Глава 1. Представление информации в ЭВМ

Курс лекций «ЭВМ и периферийные устройства»

Лектор - доцент, к.т.н.

Кузьмин Александр Александрович

Исходные положения

Суть обработки информации в ЭВМ – взаимодействие «потока» исходных данных (операндов) и «потока» команд (программы) с целью получения «потока» результатов.

Взаимодействие операндов и команд в ЭВМ предполагает их хранение, выборку на исполнение, собственно выполнение операций взаимодействия и сохранение результатов (промежуточных и итоговых). Хранение и выборка информации предполагает наличие описания способа обращения к операндам и командам, и, как следствие, применение «адресной информации» (кодов адресов, признаков поиска, дескрипторов и др.)

Собственно ЭВМ, её устройства и подсистемы, а также режимы их работы подлежат идентификации и, как следствие, необходимо представление информации о состояниях (о кодах состояний) устройств и режимов их работы.

Адресная информация и коды состояний образуют внутренний поток операндов.

Следствие: наличие «машинной» специфики в типах и форматах операндов, а также в структуре, типах и форматах команд.

Типы и форматы операндов

Основные типы операндов (информационных единиц):

числа; символы (коды символов); логические данные.

Дополнительные типы операндов:

битовые поля; битовые и байтовые строки; строки символов (текстовые строки); адреса (коды адресов) и дескрипторы-указатели (теги, префиксы и др.); коды состояний и коды команд.

Производные типы операндов (информационных единиц):

массивы кодов (всяких) и файлы графических изображений, аудио- и видеоинформации.

Все типы операндов в ЭВМ существуют в конкретной форме – в стандартном формате или в нестандартной форме.

Числа (числовая информация)

Стандартные формы (форматы) чисел:

- фиксированная запятая - «естественная» форма
  - а) целые
    - с различными основаниями (q=2 и, редко, q=8, q=16);
    - знаковые и беззнаковые (неупакованные и упакованные);
    - двоично-десятичные (упакованные и зонные) в разных кодах;
  - б) дробные;
  - в) смешанные
- плавающая запятая – «нормальная» или полулогарифмическая форма
  - с различными основаниями представления порядка (q=2, q=8, q=16);
  - со смещенным порядком;
  - упакованные
- логические значения.

Числовая информация (продолжение)

Нестандартные (нетрадиционные) формы чисел:

- недвоичные системы счисления (например, троичная);
- числа с переменным основанием (например, с применением чисел Фибоначчи) и с иррациональными основаниями;
- непозиционные системы счисления (например, представление чисел в системе остаточных классов);
- числа в логарифмической форме;
- «инверсная запятая»;
- «трансформирующаяся запятая» и другие.

Форматы с фиксированной запятой

Основными в современных ЭВМ являются целочисленные форматы со знаком и без знака, кратные байту: 16 разрядов – слово, 32 разряда – двойное слово, 64 разряда – четверное слово.

Смешанный и дробный форматы в настоящее время почти не применяются (применяются крайне редко).

В последние годы почти перестали применять модифицированную форму представления чисел со знаком, а также запись отрицательных чисел в обратном коде (используется дополнительный код).

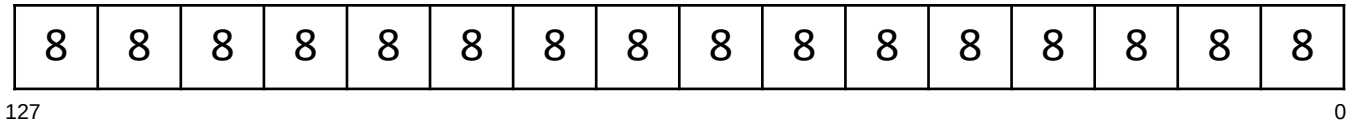
Беззнаковый формат предполагает, что все числа-операнды суть положительные (не могут быть отрицательными). При выполнении операций с числами в беззнаковом формате следует уделить особое внимание признакам переполнения разрядной сетки.

Форматы с фиксированной запятой (продолжение)

Применение целых чисел, представленных **в упакованном виде**, связано с обработкой мультимедийной информации. Команды обрабатывают все эти числа параллельно. Современная единая технология, известна под аббревиатурой SSE (Streaming SIMD Extensions).

В стандарте SSE4 за основу принято 128-разрядное слово и предусмотрены **четыре формата упакованных целых чисел**. Байты в формате упакованных байтов нумеруются от 0 до 15, причем байт 0 располагается в младших разрядах 128-разрядного слова. Аналогичная система нумерации и размещения упакованных чисел применяется для 16-разрядных (номера 0–7), 32-разрядных (номера 0–3) и 64-разрядных (номера 0–1) целых чисел.

Упакованные байты (16 по 8 бит)



Упакованные целые 16-ти разрядные числа

Упакованные целые 32-х разрядные числа

Упакованные целые 64-х разрядные числа

Форматы с фиксированной запятой (продолжение)

Применение целых **двоично-десятичных чисел** связано, прежде всего, с задачами бухгалтерского учета и статистики. Для этих целей используется принцип кодирования каждой десятичной цифры эквивалентным двоичным числом из четырех битов (тетрадой), то есть так называемым двоично-десятичным кодом (BCD — Binary Coded Decimal).

Обычно используется стандартная кодировка 8421, где цифры в обозначении кодировки означают веса разрядов. Оставшиеся шесть комбинаций могут использоваться для представления знаков, а также возможных служебных символов. Знак «+» принято обозначать как 1100 (Ch), а знак «–» — как 1101 (Dh). Такое соглашение принято исходя из того, что обозначение соответствующих шестнадцатеричных цифр можно рассматривать как аббревиатуру бухгалтерских терминов «кредит» (Credit) и «дебет» (Debit).

Другие допустимые обозначения знаков — это 1010 или 1110 для «+» и 1011 для «–». Иногда допускается представление десятичных чисел без знака, и тогда в позиции, отводимой под знак, записывается комбинация 1111.

Существуют и другие задачи, в которых применение двоично-десятичных кодов может быть эффективным. Некоторые применяемые двоично-десятичные коды приведены на следующем слайде.

Десятич-ные цифры	Код «8,4,2,1»	Код «2,4,2,1»	Код «+3»	Код «7,4, 2,1»	Код «5,4,2,1»	Код ω,x,y,z	Код ABCD	Рефлек-сный код	Код «2 из 5»	Двоично-пятиричный код
0	0000	0000	0011	0000	0000	0000	0000	0000	11000	0100001
1	0001	0001	0100	0001	0001	0001	0110	0001	00011	0100010
2	0010	0010	0101	0010	0010	0011	0011	0011	00101	0100100
3	0011	0011	0110	0011	0011	1000	0111	0010	00110	0101000
4	0100	0100	0111	0100	0100	0110	1111	0110	01001	0110000
5	0101	1011	1000	0101	1000	1111	0101	0111	01010	1000001
6	0110	1100	1001	0110	1001	1001	1101	0101	01100	1000010
7	0111	1101	1010	1000	1010	0010	1001	0100	10001	1000100
8	1000	1110	1011	1001	1011	1100	1100	1100	10010	1001000
9	1001	1111	1100	1010	1100	0100	1010	1000	10100	1010000

Форматы с фиксированной запятой (продолжение)

В вычислительных машинах нашли применение два формата представления десятичных чисел (все числа рассматриваются как целые): **упакованный** (а) и **зонный** (б). В обоих форматах каждая десятичная цифра представляется двоичной тетрадой, то есть заменяется двоично-десятичным кодом.

Байт		Байт		. . .	Байт		Байт	
Цифра	Цифра	Цифра	Цифра		Цифра	Цифра	Цифра	Знак

а

Байт		Байт		. . .	Байт		Байт	
Зона	Цифра	Зона	Цифра		Зона	Цифра	Знак	Цифра

б

Примеры: число - 6719 в упакованном формате: 0000 0110 0111 0001 1001 1101, где последняя тетрада 1101 есть знак «-»; число +3651 в зонном формате: 1111 0011 1111 0110 1111 0101 1100 0001, где зоны обозначены кодами 1111, а предпоследняя тетрада 1100 есть знак «+».

Форматы с плавающей запятой

Наиболее существенным недостатком использования форматов чисел с фиксированной запятой является ограниченность диапазона представления чисел в разрядной сетке, следствием чего является возможность переполнения разрядной сетки и необходимость применения процедуры масштабирования.

От этого недостатка свободна «нормальная» форма представления чисел: число разбивается на две группы цифр, первая из которых называется мантиссой, вторая — порядком. Число A представляется формулой:

$$A = \pm M_A \cdot d^{\pm p_a},$$

где M_A – модуль мантиссы (остальное в формуле – характеристика); p_a – модуль порядка числа A; d – основание характеристики.

Из формулы следует, что при изменении модуля мантиссы должен изменяться модуль порядка, что трактуется как «плавание» запятой. Отсюда наименование формы. Диапазон и точность представления чисел с плавающей запятой (ПЗ) зависят от числа разрядов, отводимых под порядок и мантиссу.

Вариант разрядной сетки чисел ПЗ:

Знак мантиссы	Знак порядка	p_{k-1}	. . .	p_1	p_0	$m_{=1}$	m_{-2}	m_{-3}	. . .	m_{-n}

Форматы с плавающей запятой (продолжение)

Диапазон и точность представления чисел с ПЗ зависят от числа разрядов, отводимых под порядок и мантиссу. Диапазон представления чисел зависит и от основания характеристики d, которое может быть отличным от 2. В реальных ныне редко используемых компьютерах d может быть равно 8 или 16. Например, в некоторых универсальных ЭВМ фирмы IBM используется d=16.

В современных ЭВМ для упрощения операций над порядками последние приводят к целым положительным числам, применяя так называемый ***смещенный порядок***. Для этого к истинному порядку добавляется целое положительное число — смещение, которое выбирается равным половине представимого диапазона порядков. Смещенный порядок занимает все биты поля порядка, в том числе и тот, который ранее предназначался для записи знака порядка.

Мантисса в числах с ПЗ представляется в ***нормализованной*** форме. Это означает, что на мантиссу налагаются такие условия, чтобы она по модулю была меньше единицы ($|q| < 1$), а первая цифра после запятой должна быть значимой.

Числа с ПЗ в разных компьютерах могут различаться форматами. Например, в семействах ЭВМ фирмы IBM и фирмы DEC (компьютеры VAX) совпадает длина (разрядность) чисел: 32 разряда одинарный и 64 разряда двойной формат, но в компьютерах IBM порядок занимает 7 бит, используется d=16 и длина мантиссы 24 (одинарный) или 56 (двойной) бит, а в компьютерах DEC порядок 8-ми разрядный, d=2, а длина мантиссы 23 или 55 бит.

Форматы с плавающей запятой (продолжение)

В настоящее время для всех ЭВМ рекомендован **стандарт IEEE 754**, разработанный международным центром стандартизации IEEE (Institute of Electrical and Electronics Engineers). Он был разработан с целью облегчить перенос программ с одного процессора на другие и нашел широкое применение практически во всех процессорах и арифметических сопроцессорах.

Стандарт IEEE 754 определяет 32-битовый (одинарный) и 64-битовый (двойной) форматы с 8- и 11-разрядным порядком, соответственно. Основанием характеристики является 2. В дополнение, стандарт предусматривает два расширенных формата (одинарный и двойной), фактический вид которых зависит от конкретной реализации. Расширенные форматы предусматривают дополнительные биты для порядка (увеличенный диапазон) и мантиссы (повышенная точность).

Для технологии SSE4 (применение команд, служащих для увеличения производительности систем при обработке мультимедийной информации) используются форматы упакованных чисел с ПЗ. Каждая такая команда работает с четырьмя операндами с ПЗ одинарной точности или двумя операндами двойной точности. Операнды упаковываются в 128-разрядные группы.

Размещение числовой информации в памяти

В настоящее время разрядность ячейки памяти компьютера, как правило, равна одному байту, а реальная длина кодов чисел составляет 1,2, 4, 8 или 16 байтов. При хранении таких чисел в памяти их байты размещают в нескольких ячейках со смежными адресами, при этом для доступа к числу указывается только наименьший из адресов.

При разработке архитектуры системы команд следует определить порядок размещения байтов в памяти, то есть какому из байтов (старшему или младшему) будет соответствовать этот наименьший адрес. Например, фирмы DEC и Intel предпочитают размещать в первой ячейке младший байт, а IBM и Motorola ориентируются на противоположный вариант. В последнее десятилетие в большинстве ЭВМ предусматривается использование обоих вариантов. (Выбор может быть произведен программным путем за счет соответствующей установки регистра конфигурации).

Представляется почти очевидным, что порядок размещения байтов должен быть увязан с выбором адреса, с которого начинается запись числа, что в свою очередь следует увязать с современной с физической реализацией полупроводниковых запоминающих устройств, где обычно предусматривается возможность считывания (записи) четырех байтов подряд.

Размещение числовой информации в памяти (продолжение)

Операции считывания и записи выполняются быстрее, если адрес первого байта числа A отвечает условию: (A) mod S = 0, где S = 2, 4, 8 или 16.

Числа, размещенные в памяти в соответствии с этим правилом, называются выравненными:

Современные компиляторы персональных компьютеров генерируют коды, в которых предусмотрено выравнивание чисел в памяти.

Символьная информация

Символьную информацию можно определить как совокупность кодов, каждый из которых поставлен в соответствие символу. Множество символов образуют подмножества:

- буквы разных алфавитов;
- цифры (не числа – не путать!);
- математические знаки и операторы;
- диакритические и фонетические знаки;
- типографские знаки и знаки пунктуации;
- общеупотребительные обозначения (например, денежных единиц) и др.

В подавляющем большинстве случаев в различных ***таблицах кодировок*** применяются двоичные коды. При этом часто соблюдаются некоторые общепринятые традиции и принципы, например, «весовой» принцип.

На первый взгляд кажется, что обработку символьной информации (манипуляцию с кодами символов) нельзя отнести к вычислениям, а следовательно она как бы противоречит исходной терминологии и понятиям «вычислительная машина» и «компьютер». Однако, эта обработка полностью отвечает ранее введенным определениям ЭВМ (см. Введения).

Символьная информация (продолжение)

В течении продолжительного времени (до 1993 года) применялись кодовые таблицы, в которых символы кодируются с помощью восьмиразрядных двоичных комбинаций (байтов), позволяющих представить 256 различных символов, среди которых наиболее употребительными являются:

- американский стандартный код для обмена информацией ASCII (American Standard Code for Information Interchange);
- разработанный корпорацией IBM расширенный двоично-кодированный код EBCDIC (Extended Binary Coded Decimal Interchange Code) и его российский аналог ДКОИ-8 (добавлены буквы кириллицы).

Стандартный код ASCII - 7-разрядный, восьмая позиция отводится для бита четности. Европейская модификация ASCII, называемая Latin 1 (стандарт ISO 8859-1) В ней используются все 8 разрядов. «Старшие» комбинации (коды 128–255) отводятся для представления специфических букв алфавитов западноевропейских языков, символов псевдографики, некоторых букв греческого алфавита, а также ряда математических и финансовых символов. Именно эта кодовая таблица считается мировым стандартом де-факто, который применяется с различными модификациями во всех странах.

Код ASCII достаточно удобен, но он все же не вмещает множества необходимых символов.

Символьная информация (продолжение)

В 1993 году консорциумом компаний Apple, Microsoft, Hewlett-Packard, DEC и IBM был разработан 16-битовый стандарт ISO 10646, определяющий универсальный набор символов (UCS, Universal Character Set). Новый код, известный под названием **Unicode**, позволяет задать до 65 536 символов, то есть дает возможность одновременно представить всё расширяющееся множество знаков и символы всех основных «живых» и «мертвых» языков. Для букв русского языка выделены коды 1040–1103.

Блоки символов в стандарте Unicode:

Коды	Символы
0 - 8191	Алфавиты — английский, европейские, фонетический, кириллица, армянский, иврит, арабский, эфиопский, бенгали, деванагари, гур, гуджарати, ория, телугу, тамильский, каннада, малайский, сингальский, грузинский, тибетский, тайский, лаосский, кхмерский, монгольский
8192–12287	Знаки пунктуации, математические операторы, технические символы, орнаменты и т. п.
12288–16383	Фонетические символы китайского, корейского и японского языков
16384–59391	Китайские, корейские, японские идеографы. Единый набор символов каллиграфии хань
59392–65024	Блок для частного использования
65025–65536	Блок обеспечения совместимости с программным обеспечением

На перспективу предусмотрена возможность расширения кода до 21 бита (разделение кодов на 17 «плоскостей», две последние из которых предназначены «для частного использования»). В настоящее используется не более 10% потенциального пространства кодов.

Логические данные

Элементами логических данных является логические (булевы) аргументы и переменные, которые могут принимать лишь два значения: «истина» или «ложь» («да» и «нет»). Кодирование логического значений принято осуществлять битом информации: единицей кодируют истинное значение, нулем — ложное.

В современных компьютерах оперируют наборами логических переменных длиной в машинное слово, либо используют специальные операционные устройства, именуемые «битовыми процессорами», входящими в состав основного процессора. Обрабатываются отдельные биты или слова с помощью команд логических операций (И, ИЛИ, НЕ, Исключающее ИЛИ и, редко, другими), при этом все биты многоразрядного слова обрабатываются одинаково, но независимо друг от друга (переносов между разрядами нет).

Дополнительные типы данных

Битовые поля (БП) - некоторое количество семантически связанных между собой бит, расположенных последовательно в памяти. Если разрядность битового поля не кратна размеру слова и/или битовое поле не выровнено по границе слова, для получения значения битового поля требуются дополнительные команды процессора: операция «битовое И» (используется для выбора бит битового поля по маске); логический сдвиг вправо (используется для сдвига бит битового поля в младшие разряды слова).

Недостаток использования БП: дополнительные команды замедляют выполнение кода. Достоинство: при использовании битовых полей достигается максимально плотная упаковка информации.

Строки – это операнды, образованные непрерывной последовательностью битов, байтов, слов или двойных слов. ***Битовая строка*** может начинаться в любой позиции байта и содержать любое количество битов (вплоть до полного размера оперативной памяти ЭВМ). ***Байтовая строка*** может состоять из байтов, слов или двойных слов.

Если в байтовой строке содержатся коды символов, то говорят о ***текстовой строке***. Для указания конца строки в последний байт заносится код-ограничитель — обычно это нули во всех разрядах байта. Иногда вместо ограничителя длину строки указывают числом (тэгом-указателем или префиксом), расположенным в первом или в двух первых байтах строки.

Дополнительные типы данных (продолжение)

Указатели (теги, дескрипторы, префиксы и т.п.) и **коды состояний** устройств или режимов работы нельзя отнести ни к числовой информации, ни к символьной. Эти типы данных суть кодовые комбинации используются и/или образуются в процессе работы ЭВМ и выполняют служебную функцию диспетчеризации процесса обработки основной информации.

Так, например, дескрипторы в общем случае определяются, как лексические единицы (слова, словосочетания) информационно-поискового языка, служащие для описания основного смыслового содержания документа или формулировки запроса при поиске документа (информации) в информационно-поисковой системе, а также в узком значении как служебные структуры данных.

Коды состояний вместе с признаками (флагами) результата используются в командах передачи управления и в простых программах, и в многозадачных вычислительных системах.

По принципам обработки эти виды информации имеют много общего с обработкой логических и символьных данных, но не тождественны им.

Производные типы операндов и иные виды информации

Видеоинформация бывает как статической (подобно числовой или символьной), так и динамической. Статическая видеоинформация включает в себя текст, рисунки, графики, чертежи, таблицы и др. Динамическая видеоинформация используется либо для передачи движущихся изображений (анимация), либо для последовательной демонстрации отдельных кадров (слайд-фильмы).

Существует ***два способа*** представления графических изображений: ***матричный*** (растровый) и ***векторный***. В матричных форматах изображение представляется прямоугольной матрицей точек — пикселов (**pic**ture **el**ement), положение которых в матрице соответствует координатам точек на экране. Помимо координат, каждый пиксел характеризуется своим цветом, цветом фона или градацией яркости.

Основной недостаток матричной (растровой) графики заключается в большой емкости памяти, требуемой для хранения изображения, из-за чего прибегают к различным методам сжатия данных. В настоящее время существует множество форматов графических файлов, различающихся алгоритмами сжатия и способами представления матричных изображений, а также сферой применения: JPEG (Joint Photographic Experts Group), GIF (Graphics Interchange Format), PCX (PC Paintbrush File Format) и др.

Векторное представление задает изображение графическими примитивами, которые могут быть описаны математически: окружности и эллипсы, ломаные линии, сплайны, безигоны и др.

Так как все примитивы в векторной графике определены соответствующими математическими выражениями, для их описания в рамках изображения достаточно указать лишь несколько параметров, а соответствующее графическое представление получить путем вычислений. Недостатком векторных изображений является их некоторая искусственность, заключающаяся в том, что любое изображение необходимо разбить на конечное множество составляющих его примитивов.

Существует несколько форматов графических векторных файлов: CDR (Corel Drawing), VSD (Microsoft Visio format), PS (PostScript) и др.

Графические форматы, позволяющие сочетать матричное и векторное описание изображения, называются метафайлами. Существует несколько типов форматов метафайлов.

Аудиоинформация. Чтобы быть представленной в ЭВМ звуковая информация должна быть преобразована в цифровую форму. Для качественного представления аудиоинформации необходимо 16-разрядное представление амплитуды сигнала (2^{16} градаций уровня звука) и частота выборки порядка 40 кГц (промежуток времени между последовательными выборками не более 25 мкс).

Цифровой эквивалент аудио сигналов обычно хранится в виде файлов, причем широко используются различные методы сжатия такой информации. Существует целый ряд форматов хранения аудиоинформации: MIDI (Musical Instrument Digital Interface), WAV (WAVeform Extension), AVI (Audio Video Interleave) и др.

Теговая организация обращения к данным

В различных архитектурах ЭВМ типы данных определяются программой, иначе говоря, типы данных определяются кодом операции, что приводит к необходимости использовать разные команды для обработки различных типов данных. Каждый тип данных предусматривает наличие специального подмножества команд.

Принцип теговой организации обращения к данным заключается в ***самоопределение данных***, то есть каждая ячейка памяти имеет специальное дополнительное поле (тег), которое служит для описания атрибутов его содержимого. Это позволяет получить инвариантность команд к типам используемых данных. Сокращается число и длина команд. Определение типа данных откладывается с момента выборки команды на момент выборки этих данных. Вероятно возникновение исключительных ситуаций, связанных с несоответствием типов командам, что как правило является ошибкой программирования.

Расширенный принцип тегирования заключается в задании не только типа хранимых данных, но и длины операнда, указателя занятости ячейки и т.д. Используются дополнительные флаги: бит занятости (определяет, свободна ячейка или занята), бит захвата (необходимость выполнения прерывания при доступе к ячейке), биты защиты (определяют, какие обращения допустимы к этой «порции» данных - только чтение, только запись, чтение/запись), поле длины данных.

Преимущества механизма тегирования:

1. Компактная система команд.

2. Строгий контроль типов данных.

3. Более простые компиляторы. Меньший объем программного кода.

4. Уменьшение интенсивности пересылки между основной памятью и процессором, т.к. уменьшается код программы, загружающий интерфейс на 50%, а увеличивающийся объем читаемых данных, загружает интерфейс на 30%.

Недостатки теговой организации:

1. Строгий контроль типов снижает выразительные способности языка.

2. Уменьшается быстродействие процессора за счет откладывания привязки атрибутов данных на этап выполнения команды. В момент дешифрации команды сразу определяются типы данных, здесь же в момент дешифрации команды типы данных не определяются и поэтому то время, которое связано с подгрузкой операнда могло бы быть использовано для определения типов данных, что и происходит. Т.е. сокращено время дешифрации команды, но доступ к памяти медленный и снижается быстродействие процессора.

3. Дополнительные расходы основной памяти в области данных для хранения полей тега.

Структура и формат команды

Командой (инструкцией) называется некоторым образом кодированная информация, определяющая выработку в ЭВМ последовательностей сигналов, предназначенных для выполнения определенной операции (действия) машины над заданными числовыми и нечисловыми кодами.

Конечный вид команды – цифровой код.

В соответствии с введенным определением команды, её структура имеет операционную и адресную части, а также служебную (модифицирующую) часть, определяющую особенность выполнения данной команды (модификацию) в зависимости от значений тех или иных признаков.

Операционная часть (код операции)	Поля модификации	Адресная часть

Адрес (адреса) - содержимое адресной части команды - в наиболее распространенном случае, указывает номер ячейки ОЗУ, в которой записано (хранится) участвующее в операции число (точнее его код) – слагаемое, множимое, делимое и т.д., либо код нечисловой информации (адрес, строка символов, состояние устройства и т.д.). Обобщённо и для простоты обычно говорят: в ЗУ хранятся операнды.

Типы команд

Существуют основные типы операций, которые включаются практически во все современные системы команд компьютеров:

- команды пересылки данных;
- команды арифметической и логической обработки, сдвиги;
- команды преобразования;
- команды ввода/вывода;
- команды управления потоком команд (передачи управления в программах);
- команды управления системой (режимами системы и её отдельных устройств);
- команды работы со строками;
- команды SIMD (групповые команды);
- специальные (служебные) команды.

Адресность

Сколько адресов должно быть в команде?

Если говорить о команде как о предписании, то наиболее полная его форма должна указывать не только адреса всех чисел, участвующих в операции, но и адрес ячейки, в которую следует поместить результат, а также адрес источника следующей команды. Это перечисление подводит к необходимости иметь в команде четыре адреса.

Эволюция ЭВМ сопровождалась эволюцией адресности.

Адрес источника следующей команды может отсутствовать, поскольку программу (последовательность команд) наиболее естественно и целесообразно размещать в некотором массиве рядом расположенных ячеек ЗУ. Если команды записаны последовательно, то весьма просто реализовать смену адреса ячейки-источника очередной команды аппаратно счетчиком адресов. Такой порядок выборки называется ***естественным***, тогда как при задании адреса команды в команде порядок выборки именуется ***принудительным***.

Важно отметить, что использование естественного порядка выборки команд приводит к необходимости введения специальных команд. Эти команды должны изменять порядок вычислений путём изменения содержимого счетчика номеров (адресов) команд. Это необходимо для обеспечения алгоритмической вычислимости. При принудительном порядке выборки специальные команды не нужны, т.к. каждая команда «меняет» порядок выборки.

В самом обобщённом виде: число адресов варьируется от 4 до 0 в универсальных ЭВМ, а в специализированных число адресов может быть практически любым.

Модифицирующая часть команды

Цель использования этой части команды – обеспечение универсальности и унификации форматов команд.

Поле (поля) модификации может располагаться в разных частях многоразрядной команды, а может и вообще отсутствовать. Коды, размещаемые в этих полях, могут быть использованы:

- для задания типов операций;
- для задания порядка использования адресов;
- для задания способа адресации;
- в качестве префикса;
- для указания размеров операндов;
- для хранения атрибутов обращения к памяти и её защиты и др.

Цикл исполнения команды

В основной цикл, реализуемый при выполнении любой команды, входят следующие этапы:

- выборка команды;
- формирование адреса следующей команды;
- декодирование команды;
- вычисление адресов операндов;
- выборка операндов;
- исполнение операции;
- формирование признаков результата;
- запись результата.

Не все из этапов присутствуют при выполнении любой команды (зависит от типа команды), однако этапы выборки, декодирования, формирования адреса следующей команды и исполнения операции имеют место всегда. В определенных ситуациях возможны еще два этапа:

- косвенная адресация;
- реакция на прерывание.

Для ускорения обработки команд используется **конвейерный способ** выполнения этапов. Существуют приёмы разбиения «длинных» этапов на фазы.