Микрокомпьютеры семидесятых

АРГО ЧИПА

В статье рассказывается о принципах работы микропроцессора, структуре и возможностях программирования микро-ЭВМ, областях их применения.

АРГО АЛЬФРЕДОВИЧ ЧИПА (род. в 1933 г. в Рижском районе) окончил физмат ЛГУ в 1957 г. С 1960 г. работает в Институте электроники и вычислительной техники АН ЛатвССР. В 1970 г. защитил диссертацию на соискание ученой степени кандидата технических наук. За разработку и внедрение специализированных компьютеров удостоен Государственной премии Латвийской ССР. Автор 30 научных работ.

ВИДЫ ЗАПИСИ МАТЕМАТИЧЕСКИХ ДЕЙСТВИЙ. В арифметической записи математических действий (например 3+4=7) указаны конкретные числа, над которыми производится действие. В алгебраическом выражении (a+b=c) указано, какие действия и в какой последовательности надо произвести над числами, «скрытыми» за этими буквами.

Задумаемся, в чем принцип записи. Если писать на бумаге, то, исписав страницу, можно взять другую; на доске старую запись можно стереть губкой. И в ЭВМ для записи каждой цифры используется отдельный элемент, число можно вписать в него и стереть в нем. Группа таким образом организованных элементов называется памятью ЭВМ.

Попробуем взглянуть на символы, используемые в записи математического действия, как на указатели места, где находятся данные числа. Поделим доску на квадраты, пронумеровав их, как показано на рис. 1. Номер каждого квадрата (ячейки) — это его адрес. Предположим, что в каждой ячейке можно записать одно и только одно число, которое в случае необходимости стирается. Например, в ячейке с адресом *2* запишем число 4, а в ячейке с адресом *5* — число 3. Теперь к числу в 5-й ячейке прибавим число во 2-й, и результат запишем в ячейку с адресом *11*.

Для сокращенного выражения действия в ЭВМ применяется адресный код, который в упомянутом примере выглядит как 5+2=11 и показывает, какие действия следует выполнить с числами, находящимися по данным адресам, и куда поместить результат. Точно так же 4:7=10 означает «поделить число, находящееся в 4-й ячейке, на число, находящееся по 7-му адресу, а результат записать по 10-му адресу». Цифры, содержащиеся в программе работы машины, — это, как правило, не числа, над которыми надо произвести те или иные действия, а их адреса.

ЭВМ выполняет не только арифметические, но и целый ряд других действий, например, пересылку числа из одного адреса в другой, что обозначается стрелкой: 3→8. Это означает, что число, находящееся по 3-му адресу, мы пересылаем и повторяем по 8-му адресу, теперь в обоих адресах у нас одно и то же число.

Для упрощения операций в ЭВМ используются, кроме того, такие свойства:

— когда мы считываем число с какого-либо адреса, то его значение в этом адресе сохраняется;

— прежде чем записать число в какой-либо адрес, предыдущая величина в данном адресе стирается.

Если попросить читателя написать на листе бумаги арифметическое выражение, допустим, 3+4=7, то он, естественно, спросит: «Ну, и что дальше?». Перед тем же вопросом стоит и машина. В результате мы обобщаем каждую подобную запись действия и, представив ее в адресном виде, трактуем как команду машине. Любая такая команда указывает:

- какое действие должна выполнить машина;
- где взять одно или оба числа, над которыми данное действие производится;
- куда поместить результат;
- что делать после выполнения данной команды, то есть какую команду выполнять вслед за ней.

Как же записать команду, чтобы ее без труда могла «понять» машина? Оказывается, что для ЭВМ целесообразно обозначать действия цифровым способом, например, сложению присвоить код 01, вычитанию — код 02 и т. п. В результате у нас будет столько различных чисел, сколько разных действий машина способна выполнить. Количество подобных действий в микрокомпьютерах достигает ста—двухсот, так как кодируются еще операции по пересылке, логические операции. Но мы примем, что для кодирования достаточно двух цифр.

Взглянем еще раз на рис. 1 и убедимся, что все три числа, участвующих в арифметических действиях, могут быть помещены в любую ячейку. А это, в свою очередь, означает, что мы должны предусмотреть место и зарезервировать возможность в любой момент записать и самый большой адрес. При этом в одной команде адрес надо записать трижды: для обоих слагаемых и суммы. Количество адресов в реальных ЭВМ достигает нескольких миллионов, поэтому для краткой числовой записи приходится прибегать к остроумным техническим средствам. Но мы предположим, что количество адресов невелико и для кодирования любого достаточно трехзначного числа.

В машине, по конструктивным соображениям, знак действия записывается первым, затем следуют все три адреса. Таким путем сложение можно изобразить следующими числовыми комбинациями, получив трехадресную систему:

Код вида действия	Адрес первого слагаемого (числа)	Адрес второго слагаемого (числа)	Адрес суммы (результата)
01	005	002	011

Знак арифметического действия и знак равенства отделяют одно число от другого. В ЭВМ для этой цели любой адрес всегда записывается одинаковым количеством знаков, а отсутствующие позиции перед числом заполняются нулями. Таким образом, в нашем примере код действия всегда будет двузначным, адреса — трехзначным. Следовательно, все цифры можно записать одним большим числом, не создавая этим никаких неясностей: 01005002011. При других действиях и адресах поменяются отдельные цифры этого числа, но не количество знаков и не смысл отдельных цифровых групп.

1	4 2	9 3	4
3 5	6	7	9 8
9	10	7 11	12

Тем самым любая машинная команда превращается в число и, по техническим причинам, хранится в той же памяти, что и сами числа. В свой черед, это означает, что должна иметься возможность записать и хранить в каждой ячейке одиннадцатизначные числа! Какой же длины должно быть такое число в настоящей ЭВМ, где происходит адресация нескольких миллионов ячеек?! Поэтому в реальном компьютере длину такого числа сокращают... Вне нашей доски поместим еще одну, маленькую досочку, или специальную ячейку, в которую можно записать только одно число. В нее с помощью операции пересылки до выполнения арифметического действия всегда будем помещать первое число (см. таблицу), а после выполнения действия туда же запишем результат, предварительно стерев прежнее содержимое ячейки. Теперь отпадает необходимость указывать

в каждой команде адрес первого числа и результата, эти адреса вообще можно исключить из команды, существенно уменьшив ее длину.

Истины ради следует, правда, сказать, что в машине применяется двоичная (бинарная) система счисления, то есть все числа изображаются комбинациями цифр 1 и 0, поскольку их легко реализовать средствами схемы. Однако структуры команды это не меняет, подобно тому как не меняется сумма в 30 рублей, выданная трешками, или пятерками, или десятками.

Выполнив данную команду, машина приступает к выполнению той, которая находится в следующей по порядку ячейке, — ее адрес больше прежнего на единицу. Чтобы машина не сбилась с последовательности, вводится специальный узел — счетчик команд, указывающий адрес (номер) выполняемой команды. Адрес следующей команды получают после выполнения предыдущей прибавлением единицы в этом счетчике.

СТРУКТУРА МИКРО-ЭВМ. Для механического выполнения арифметических действий необходима определенная аппаратура, составляющая основные узлы ЭВМ (см. рис. 2): микропроцессор, блок памяти и узел управления вводом — выводом.

МП в этой системе представляет собой центральный узел управления и обработки информации, размещенный на одном или нескольких кристаллах больших интегральных схем. В выработке множества управляющих сигналов особую роль играют два блока:

— счетчик команд, о котором уже шла речь;

— регистр команд, в который вводится из памяти код команды, то есть содержимое той ячейки, чей адрес указывает в данный момент счетчик команд.

Обычно МП содержит один лишь сумматор, выполняющий сложение и вычитание. Деление, умножение и другие действия сводятся с помощью программы к многократному сложению или вычитанию.

Для сокращения длины команд сумматор соединен с аккумулятором — специальной ячейкой, в которой хранится одно слагаемое и куда после окончания действия помещается результат.

Кроме того, выполнение каждого арифметического действия сопровождается логической оценкой его результата. Эта оценка в виде ответа да/нет помещается в особую ячейку — регистр флажков и может указывать дальнейшее направление выполнения программы, упрощая ее составление.

Память состоит из множества ячеек, в каждой из которых хранится одно машинное слово, то есть многозначное число. Все ячейки памяти имеют сквозную нумерацию. Если на узел памяти по одной группе выводов подается адрес, то по второй группе выводов можно считывать содержимое указанной ячейки или же, в зависимости от управляющего сигнала, записать в нее новую информацию.

Узел управления вводом—выводом осуществляет связь МП с внешней средой, и с помощью программирования эту большую интегральную схему можно приспособить для обмена данными с любым другим устройством, имеющимся в распоряжении потребителя. В этом случае код операции в команде будет указывать, надо ли вводить числа в машину или выводить из нее. В свою очередь, поле адреса команды указывает номер (адрес) того устройства, которое в данный момент должно участвовать в операции ввода либо вывода. Тем самым один МП может обслуживать целый ряд различных устройств.

Введенная числовая информация всегда попадает в аккумулятор микро-ЭВМ, и о ее дальнейшем использовании должен заботиться тот, кто пользуется системой.

ЗА ОДНУ СЕКУНДУ МП выполняет от ста тысяч до миллиона команд. Время выполнения каждой из них можно подразделить на три последовательных этапа:

▲ помещение команды в регистр команд;

▲ формирование необходимых для ее выполнения управляющих сигналов;

▲ выполнение операций, требуемых командой.

Прежде всего в счетчик команд попадает номер выполняемой команды. Затем на первом этапе содержимое счетчика подается в память в виде адреса, по которому хранится команда. В памяти по указанному адресу разыскивается нужная ячейка, и ее содержимое подается на вывод данных. Блок управления направляет это число в регистр команд МП (см. рис. 2).

На втором этапе каждая из рассмотренных выше частей команды обрабатывается отдельно. Код операции определяет последовательность управляющих сигналов, а также настраивает МП для выполнения заданного действия, например сложения. Поле адреса команды подается в виде адреса в память или узел управления вводом—выводом и определяет номер той ячейки памяти или того устройства ввода—вывода, которое в дальнейшем будет участвовать в выполнении предписанного действия. Одновременно управляющие сигналы определяют и направление потока данных, как показано стрелками на рис. 2.

На третьем этапе МП выполняет указанное в коде операции действие, например сумматор складывает содержимое аккумулятора и указанной ячейки памяти, помещая результат в аккумулятор. Затем происходит логическая оценка результата операции и установка соответствующих флажков. Флажок показывает, будет ли программа продолжена с адресом, который указан в счетчике команд, или же она будет продолжена с тем адресом, который указан в поле адреса регистра команд. Это свойство позволяет осуществить ветвление программы и, в зависимости от полученного результата, управлять процессом вычисления. На том же этапе в счетчике команд прибавляется единица, и после выполнения операции МП готов выполнить следующую команду — и так до тех пор, пока не получит «приказ» остановиться.

ПРИ ПОЛЬЗОВАНИИ МИКРО-ЭВМ потребитель соприкасается только с выводами узла управления вводом—выводом, которые с помощью программирования можно приспособить к обслуживанию любого устройства ввода или вывода. Выводы этого узла взаимодействуют с устройствами, находящимися в распоряжении потребителя. Любой из выводов интегральной схемы можно запрограммировать как ввод или вывод микро-ЭВМ, а также на любой из выводов в нужное время подать напряжение либо отключить его. С этой точки зрения микро-ЭВМ можно рассматривать как схему (рис. 3), которая в нужный момент по программе включает или выключает выводы, в то время как вводы этой схемы служат ключами, с помощью которых можно менять сами выполняемые программы. Причем число выводов такой схемы нетрудно увеличить, использовав несколько больших интегральных схем узлов управления вводом—выводом.

Но еще существеннее то, что микро-ЭВМ, построенная, как показано на рис. 2, может использоваться для управления самыми различными устройствами — меняется только программа ее работы. Поэтому сегодня начато промышленное производство таких небольших микро-ЭВМ. Возможности их вполне достаточны для организации небольшой системы управления.