

С. А. МАЙОРОВ ♦ Г. И. НОВИКОВ

СТРУКТУРА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

10

110

10 011

00 11101

0001 001100

С. А. МАЙОРОВ, Г. И. НОВИКОВ

СТРУКТУРА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

*Второе издание
переработанное и дополненное*



ЛЕНИНГРАД «МАШИНОСТРОЕНИЕ»
ЛЕНИНГРАДСКОЕ ОТДЕЛЕНИЕ
1979

ББК 32.973

М14

УДК 681.32

Майоров С. А., Новиков Г. И.

М14 Структура электронных вычислительных машин. — Л.: Машиностроение. Ленингр. отд-ние, 1979. — 384 с., ил.

1 р. 60 к.

В книге изложены принципы построения и методы схемотехнического проектирования ЭВМ и устройств цифровой вычислительной техники. В ней описаны структуры ЭВМ общего назначения, мини- и микро-ЭВМ; рассмотрены методы проектирования структурных и функциональных схем процессоров, каналов ввода — вывода и средств управления внешними устройствами, методы синтеза операционных и управляющих автоматов.

Во втором издании (1-е изд. вышло в 1970 г. под названием «Структура цифровых вычислительных машин») изложен функциональный подход к проектированию структур ЭВМ, а также описаны новые принципы построения схем управления и обработки информации.

Книга предназначена для инженерно-технических работников, занимающихся проектированием и эксплуатацией ЭВМ и систем на их основе.

М $\frac{30502-50}{038(01)-79}$ 50—79

2405000000

ББК 32.973
6Ф7.3

ПРЕДИСЛОВИЕ

Вычислительная техника играет значительную роль в научно-техническом прогрессе нашей страны, способствуя повышению эффективности общественного производства, улучшению качества продукции, росту производительности труда. С каждым годом увеличиваются масштабы применения электронных вычислительных машин (ЭВМ) во всех отраслях народного хозяйства, в связи с чем возрастают требования к характеристикам ЭВМ и, следовательно, к принципам их построения и проектирования.

Цель этой книги, как и ее первого издания, — изложить основные принципы построения и проектирования схем ЭВМ. Однако это издание значительно отличается от предыдущего в следующих аспектах.

В основу методики изложения положен принцип первичности функций в отношении структур, из которого следует, что структура ЭВМ, т. е. конфигурация схем, предопределяется функцией (назначением) ЭВМ. Этот принцип предопределил порядок изложения материала в книге. В отличие от традиционного подхода, когда принципы построения и проектирования ЭВМ начинают рассматриваться от элементов и заканчиваются описанием ЭВМ как целого, материал излагается в обратном порядке: сначала выявляются функции ЭВМ, которые затем детализируются до отдельных устройств, узлов и, наконец, до элементов. Такой порядок изложения представляется наиболее результативным, поскольку, во-первых, совпадает с порядком проектирования ЭВМ и, во-вторых, упрощает изучение основ вычислительной техники, так как в этом случае каждое схемотехническое решение воспринимается как необходимое, предопределенное предыдущими решениями.

Для унификации многочисленных структурных решений, применяемых в инженерной практике, используются понятия операционного устройства и интерфейса. Эти понятия вводятся на функциональном уровне и соответствуют преобразователю дискретной информации с произвольной функцией и алгоритму обмена информацией между устройствами. Столь широкое толкование указанных понятий создает основу для построения методики проектирования разнообразных операционных устройств, в частности процессоров, каналов ввода—вывода и контроллеров внешних устройств ЭВМ.

В основу проектирования операционных устройств различного назначения положен принцип функционального микропрограм.

мирования и представления устройства как композиции операционного и управляющего автоматов. При этом содержание термина микропрограммирование расширено по сравнению с традиционным. В этой книге микропрограммирование понимается как способ описания функций операционных устройств безотносительно к техническим средствам, используемым для их реализации. Такое толкование микропрограммирования позволяет формализовать синтез структур любых операционных устройств независимо от способа управления работой устройства. Наиболее детально описываются принципы структурной организации операционных автоматов, формализованные методы синтеза которых только начинают разрабатываться. Следует отметить, что принципы построения и методы проектирования операционных и управляющих автоматов являются той основой, на которой базируется теория и практика проектирования большей части устройств ЭВМ.

В инженерной практике используется широкая номенклатура устройств, каждое из которых может быть реализовано большим числом способов. Естественно, что в рамках книги, затрагивающей обширный круг вопросов вычислительной техники, невозможно даже перечислить все многообразие способов построения ЭВМ и устройств. По этой причине основное внимание уделялось изложению общих принципов и основополагающих вариантов схем, а частные случаи использовались только для их иллюстрации.

Авторы стремились излагать материал в наиболее доступной форме и поэтому намеренно опускали из рассмотрения отдельные детали, которые могут оказаться в какой-то степени существенными при практическом применении описываемых схем и методов. Предполагается, что читатель знаком с двоичной арифметикой, булевой алгеброй, основами теории автоматов (переключаемых схем) и программирования.

Авторы глубоко благодарны всем, кто прислал свои замечания по первому изданию книги, и с признательностью примут новые отзывы и замечания, которые просим направлять в адрес Ленинградского отделения издательства «Машиностроение».

Методы описания и проектирования столь сложных объектов, какими являются ЭВМ, базируются на основополагающих принципах, сформулированных в общей теории систем [47]. Поэтому, прежде чем приступить к изложению способов построения и методов проектирования ЭВМ, раскроем смысл основных понятий и принципов, которые относятся к общей теории систем и широко используются для ЭВМ.

Функция, структура и организация систем. *Система* — это совокупность элементов, объединенных в одно целое для достижения определенной цели. При этом под целью понимается множество результатов, определяемых назначением системы. ЭВМ в смысле указанного определения является системой, предназначенной для автоматизации вычислений на основе алгоритмов. Дополнительно отметим, что понятие система приложимо как к ЭВМ в целом, так и к отдельным частям ЭВМ, например к устройствам ЭВМ. И в дальнейшем будем пользоваться термином система в качестве эквивалента фразы «ЭВМ или отдельное устройство», считая, что все утверждения о системах в равной степени относятся и к ЭВМ, и устройствам ЭВМ.

Чтобы описать систему, необходимо определить функцию системы и структуру системы.

Функция системы — правило получения результатов, определяемых назначением системы. Иначе говоря, функция системы — это описание процессов, которые имеют место в системе. Функции систем стремятся описывать в математической форме ввиду ее компактности и четкости. Наряду с этим, для описания функций широко используется и словесная форма, которая, впрочем, является далеко не худшим способом выражения существа вещей. Функции ЭВМ чаще всего описываются в форме алгоритмов.

Структура системы — фиксированная совокупность элементов и связей между ними. Структура наглядно изображает, как устроена система — из каких частей она состоит и как эти части связаны друг с другом. Математической, а в этом смысле наиболее абстрактной и универсальной, формой изображения структуры является *граф*. Граф есть совокупность вершин и дуг (ребер), представляющих односторонние (соответственно двусторонние) связи между вершинами. Вершины графа отождествляются с элементами (минимальными частями) системы, а дуги и ребра графа — со связями между соответствующими элементами.

Инженерная форма отображения структуры — *схема*. Схема и граф тождественны по своему содержанию и различаются лишь по форме. В схемах для обозначения элементов используются различные геометрические фигуры, разнообразие форм которых облегчает чтение схем.

Система описана, если заданы ее функция и структура. Функция определяет порядок процессов в системе, а структура — состав и взаимосвязь частей (элементов), из которых состоит система.

Системам присуще следующее качество. Свойства совокупности элементов, объединенных в одну систему, не являются простой суммой свойств элементов, а имеют новое качество, отсутствующее в элементах. Например, в совокупности электронных элементов (транзисторов, резисторов и т. п.), определенным образом соединенных между собой, появляется эффект, который отождествляется с операциями математической логики, т. е. совокупность электрических элементов превращается в систему, функции которой описываются не законами электротехники, а законами математической логики. Такого рода система из электронных элементов приобретает новое свойство и становится логическим элементом. В свою очередь, объединение логических элементов, каждый из которых реализует логическую операцию, приводит к схеме, которая обладает свойством складывать числа. Принцип (способ), по которому объединение элементов приводит к появлению новых свойств, отличных от свойств элементов, называется *принципом организации*. Другими словами, *организация* — это способ аранжировки с целью осуществления определенных функций в системах, состоящих из большого числа элементов.

Организация — понятие более высокого ранга, чем функция и структура. Конкретный принцип организации — это способ построения различных систем, обладающих одинаковыми свойствами, т. е. один принцип организации, применяемый к различным случаям, приводит к системам с различными функциями и структурами. Так что функция и структура — это конкретизация принципа организации, всего лишь один вариант организации. В свою очередь, различные принципы организации приводят к системам, различающимся своими функциями и структурами, но тождественным по своим свойствам, своему назначению.

Когда речь идет о принципе порождения функций, необходимых и достаточных для обеспечения определенных свойств систем, пользуются термином *функциональная организация*. *Функциональная организация* — это принципы построения абстрактных систем, заданных своими функциями. Об абстрактной системе известно только ее назначение и не известно, как она устроена, из каких элементов состоит, т. е. абстрактная система — это лишь описание, существующее на бумаге. Когда речь идет о принципе порождения структур, необходимых и достаточных для реализации заданных функций, используется термин *структурная организация*. Структурная организация — это принципы перевода

абстрактных систем, заданных в виде функций, в материальные системы, состоящие из физически существующих элементов.

Проектирование систем — ЭВМ и составляющих их устройств — базируется на определенных принципах организации систем с требуемыми свойствами: автоматически выполнять вычисления, хранить информацию, выполнять арифметические операции и т. д. Эти принципы изобретаются людьми и используются всякий раз, когда необходимо создать систему с заданными свойствами.

Принцип иерархии в сложных системах. Существенной особенностью ЭВМ является сложность. *Сложность* — это свойство систем, состоящих из большого числа элементов. Для сложных систем характерно, что функция, реализуемая системой, не может быть представлена как композиция функций, реализуемых наименьшими элементами системы. Так, порядок функционирования ЭВМ невозможно рассмотреть с точки зрения лишь электрических процессов, происходящих в цепях ЭВМ. Функции ЭВМ выявляются только в том случае, если процессы в ЭВМ рассматриваются в информационном и алгоритмическом аспекте. Это объясняется эффектом организации, порождающим в совокупностях элементов новые свойства: в совокупности электронных элементов ЭВМ — свойство выполнять логические операции над значениями «ложь» и «истина»; в совокупности логических элементов — свойство складывать или сравнивать числа; в совокупности устройств — свойство реализовать вычисления на основе алгоритма. Сказанное свидетельствует об органической потребности в использовании нескольких форм описания функции и структуры системы — *иерархии функций и структур*.

Иерархический подход к описанию сложных систем предполагает, что на высшем уровне иерархии система рассматривается как один элемент, имеющий входы и выходы для связи с внешними объектами. В этом случае функция системы не может быть задана подробно и представляется как отображение состояний входов на состояния выходов системы. Чтобы раскрыть устройство и порядок функционирования системы, глобальная функция и сама система разделяются на части — функции и структурные элементы следующего, более низкого уровня иерархии. В свою очередь, эти части опять детализируются, и так до тех пор, пока функция и структура системы не будут раскрыты полностью, с необходимой степенью подробности.

Становится очевидным, что *элемент* — это прежде всего удобное понятие, а не физическое свойство. То, что элементарно на одном уровне рассмотрения процессов и структуры системы, оказывается разложимым на совокупность объектов — элементов низшего уровня, взаимосвязь которых порождает определенные свойства в элементах высшего уровня иерархии.

Таким образом, любая сложная система содержит в себе иерархию процессов и элементов, наличие которой приводит к необходимости использовать иерархию функций и структур для

описания того, как устроена и работает система. Нижележащий уровень иерархии раскрывает сущность процессов и устройство элементов, относящихся к более высокому уровню.

Стратегия проектирования сложных систем. Устройство и порядок функционирования систем, в том числе и ЭВМ, предопределяются, с одной стороны, назначением системы, а с другой стороны, — элементной базой, «материалом», из которого строится система. Отсюда следует, что система должна проектироваться исходя из ее назначения с учетом свойств материальной базы, существующей на момент проектирования или потенциально возможной.

Наиболее естественным представляется следующий подход к проектированию систем. Пусть назначение системы задается в виде функции F . Исходя из содержания и свойств функции F , можно определить номенклатуру элементов, пригодную для создания системы. Когда номенклатура элементов определена, становится известной номенклатура функций $\varphi_1, \dots, \varphi_N$, реализуемых элементами. Заметим, что функции системы F относятся к верхнему уровню иерархии, и, следовательно, она описана в базисе операций, отличных от операций, которые используются в функциях $\varphi_1, \dots, \varphi_N$, относящихся к более низкому уровню описания системы. Функции элементов $\varphi_1, \dots, \varphi_N$ более просты, чем функция системы F . По этой причине, чтобы выявить структуру S , реализующую функцию F , необходимо путем формальных преобразований построить эквивалентную F функцию Φ , представленную в виде композиции функций $\varphi_1, \dots, \varphi_N$. Этим самым функция F будет детализирована до элементарных функций $\varphi_1, \dots, \varphi_N$, для реализации каждой из которых имеется определенный структурный элемент. Местоположение функций $\varphi_1, \dots, \varphi_N$ в записи функции Φ предопределяет места соответствующих элементов в структуре. Поэтому функцию Φ можно использовать в качестве формы для построения структуры S , состоящей из элементов типа $\varphi_1, \dots, \varphi_N$ и реализующей заданную функцию $\Phi = F$, т. е. функцию системы. Если система сложная, указанная процедура повторяется по отношению к каждой из функций $\varphi_1, \dots, \varphi_N$, реализуемых элементами структуры S . В результате этого будет выявлена структура элементов верхнего уровня иерархии и т. д.

Описанный подход к проектированию систем базируется на предположении о *доминирующей роли функций в отношении структур*. Это означает, что функции первичны, а структуры, их реализующие, вторичны, т. е. функция предопределяет структуру. Как следствие доминирующей роли функций, структуры системы, определенные для верхних уровней иерархии, играют преобладающую роль в отношении структур низших уровней, т. е. состав элементов и связей в структурах верхних уровней предопределяет состав элементов и связей в структурах низших уровней. Из сказанного следует, что системы должны проектироваться по принципу «сверху—вниз» — от верхнего уровня

представления функций и структур к нижнему. Подход к проектированию систем, основанный на признании доминирующей роли функций в отношении структур, можно назвать *функциональным подходом*.

Способы преобразования функций, присущих верхнему уровню иерархии, в функции, соответствующие нижнему уровню, а также способы порождения структур по заданным функциям являются следствием определенных принципов функциональной и структурной организации систем. Эти способы порой удается формализовать и тем самым упростить процесс проектирования, а когда формальные методы отсутствуют, приходится прибегать к неформальным методам синтеза систем, которые в таких случаях носят характер инженерной импровизации.

ПРИНЦИПЫ ПОСТРОЕНИЯ И ФУНКЦИОНИРОВАНИЯ ЭВМ

1.1. Основные факторы, влияющие на принципы построения ЭВМ

ЭВМ — это искусственная (инженерная) система, предназначенная для выполнения вычислений на основе алгоритмов. Принципы построения ЭВМ определяются, с одной стороны, назначением ЭВМ и, с другой стороны, элементной базой — набором элементов, который может быть использован для создания ЭВМ. Назначение ЭВМ — выполнение вычислений на основе алгоритмов, и поэтому свойства алгоритмов предопределяют принципы построения ЭВМ — организацию ЭВМ.

К числу важнейших свойств алгоритмов, наиболее существенно влияющих на организацию ЭВМ, относятся: 1) дискретность информации, с которой оперируют алгоритмы; 2) конечность и элементарность набора операций, выполняемых при реализации алгоритмов; 3) детерминированность вычислительных процессов, порождаемых алгоритмами. Перечисленные свойства алгоритмов обуславливают необходимость представления информации в дискретной (числовой, символьной) форме, реализации в ЭВМ ограниченного числа достаточно простых операций и использования алгоритмов как источника управления процессом вычислений.

Характеризуя назначение ЭВМ, указывают не только класс алгоритмов (задач), выполнение которых является функцией ЭВМ, но и требования к производительности и надежности ЭВМ. Производительность определяется числом задач, решаемых ЭВМ в единицу времени, и для достижения требуемой производительности ЭВМ приходится наделять специальными функциями, обеспечивающими, например, параллельную обработку нескольких задач. Таким образом, существенное влияние на организацию ЭВМ оказывают не только свойства алгоритмов, но и требования к производительности — времени выполнения алгоритмов. То же самое можно сказать и о влиянии надежности на функциональную организацию ЭВМ.

Влияние элементной базы на принципы построения ЭВМ сводится, в основном, к следующему. Оказывается, что конструкция ЭВМ предельно упрощается и ЭВМ работает наиболее надежно (устойчиво), если сигналы, циркулирующие в электронных схемах ЭВМ, используются для представления только двух значений — 0 и 1. Таким образом, свойства электронных элементов заставляют представлять информацию, с которой оперирует ЭВМ, исключительно в двоично-кодированной форме — в виде последовательностей из нулей и единиц. Столь же существенное влияние

на принципы построения ЭВМ оказывает специфика средств, используемых для организации машинной памяти, а также для ввода в ЭВМ и вывода информации.

1.2. Принцип программного управления

Современные ЭВМ строятся на одном принципе — *принципе программного управления*. В основе этого принципа лежит представление алгоритма в форме операторной схемы, которая задает правило вычислений как композицию операторов (операций над информацией) двух типов: операторов, обеспечивающих преобразование информации, и операторов, анализирующих информацию с целью определения порядка выполнения операторов. Принцип программного управления может быть реализован в ЭВМ многими способами. Один из способов реализации программного управления был предложен в 1945 г. Дж. фон Нейманом [12], и с тех пор неймановский принцип программного управления используется в качестве основного принципа построения всех современных ЭВМ.

Неймановский принцип программного управления. Этот принцип состоит в следующем.

1. Информация кодируется в двоичной форме и разделяется на единицы (элементы) информации, называемые *словами*.

2. Разнотипные слова информации различаются по способу использования, но не способами кодирования.

3. Слова информации размещаются в ячейках памяти машины и идентифицируются номерами ячеек, называемыми *адресами слов*.

4. Алгоритм представляется в форме последовательности управляющих слов, которые определяют наименование операции и слова информации, участвующие в операции, и называются *командами*. Алгоритм, представленный в терминах машинных команд, называется *программой*.

5. Выполнение вычислений, предписанных алгоритмом, сводится к последовательному выполнению команд в порядке, однозначно определяемом программой.

Прокомментируем происхождение и содержание перечисленных пунктов принципа программного управления.

Использование в ЭВМ двоичных кодов продиктовано, как уже отмечалось, спецификой электронных схем, применяемых для передачи, хранения и преобразования информации. Совокупности нулей и единиц (битов информации), используемые для представления отдельных чисел, команд и т. п., рассматриваются как самостоятельные информационные объекты и называются словами. Слово обрабатывается в ЭВМ как одно целое — как машинный элемент информации.

Во втором пункте утверждается, что нет необходимости в средствах, позволяющих различать разнотипные слова информации.

Поэтому все слова, представляющие числа, команды и прочие объекты, выглядят в ЭВМ совершенно одинаково и сами по себе неразличимы. Только порядок использования слов в программе вносит различия в слова. Благодаря такому «однообразию» слов оказывается возможным использовать одни и те же операции для обработки слов различной природы, например для обработки и чисел, и команд. Так что команды программы становятся в такой же степени доступными для отработки, как и числа, а это приводит к интересным возможностям.

Третий пункт принципа программного управления фиксирует специфику хранения и идентификации (обозначения) информации, порождаемую свойствами машинной памяти. Дело в том, что ныне машинная память не имеет ничего общего с памятью, существующей в живой природе. Машинная память — совокупность ячеек, каждая из которых служит местом для хранения слова информации, и наиболее подходящий синоним термина «машинная память» — «склад информации». Ячейка памяти выделяется для хранения значения величины, в частности константы или команды. Чтобы записать слово в память, необходимо указать адрес ячейки, отведенной для хранения соответствующей величины. Чтобы выбрать слово из памяти (прочитать его), следует опять же указать адрес ячейки памяти. Таким образом, адрес ячейки, в которой хранится величина или команда, становится *машинным идентификатором* (именем) величины и команды. Для обозначения величин и команд в ЭВМ нет никаких средств, кроме адресов, присваиваемых величинам и командам в процессе составления программы вычислений. Дополнительно отметим, что выборка (чтение) слова из памяти не разрушает информацию, хранимую в ячейке. Это позволяет любое слово, записанное однажды, читать какое угодно число раз, т. е. из памяти выбираются не слова, а копии слов.

Принцип программного управления работой ЭВМ предполагает, что алгоритм представляется в ЭВМ в виде упорядоченной последовательности команд следующего вида:

$$\underbrace{bb \dots b}_{\text{КО}} \quad \underbrace{bb \dots b}_{A_1} \quad \underbrace{bb \dots b}_{A_2} \quad \dots \quad \underbrace{bb \dots b}_{A_k}.$$

Здесь b — двоичная переменная, принимающая значение 0 или 1, так что команда — это последовательность нулей и единиц. Определенное число первых разрядов команды характеризует *код операции* (КО). *Код* — это любое обозначение, отличное от общепринятого. Так, операция сложения может представляться в команде кодом 001010. Последующие наборы двоичных переменных $bb \dots b$ определяют адреса $A_1, \dots, A_k$ операндов (аргументов и результатов), участвующих в операции, заданной кодом КО.

Чтобы изобразить, из каких частей состоит команда, обычно пользуются следующей формой:

$$\begin{array}{c}
 1 \quad l \quad 1 \quad m \quad 1 \quad m \qquad \qquad \qquad 1 \quad m \\
 \boxed{\begin{array}{|c|c|c|c|c|} \hline \text{КО} & A_1 & A_2 & . \ . \ . & A_k \\ \hline \end{array}}
 \end{array} \quad (1.1)$$

Здесь прямоугольник обозначает слово информации, части которого, имеющие определенный смысл, принято называть *полями*. Так КО, A_1 , A_2 , ..., A_k — поля команды, представляющие соответственно код операции и адреса операндов, участвующих в операции. Сверху указаны номера разрядов полей: поле КО состоит из l двоичных разрядов, каждое поле A_1 , A_2 , ..., A_k содержит m двоичных разрядов. Известно, что с помощью K двоичных разрядов можно закодировать 2^K различных значений или объектов. С учетом этого команда (1.1) позволяет инициировать одну из 2^l операций и каждый адрес может принимать до 2^m различных значений, обеспечивая ссылку на любую из 2^m величин или команд. Форма (1.1) характеризует *структуру*, или иначе *формат команды*. Требуемый порядок вычислений предопределяется алгоритмом и описывается последовательностью команд, образующих программу вычислений.

Процесс вычислений, выполняемых ЭВМ по заданной программе, состоит в последовательном выполнении команд. Первой выполняется команда, заданная пусковым адресом программы. Обычно это адрес первой команды программы. Адрес следующей команды однозначно определяется в процессе выполнения текущей команды и может быть либо адресом следующей по порядку команды, либо адресом любой другой команды. Процесс вычислений продолжается до тех пор, пока не будет выполнена команда, предписывающая прекращение вычислений.

Важно видеть, что вычисления, производимые машиной, определяются программой. Именно программа «настраивает» ЭВМ на получение требуемых результатов. Замена программы приводит к изменению функций, реализуемых ЭВМ. Следовательно, многообразие программ, которые могут быть выполнены ЭВМ, определяет класс функций, который способен реализовать ЭВМ.

Другие принципы программного управления. Неймановский принцип не является единственно возможным принципом функциональной организации ЭВМ. Так, Дж. Айлиф в [1] предлагает принципы построения машины, отличной от традиционных неймановских машин. Отличия сводятся в основном к следующему. Во-первых, неймановский принцип программного управления предполагает, что коды слов информации не зависят от типа информации. Это приводит к тому, что программист сам обязан следить за тем, чтобы для обработки информации определенного типа, например целых или действительных чисел, использовались соответствующие операции, чтобы был запрограммирован перевод

чисел из одной формы представления в другую и т. д. Если эти правила не соблюдаются, то в программе появляются ошибки, приводящие к получению курьезных результатов. Дж. Айлиф предлагает тип информации отображать в кодах данных. В результате этого числа, адреса, команды оказываются различными и операция, указываемая в команде, производится машиной в форме, соответствующей типам операндов. Это приводит к сокращению списка машинных операций и уменьшению числа ошибок в программе. Во-вторых, память неймановской машины сугубо линейна — это последовательность адресов от 0 до L . И какой бы не была структура данных, т. е. из каких бы элементов (скаляров, векторов, матриц) не состояли данные, и как бы они не были взаимосвязаны, программист должен эти данные спроецировать на линейную цепочку адресов 0, 1, ..., L . Затем при составлении программы приходится находить способ выделения адресов, соответствующих отдельным структурным элементам данных. Процедуры размещения информации в памяти и выделения элементов информации весьма сложны. Дж. Айлиф предлагает вносить описание структуры информации непосредственно в память машины, за счет чего обеспечивается возможность автоматического выявления адресов отдельных элементов в процессе работы машины. Благодаря этому значительно упрощается как процесс программирования, так и сами программы, что уменьшает число содержащихся в программах ошибок.

Ясно, что дополнительные возможности ЭВМ обеспечиваются за счет введения в машину дополнительной аппаратуры, что приводит к увеличению стоимости ЭВМ. К настоящему времени попытки определить наиболее рациональные принципы построения ЭВМ обычно заканчиваются созданием машин, построенных на неймановском принципе. Это объясняется тем, что возможности неймановских машин, хотя они и доведены до предела, пока обеспечивают потребности в вычислениях. Однако к настоящему времени сложность задач, для решения которых применяются ЭВМ, достигла такого уровня, что затраты на программирование и решение задач близки к экономическим возможностям общества, а порой, по-видимому, превышают их. В связи с этим уже сейчас ощущается потребность в пересмотре классического неймановского принципа построения ЭВМ с тем, чтобы приблизить машинные формы представления данных и алгоритмов к естественным, повседневно используемым способам представления и обработки информации.

В настоящее время наращивание возможностей машин производится в основном за счет программных средств. Наряду с этим делаются попытки изменить принципы построения самих машин. Так, широкое распространение получил принцип построения ЭВМ с развитыми системами интерпретации [17], разработанный коллективом, возглавляемым академиком В. М. Глушковым. ЭВМ этого класса обеспечивают восприятие алгоритмов, записан-

ных на языках высокого уровня — в виде знаков операций, наименований величин и данных, представляемых в естественной форме, причем указанные возможности реализуются за счет введения в ЭВМ нетрадиционных средств адресации и операций над информацией.

1.3. Состав и порядок функционирования ЭВМ

Теперь определим номенклатуру устройств, из которых должна состоять ЭВМ. Номенклатура устройств органически проистекает из неймановского принципа программного управления и свойств технических средств, обеспечивающих хранение, обработку, ввод и вывод информации.

Основные устройства ЭВМ. Принцип программного управления предполагает, что ЭВМ строится по рис. 1.1. Для хранения инфор-

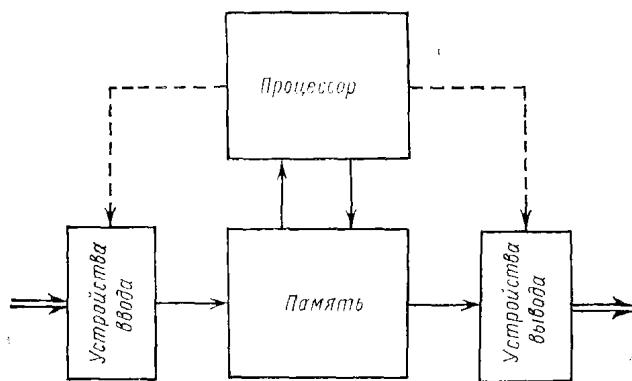


Рис. 1.1. Состав ЭВМ

мации, необходимой для производства вычислений, служит *память*. В памяти размещаются программы, задающие порядок вычислений, и данные, представляющие исходные значения, промежуточные и конечные результаты вычислений. Для ввода информации — программ и данных — в память ЭВМ используются *устройства ввода*, которые обеспечивают считывание информации с определенных носителей (перфокарт, перфолент, клавиатур и т. п.) и представление считанной информации в форме электрических сигналов, воспринимаемых памятью. Вывод информации из памяти для ее последующего использования обеспечивается *устройствами вывода*, которые преобразуют электрические сигналы, поступающие из памяти и несущие в себе информацию о результатах вычислений, в форму печатного текста, пробивок на перфолентах и перфокартах и т. п. Одним словом, устройства ввода и вывода обеспечивают обмен информацией со средой, внешней по отношению к ЭВМ. В отдельных случаях ввод и вывод реализуются одним физическим устройством, например

электрической пишущей машинкой, с клавиатуры которой вводится информация, а на лист бумаги выводится информация. Устройства, обеспечивающие как ввод, так и вывод информации, называются *устройствами ввода—вывода*. Этим же термином пользуются для обозначения любого устройства, относящегося к классу устройств ввода и вывода информации.

Вычисления, заданные программой, реализуются *процессором*. Функцией процессора является выборка команд из памяти и выполнение действий, предписанных командами. Команды выбираются последовательно одна за другой из ячеек памяти, адреса которых определяются во время выполнения предшествующей команды. Процессор выполняет все операции за исключением операций ввода—вывода информации. Команды ввода—вывода, как и любые другие команды, поступают в процессор, но функции процессора в отношении этих команд ограничиваются инициированием операций ввода—вывода, а сами операции реализуются устройствами ввода—вывода.

Процессор — это центральное устройство ЭВМ. Процессор «воспринимает» программу и на ее основе управляет работой всех устройств ЭВМ, иницируя выполнение действий в памяти и устройствах ввода—вывода.

Порядок функционирования ЭВМ. Работа ЭВМ протекает следующим образом. Программа и исходные данные, представленные на машинном носителе информации, например на перфокартах, считываются устройством ввода и загружаются в память ЭВМ. Затем в процессор посылается пусковой адрес программы, и процессор начинает выполнять программу от команды с заданным адресом, что сводится к последовательной выборке команд из памяти и их выполнению средствами процессора и устройств ввода—вывода. Этот процесс заканчивается в момент выборки команды, отмечающей конец вычислений.

Для загрузки и запуска программы необходимо выполнить определенную последовательность действий: указать адрес области памяти, в которую должна быть загружена программа; инициировать операцию ввода в определенном устройстве; передать в процессор пусковой адрес программы и т. д. Для таких действий служит *инженерный пульт управления*, который снабжается органами управления и средствами индикации состояний процессора. Чтобы упростить процесс загрузки и запуска программ, используется специальная программа — *программа начальной загрузки*. После включения ЭВМ под напряжение программа начальной загрузки вводится в память под управлением с инженерного пульта. Загрузка и запуск всех последующих программ производятся уже в автоматическом режиме под управлением программы начальной загрузки, которая задает необходимую последовательность действий, обеспечивающих ввод программы в память и запуск программы. Для инициирования программы начальной загрузки и других вспомогательных программ