. Все это получило дальнейшее развитие в компьютерах четвертого поколения, для которых разрядность микропроцессора стала одной из важнейших характеристик. Рассмотрение особенностейстроения памяти ЭВМ четвертого поколения отложим до следующего раздела.