

Типы и форматы операндов

Машинные команды оперируют данными, которые в этом случае принято называть *операндами*.

Базовые типы операндов:

- адреса;
- числа;
- символы;
- логические данные.

ВМ обеспечивает обработку и более сложных информационных единиц: графических изображений, аудио-, видео- и анимационной информации. Такая информация является производной от базовых типов данных и хранится в виде файлов на внешних запоминающих устройствах.



Числовая информация

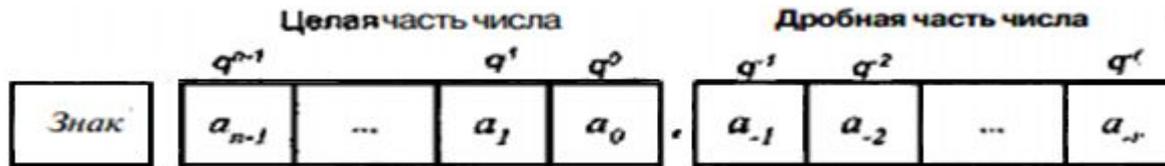
В ЭВМ применяют две формы представления чисел: *с фиксированной запятой (точкой)* и *с плавающей запятой (точкой)*.

Эти формы называют также соответственно *естественной* и *полулогарифмической*.

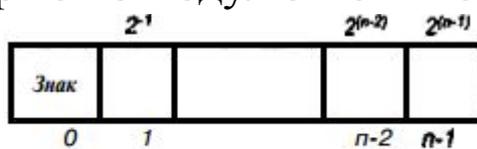


Числа в форме с фиксированной запятой

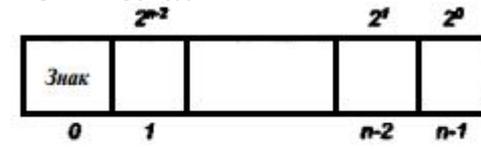
При представлении чисел с фиксированной запятой положение запятой фиксируется в определенном месте относительно разрядов числа.



Обычно подразумевается, что запятая находится или перед старшим разрядом, или после младшего. В первом случае могут быть представлены только числа, которые по модулю меньше 1, во втором – только целые числа.

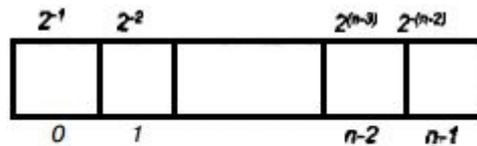


$$2^{-(n-1)} \leq \text{abs}(X) \leq 1 - 2^{-(n-1)}$$

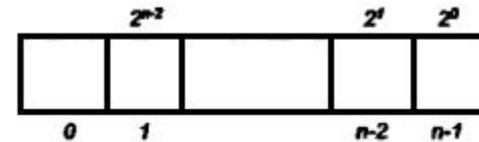


$$0 \leq \text{abs}(X) \leq 2^{n-1} - 1$$

Используют два варианта представления целых чисел: со знаком и без знака. В последнем случае все разряды разрядной сетки служат для представления модуля числа.



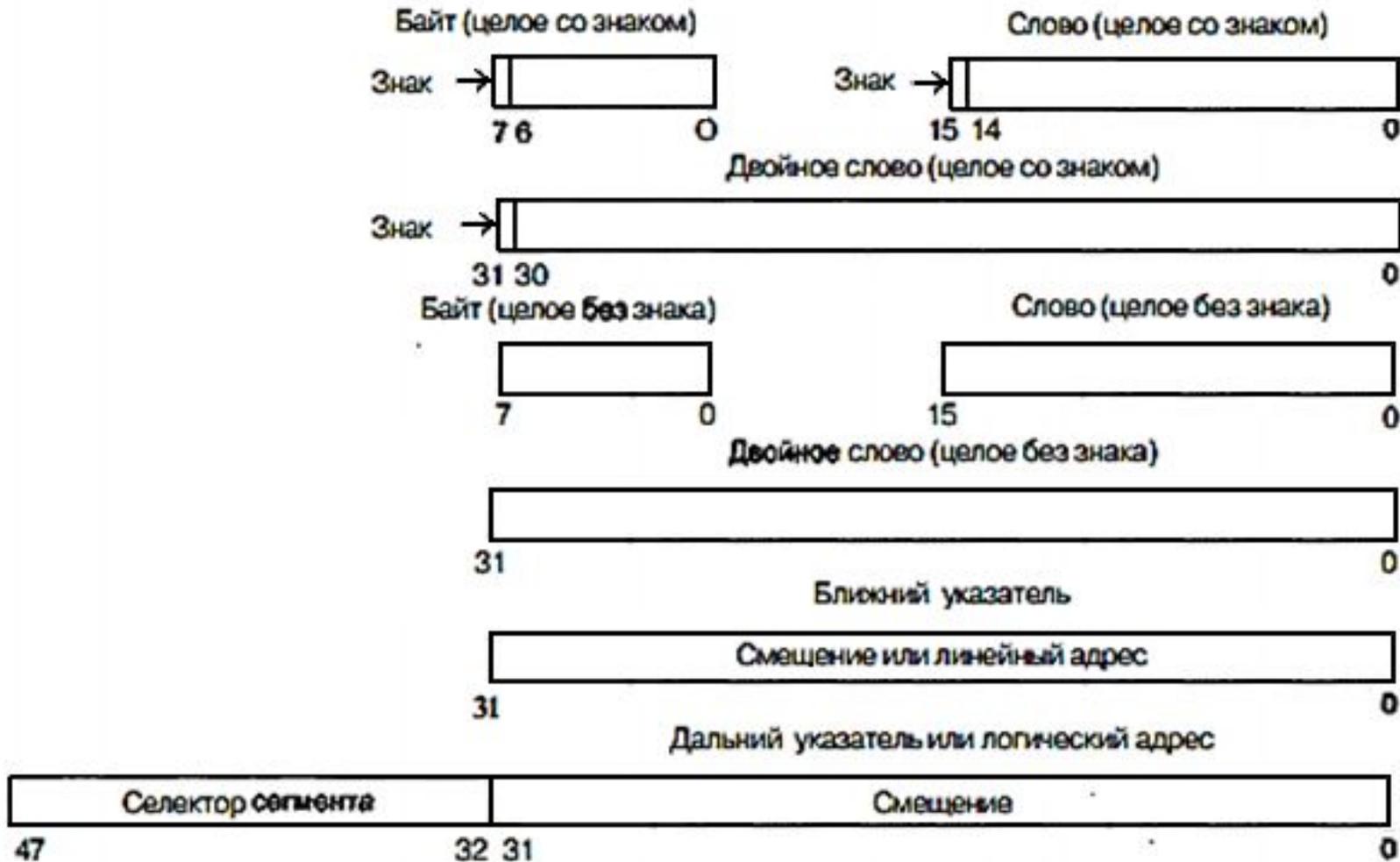
$$2^{-(n-2)} \leq x \leq 1 - 2^{-(n-2)}$$



$$0 \leq \text{abs}(X) \leq 2^{n-1} - 1$$

Особенностью представления целых чисел со знаком в форме с фиксированной запятой в современных ЭВМ является использование дополнительного кода для отрицательных чисел.

Целочисленные форматы микропроцессоров фирмы Intel



Целые числа применяются также для работы с адресами. На рисунке это 32-разрядный формат ближнего и 48-разрядный формат дальних указателей

Упакованные целые числа

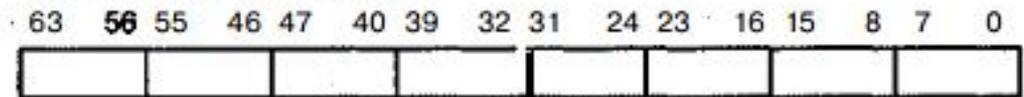
Формат предполагает упаковку в пределах достаточно длинного слова (обычно 64-разрядного) нескольких небольших целых чисел, а соответствующие команды обрабатывают все эти числа параллельно.

Неиспользованные разряды заполняются нулями.

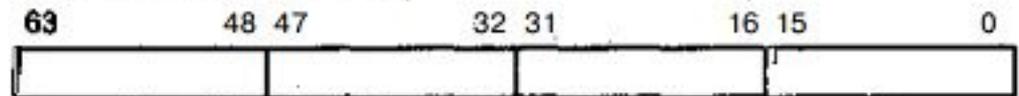
В микропроцессорах фирмы Intel, начиная с Pentium MMX, присутствуют специальные команды для обработки мультимедийной информации (MMX-команды), оперирующие целыми числами, упакованными в квадрослова (64-разрядные слова).

Идентичные форматы упакованных данных применяются также в другой технологии обработки мультимедийной информации, предложенной фирмой AMD (технология 3DNow).

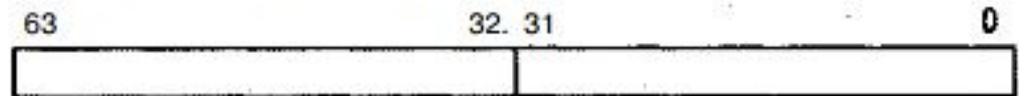
Упакованные байты (8x8 бит)



Упакованные слова (4x16 бит)



Упакованные двойные слова (2x32 бит)



Форматы упакованных целых чисел в технологиях MMX и 3DNow!

Десятичные числа

В основу представления десятичных данных положен принцип кодирования каждой десятичной цифры эквивалентным двоичным числом из четырех битов (*тетрадой*), то есть так называемым двоично-десятичным кодом (*BCD – Binary Coded Decimal*).

Используются два формата представления десятичных чисел (все числа рассматриваются как целые):

1 зонный (распакованный);

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

2 уплотненный (упакованный).

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

В обоих форматах каждая десятичная цифра представляется двоичной тетрадой, то есть заменяется двоично-десятичным кодом. Из оставшихся четырехразрядных двоичных комбинаций две служат для кодирования знаков «+» и «-».

Например, в ВМ семейства IBM 360/370/390 для знака «плюс» выбран код 11002= C16, а для знака «минус» – код 11012= D16.

Зонный формат применяется в операциях ввода/вывода. При выполнении операций сложения и вычитания над десятичными числами обычно используется упакованный формат и в нем же получается результат (умножение и деление возможно только в зонном формате)



Существует большое разнообразие десятичных двоично-кодированных систем ($A_{16}^{10} = 2,9 \cdot 10^{10}$ вариантов). Это многообразие вытекает из избыточности двоичного кода, при котором из 16 возможных комбинаций в каждом разряде используется по прямому информационному назначению лишь 10.

Наиболее широкое применение находят системы кодирования 8421 и 8421+3 (код Штибитца).

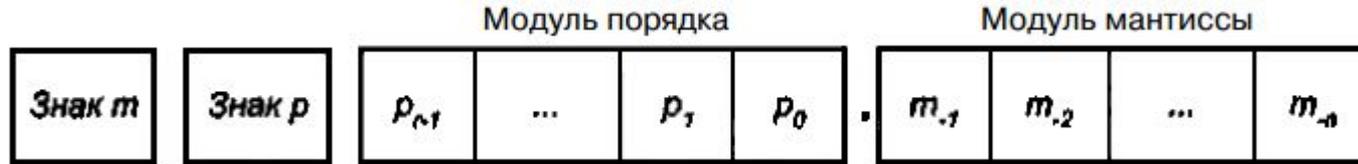
8421		8421+3	
0000 – 0	неудобна тем, что при выполнении операции вычитания нет прямого перехода от цифры каждого разряда к дополнительному коду	0000 – 0	обладает свойством самодополнения – дополнение до 9 можно получить, применяя операцию поразрядного инвертирования кода
0001 – 1		0001 – 1	
0010 – 2		0010 – 2	
0011 – 3		0011 – 3	
0100 – 4		0100 – 4	
0101 – 5		0101 – 5	
0110 – 6		0110 – 6	
0111 – 7	обладает свойством аддитивности, поскольку результаты операции сложения над числами в десятичной системе и над их изображением в системе 8421 – совпадают.	0111 – 7	
1000 – 8		1000 – 8	
1001 – 9		1001 – 9	

Числа в форме с плавающей запятой

Представление чисел с плавающей запятой (ПЗ) известно также под названиями нормальной или полулогарифмической формы.

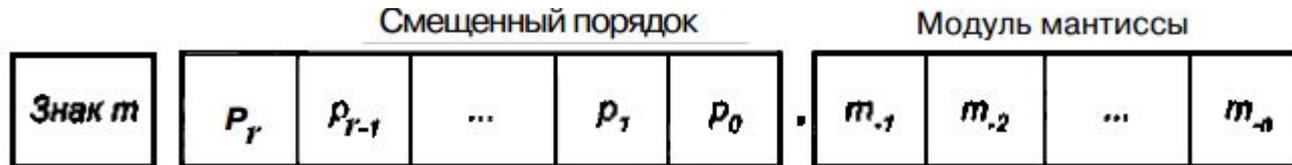
$$R = \pm m \cdot q^{\pm p}$$

где m – мантисса числа R , p – порядок числа, q – основание системы счисления.



Диапазон и точность представления чисел с ПЗ зависят от числа разрядов, отводимых под порядок и мантиссу, а также от основания используемой системы счисления, которое может быть отличным от 2. Например, в универсальных ВМ (мэйнфреймах) фирмы IBM используется база 16.

В большинстве вычислительных машин 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** (. Обычно смещение выбирается равным половине представимого диапазона порядков.



Для устранения неоднозначности смещенные порядки называют **характеристиками**.

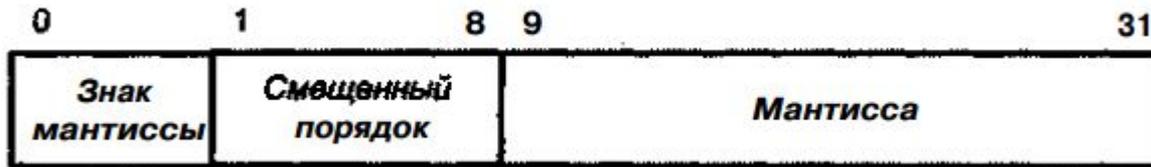


В числах с ПЗ обычно используют нормализованное представление числа в форме с плавающей точкой, то есть мантисса должна быть по модулю меньше единицы и первая значащая цифра мантиссы должна отличаться от нуля ($1/q \leq |m| < 1$). Полученная таким образом мантисса называется **нормализованной**.

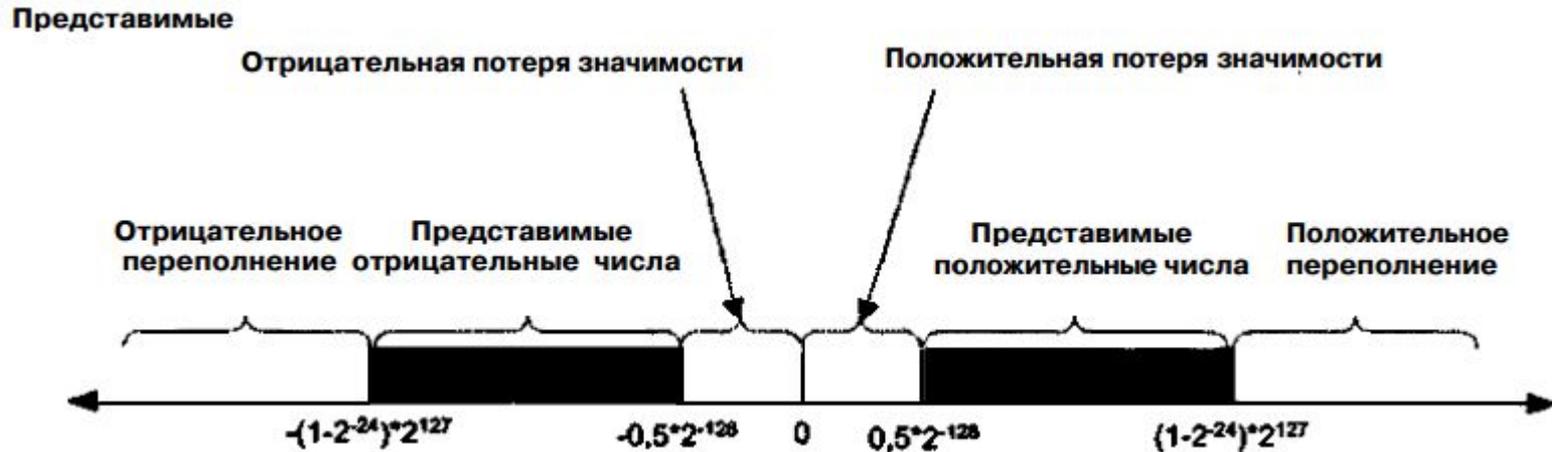
Если для записи числа с ПЗ используется база 2 ($q = 2$), то часто применяют способ повышения точности представления мантиссы, называемый **приемом скрытой единицы** (в нормализованной мантиссе старшая цифра всегда равна единице, следовательно, эта цифра может не записываться, а подразумеваться). Такая запись числа с ПЗ не учитывает нулевого значения. Для этой цели используется специальная кодовая комбинация.



Типичный 32-битовый формат числа с ПЗ



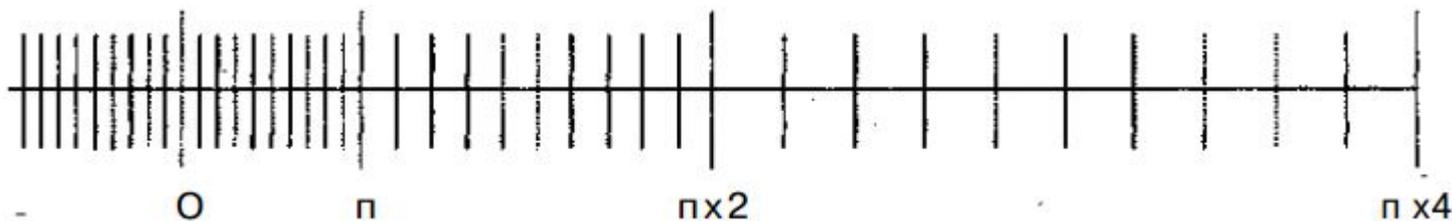
Числа с плавающей запятой, представимые в 32-битовых форматах



Переполнение – ситуация, когда в результате арифметической операции получается значение большее, чем можно представить максимальным порядком.

Потеря значимости – ситуация, когда результат представляет собой слишком маленькое дробное значение.

Числа в форме с ПЗ размещены на числовой оси неравномерно. Возможные значения в начале числовой оси расположены плотнее, а по мере движения вправо – все реже.



Стандарт IEEE754

Основные форматы IEEE754: а – одинарный; б – двойной



а



б

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).

Параметры форматов стандарта IEEE754

Параметр	Формат			
	Одинарный	Одинарный расширенный	Двойной	Двойной расширенный
Ширина слова, бит	32	>43	64	>79
Ширина порядка, бит	8	>11	11	>15
Смещение порядка	127	Не определено	1023	Не определено
Максимальный порядок	127	>1023	1023	16383
Минимальный порядок	-126	<-1022	-1022	<-16382
Диапазон чисел	$10^{-38} \dots 10^{38}$	Не определен	$10^{-308} \dots 10^{308}$	Не определен
Длина мантииссы, бит	23	>31	52	>63
Количество порядков	254	Не определено	2046	Не определено
Количество мантисс	223	Не определено	252	Не определено
Количество значений	$1,98 \times 2^{31}$	Не определено	$1,99 \times 2^{63}$	Не определено



Особые значения чисел с плавающей точкой в IEEE 754

Ноль (со знаком)

Число считается нулём, если все его биты, кроме знакового, равны нулю. При этом в зависимости от значения бита знака ноль может быть как положительным, так и отрицательным.

Неопределенность (*NaN*)

NaN – это аббревиатура от фразы "*not a number*". *NaN* является результатом арифметических операций, если во время их выполнения произошла ошибка. *NaN* представлен как число, в котором все двоичные разряды порядка – единицы, а мантисса не нулевая.

Бесконечности

Число с плавающей запятой считается равным бесконечности, если все двоичные разряды его порядка – единицы, а мантисса равна нулю. Знак бесконечности определяется знаковым битом числа

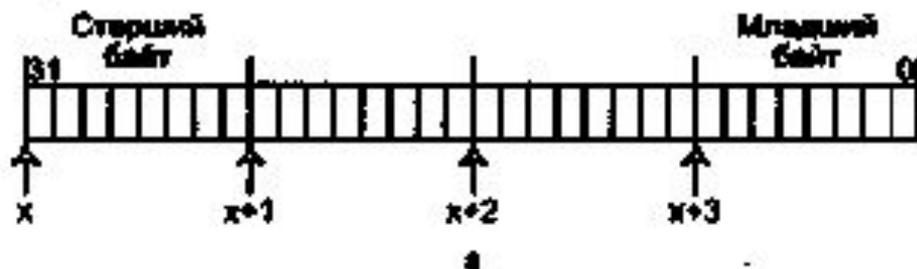


Разрядность основных форматов числовых данных

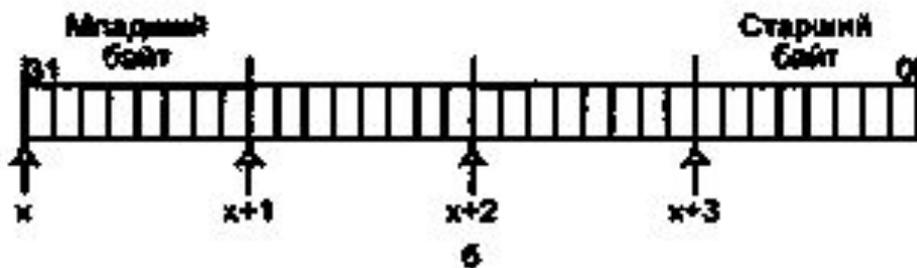
Бит	0
Полубайт (4 бита)	0101
Байт (8 бит)	11010000
Полуслово (16 бит)	0110010011010010
Слово (32 бита)	0111000010000110100100011000110001101
Двойное слово (64 бита)	11001010000101001011001110001101 01110000101110010001110010001010
Четверенное слово (128 бит)	00101100000101001011001110001110001101 011000000101110010001110010001010 11001011001101001010001110001111 00110010101101010001111011101000



Размещение числовых данных в памяти



big endian (IBM, Motorola)



little endian (DEC, Intel)

В настоящее время в большинстве машин предусматривается использование обоих вариантов, причем выбор может быть произведен программным путем за счет соответствующей установки регистра конфигурации.



Помимо порядка размещения байтов, существенным бывает и выбор адреса, с которого может начинаться запись числа. Связано это с физической реализацией полупроводниковых запоминающих устройств, где обычно предусматривается возможность считывания (записи) четырех байтов подряд. Данная операция выполняется быстрее, если адрес первого байта отвечает условию $A \bmod S = 0$ ($S = 2, 4, 8, 16$). Числа, размещенные в памяти в соответствии с этим правилом, называются *выравненными*.

Адрес

0	[битовый паттерн]		
4	[битовый паттерн]	[битовый паттерн]	[битовый паттерн]
8	[битовый паттерн]	[битовый паттерн]	[битовый паттерн]
12	[битовый паттерн]		
16	[битовый паттерн]	[битовый паттерн]	[битовый паттерн]
20	[битовый паттерн]		

Размещение 32-разрядного слова без соблюдения правила выравнивания

Большинство компиляторов генерируют код, в котором предусмотрено выравнивание чисел в памяти.



Символьная информация

Каждому символу ставится в соответствие определенная двоичная комбинация. Совокупность возможных символов и назначенных им двоичных кодов образует *таблицу кодировки*.

Требования к кодировкам:

1. Веса кодов цифр возрастают по мере увеличения цифры, причем подряд;
2. Веса букв увеличиваются в алфавитном порядке (не обязательно подряд).

Наиболее известные 8-разрядные таблицы кодировок:

1. расширенный двоично-кодированный код EBCDIC (Extended Binary Coded, Decimal Interchange Code) – внутренни1 кода в универсальных ВМ фирмы IBM, известен также под названием ДКОИ (двоичный код для обработки информации);
2. американский стандартный код для обмена информацией ASCII (American Standard Code for Information Interchange) – ASCII—7-разрядный, восьмая позиция отводится для записи бита четности.
3. Latin 1 (стандарт ISO 8859-1) – европейская модификация ASCII, использующая все 8 разрядов для кодирования (коды 128-255 отводятся для представления специфических букв алфавитов западно-европейских языков, символов псевдографики, некоторых букв греческого алфавита, а также ряда математических и финансовых символов).



Варианты стандарта ISO 8859

Стандарт	Характеристика
ISO 8859-1	Западно-европейские языки
ISO 8859:2	Языки стран центральной и восточной Европы
ISO 8859-3	Языки стран южной Европы, мальтийский и эсперанто
ISO 8859-4	Языки стран северной Европы
ISO 8859-5	Языки славянских стран с символами кириллицы
, ISO 8859-6	Арабский язык
ISO 8859-7	Современный греческий язык
ISO 8859-8	Языки иврит и идиш
ISO 8859-9	Турецкий язык
ISO 8859-10	Языки стран северной Европы (лапландский, исландский)'
ISO 8859-11	Тайский язык
ISO 8859-13	Языки балтийских стран
ISO 8859-14	Кельтский язык
ISO 8859-15	Комбинированная таблица для европейских языков
ISO 8859-16	Содержит специфические символы ряда языков: албанского, хорватского, английского, финского, французского, немецкого, венгерского, ирландского, итальянского, польского, румынского и словенского



Кодовые страницы OEM

(Original Equipment Manufacturer)

Идентификатор	Страны кодовой страницы
CP437	США, страны западной Европы и Латинской Америки
CP708	Арабские страны
CP737	Греция
CP775	Латвия, Литва, Эстония
CP852	Страны восточной Европы
CP853	Турция
CP855	Страны с кириллической письменностью
CP860	Португалия
CP862	Израиль
CP865	Дания, Норвегия
CP866	Россия
CP932	Япония
CP936	Китай
CP437	США, страны западной Европы и Латинской Америки
CP708	Арабские страны



Кодировки стандарта UNICODE)

Юникод (Unicode) – промышленный стандарт обеспечивающий цифровое представление символов всех письменностей мира, и специальных символов.

Предложен в 1991 году некоммерческой организацией «Консорциум Юникода» (Unicode Consortium, Unicode Inc.).

Стандарт состоит из двух основных разделов: универсальный набор символов (англ. UCS, universal character set) и семейство кодировок (англ. UTF, Unicode transformation format). Универсальный набор символов задаёт однозначное соответствие символов кодам – элементам кодового пространства, представляющим неотрицательные целые числа. Семейство кодировок определяет машинное представление последовательности кодов UCS.

В «естественном» варианте кодировки Unicode, известном как UCS-2, каждый символ описывается двумя последовательными байтами m и n , так что номеру символа соответствует численное значение $256 \times m + n$. Наряду с UCS-2 в рамках Unicode существуют еще несколько вариантов кодировки Unicode (UTF, Unicode Transformation Formats), основные из которых UTF-8 и UTF-7.

Стандарт Unicode обратно совместим с кодировкой ASCII, однако если в ASCII для представления схожих по виду символов (минус, тире, знак переноса) применялся общий код, в Unicode каждый из этих символов имеет уникальную кодировку.



В кодировке UTF-8 коды символов меньше, чем 128, представляются одним байтом. Все остальные коды формируются по более сложным правилам. В зависимости от символа его код может занимать от двух до шести байтов, причем старший бит каждого байта всегда имеет единичное значение.

Схема формирования кодов UTF-8

Число байтов	Двоичное представление	Число свободных битов
1	0xxxxxxx	7
2	110xxxxx 10xxxxxx	11(5 + 6)
3	110xxxxx 10xxxxxx 10xxxxxx	16(4 + 6x2)
4	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx	21(3 + 6x3)
5	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx	26(2 + 6x4)
6	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx	31(1+6x5)

Схема формирования кодов UTF-8

В UTF-7 код символа также может занимать один или более байтов, однако в каждом из байтов значение не превышает 127 (старший бит байта содержит ноль). Многие символы кодируются одним байтом, и их кодировка совпадает с ASCII, однако некоторые коды зарезервированы для использования в качестве преамбулы, характеризующей последующие байты многобайтового кода.



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



Логические данные

Элементом логических данных является логическая (булева) переменная, которая может принимать лишь два значения: «истина» или «ложь». Кодирование логического значения принято осуществлять битом информации: единицей кодируют истинное значение, нулем — ложное. Как правило, в ВМ оперируют наборами логических переменных длиной в машинное слово. Обращаются такие слова с помощью команд логических операций (И, ИЛИ, НЕ и т. д.), при этом все биты обрабатываются одинаково, но независимо друг от друга, то есть никаких переносов между разрядами не возникает



Строки

Строка – непрерывная последовательность битов, байтов, слов или двойных слов.

Битовая строка может начинаться в любой позиции байта и размещается в пределах слова.

Байтовая строка может состоять из байтов, слов или двойных слов.

Если байты байтовой строки представляют собой коды символов, то говорят о текстовой строке. Для указания конца строки в последний байт заносится код-ограничитель – обычно это нули во всех разрядах байта (**нуль-терминальная строка**). Иногда вместо ограничителя длину строки указывают числом, расположенным в первом байте строки.

