

- Типы и форматы операндов -

Операндами называют данные, которыми оперирует вычислительная машина.

Базовые типы операндов:

- Числа;
- Адреса;
- Символы;
- Логические данные;

Более сложные информационные единицы:

- графические изображение;
- аудиоинформация;
- видеоинформации;

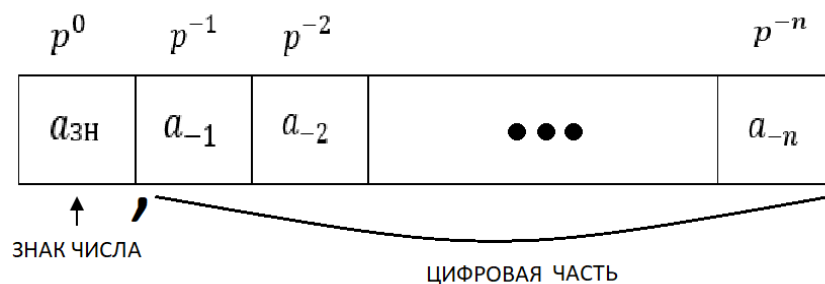
Информации является производной от базовых типов данных и храниться на внешних запоминающих устройств с использованием определенных форматов.

- Представление чисел в ЭВМ -

- Числа с фиксированной запятой (точкой).
- Числа с плавающей запятой.

Формат представления чисел с фиксированной запятой.

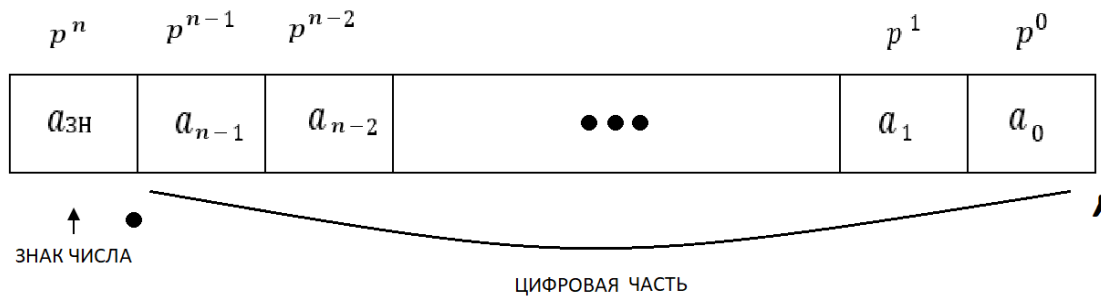
Запятая фиксируется перед старшим цифровым разрядом.



В формате с фиксированной запятой представляются правильные дроби. Такой формат использовался в ЭВМ первых поколений.

- Формат представления чисел с фиксированной точкой. -

Запятая фиксируется после младшего цифрового разряда.



В формате с фиксированной точкой представляются целые числа. Такой формат используется в современных ЭВМ.

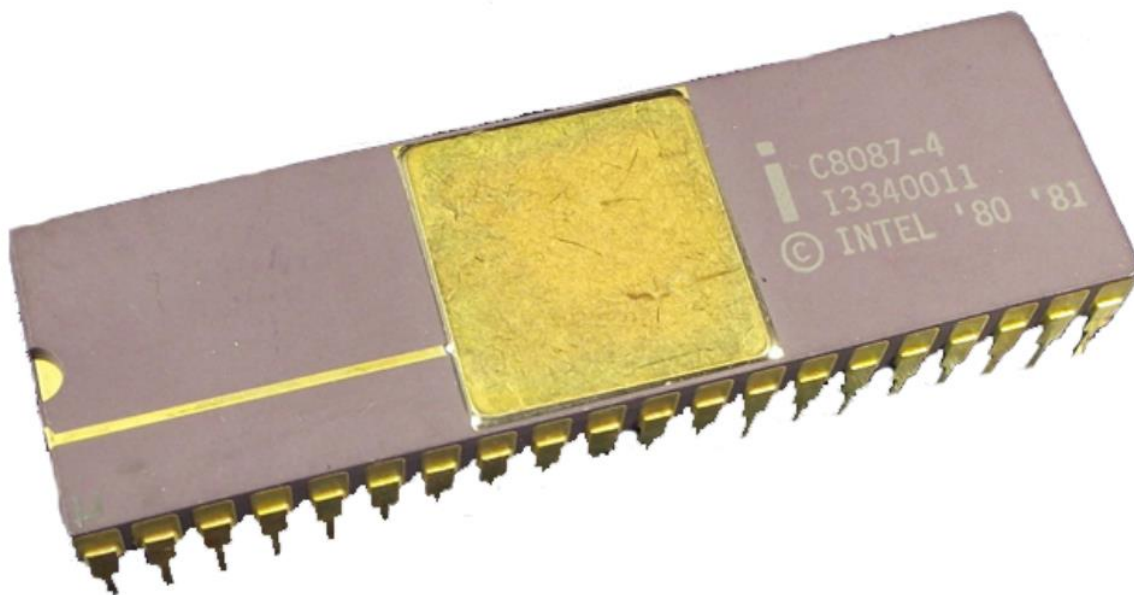
В состав современных ЭВМ входят различные типы операционных устройств, предназначенных для работы с соответствующими типами информации. Так, с целыми числами работает целочисленное арифметическое устройство, входящее в состав центрального процессора. Для работы с вещественными числами используются устройство FPU (Float Point Unit). Для работы с аудиоинформацией используется специализированный процессор (звуковая карта), для работы с видеоинформацией используется видеокарта, представляющая собой специализированный процессор.

Рассмотрим форматы на примере процессоров семейства x86.

Первый процессор этого семейства 1978 году стал **Intel 8086**.



Он мог выполнять операции только над целыми числами. Для работы с вещественными числами необходимо бы приобрести **математический сопроцессор Intel 8087**.



С появлением этого сопроцессора появился стандарт, разработанный международным центром стандартизации IEEE (Institute of electrical and electronics engineers) IEEE-754.

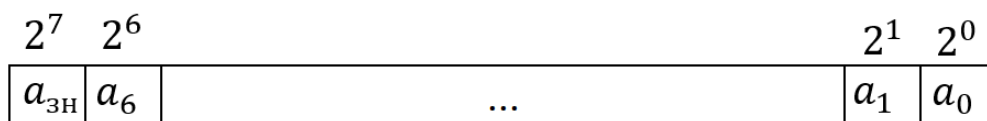
В соответствии с этим стандартом целые числа могут представляться в форматах:

- байт (8 разрядов);
- слово (16 разрядов, 2 байта);
- двойное слово (32 разряда, 4 байта);
- четверное слово (64 разряда, 8 байт);

Причём числа могут быть представлены как со знаком, так и без знака.

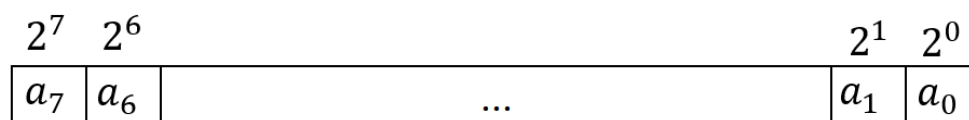
Число со знаком представляется в дополнительном коде. Нуль представляется единственным способом $\{+0\}_{\text{д.к.}} = 0,00 \dots 0$. За счёт этого появляется возможность представить дополнительное число равное 2^n , где n соответствует весу знакового разряда.

Целое со знаком размером в байт:



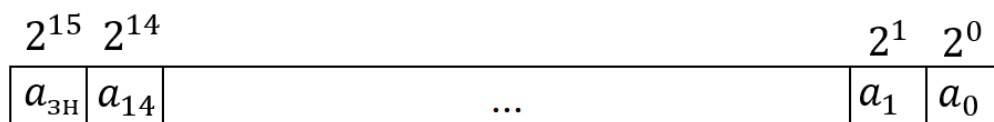
Диапазон $[-128, +128]$

Целое без знака размером в байт:



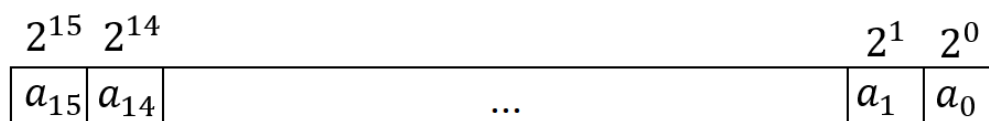
Диапазон $[0, 255]$

Целое со знаком размером в слово:



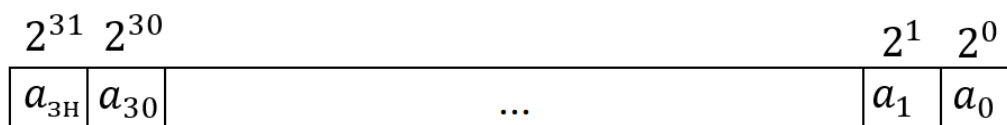
Диапазон $[-2^{15}; 2^{15} - 1] = [-32768 : +32767]$

Целое без знака размером в слово:



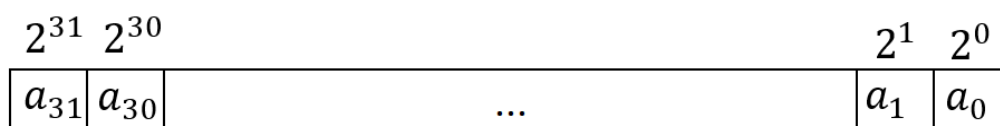
Диапазон $[0 : 2^{16} - 1] = [0 : 65535]$

Целое со знаком размером в двойное слово:



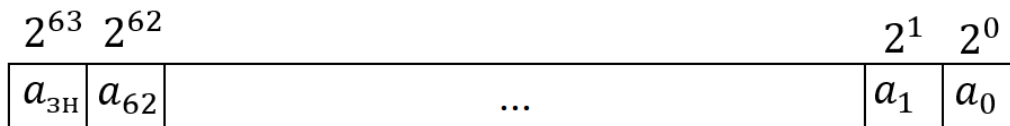
Диапазон $[-2^{31}; 2^{30} - 1] = [-2147483648 : 2147483647]$

Целое без знака размером в двойное слово:



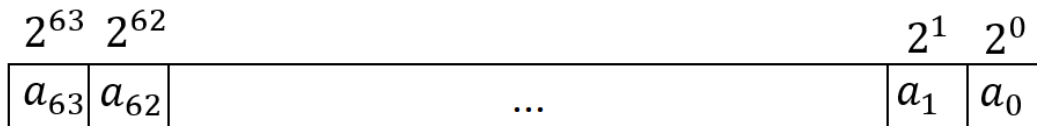
Диапазон $[0 : 2^{32} - 1]$

Целое со знаком размером в четверное слово:



Диапазонов $[-2^{63}; 2^{63} - 1]$

Целое без знака размером в четверное слово:



Диапазонов $[0 : 2^{64} - 1]$

- Упакованные целые числа -

В современных ЭВМ имеются команды, работающие с целыми числами, представленными в упакованном виде. Это связано с обработкой мультимедийной информации.

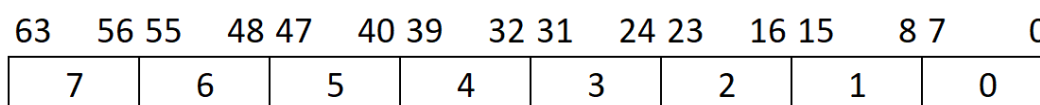
Формат предполагает упаковку в пределах одного слова нескольких небольших целых чисел, а соответствующие команды обрабатывают эти числа параллельно.

На примере процессоров **x86** такая технология появилась в процессорах Pentium и стала называться **MMX** технология. Intel добавила в систему 57 новых команд, относящихся к классу **SIMD (Single instruction Multiple Data)**. Поводом послужило широкое распространение мультимедийных приложений. Видео, трехмерная графика и звук в ЭВМ представляются большими массивами данных, элементы которых обрабатываются идентично.

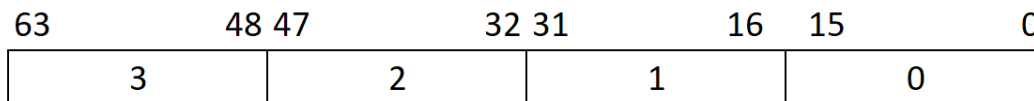
- MMX (multimedia extension) технология. -

В технология **MMX** используются форматы:

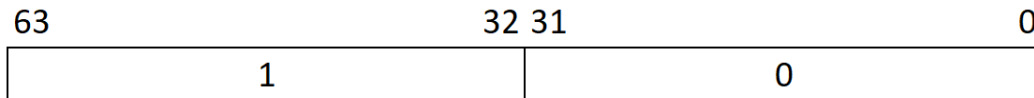
Упакованные байты (8x8)



Упакованные слова (4x16)



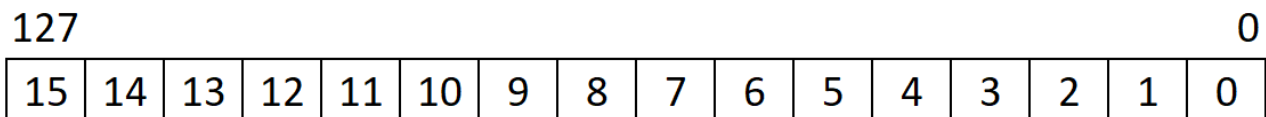
Упакованные двойные слова (2x32)



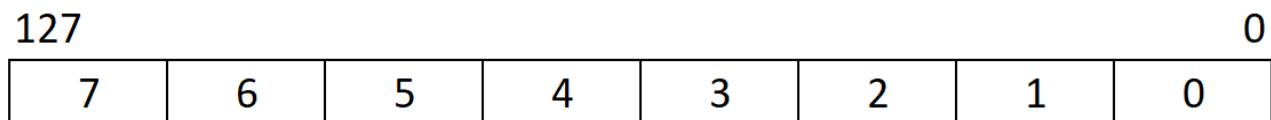
Идентичные форматы упакованных данных применяется в другой технологии, предложенной компанией **AMD** и называемая **3DNow**. Она реализована в микропроцессорах этой фирмы.

Со временем формировалась единая технология, известная под аббревиатурой **SSE (Streaming SIMD Extensions)**. В микропроцессорах последних поколений (версия **SSE IV**) за основу принято 128-разрядное слово и предусмотрены 4 формата упакованных целых чисел.

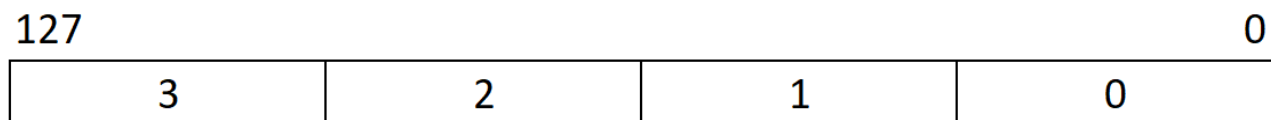
Упакованные байты (16x8):



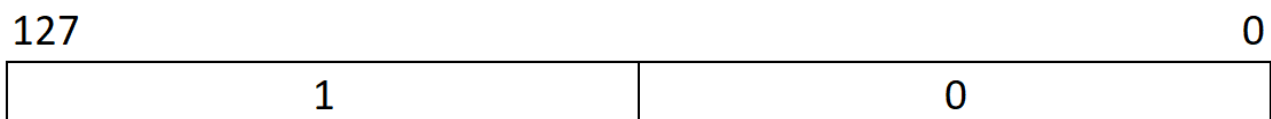
Упакованные слова (8x16):



Упакованные двойные слова (4x32):



Упакованные четверные слова (2x64):



- Десятичные числа -

В ряде задач, главным образом учётно-статического характера, приходится иметь дело с десятичными числами. Перевод в двоичную, выполнение операции и обратный переход в десятичное число сопряжен с большими расходами. Поэтому в вычислительных машинах применяются форматы, позволяющие представлять и выполнять арифметические действия над десятичными числами.

Упакованный формат предполагает что байт содержит две десятичные цифры, представленные в коде D1 с естественными весами разрядов 8,4,2,1. Знак числа представляется в младшей тетраде младшего байта. Для представления знака «+» используется комбинация 1100 (C_{16}). Для представления знака «-» используется комбинация 1101 (D_{16}). Если используется представление чисел без знака, тогда в позиции, отводимой под знак указывается комбинация 1111 (F_{16}).

байт		...	байт		байт	
цифра	цифра		цифра	цифра	цифра	знак

Представление числа «-3459» упакованном формате:

0	3	4	5	9	минус
0000	0011	0100	0101	1001	1101

Десятичное число должно занимать целое число байтов.

Зонный формат распределён, главным образом в больших универсальных вычислительных машинах семейства **IBM-360/370/390**. В нём под каждую цифру выделяется 1 байт, где младшие 4 разряда отводятся под цифры, а в старшую тетраду (поле зоны) записывается специальный код «зона», не совпадающий с кодами цифр и знаков. Для IBM код зоны $1111_2 = F_{16}$. В младшем байте в поле зоны представляется знак числа.

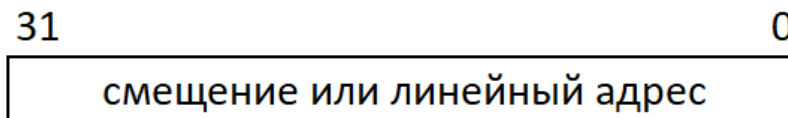
Представление числа -3459 в зонном формате:

зона	3	зона	4	зона	5	минус	9
1111	0011	1111	0100	1111	0101	1101	1001

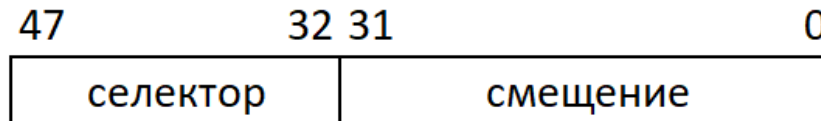
В упакованном формате байт содержит одну десятичную цифру, представленную в коде **ASCII**, так **цифра 0 кодируется 30h, 1 – 31h, ... , 9-39h**.

Целые числа применяются также для работы с адресами. Применительно к процессорам **Intel x86** используются следующие форматы:

Ближний указатель



Дальний указатель



- Целые числа в формате с плавающей запятой -

Число разбивается на две группы цифр: на мантиссу и порядок.



Диапазон зависит от числа разрядов представления порядка и от основания системы счисления, точность зависит от числа разрядов представления мантиссы.

$$X = P^{\pm m} \cdot M, \text{ где}$$

P - основание системы счисления

m - порядок числа

M – мантисса

P = 2, для больших современных ЭВМ

P = 16, для универсальных ЭВМ компании **IBM**

P = 32, для **ЭВМ B5500** фирмы **Burroughs**

В большинстве ЭВМ для упрощения арифметики порядок приводят к целым положительным числам, применяя так называемый смещённый порядок.

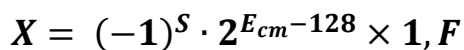
Если под порядок отвести 8 разрядов, то с помощью дополнительного кода можно представить числа из диапазона $[-128 \div +127]$. Прибавив 128 получем смещенный порядок из диапазона от 0 до 255.

$$-128 \Rightarrow 0 \quad -127 \Rightarrow 1 \quad \dots \quad 127 \Rightarrow 255$$

Для однозначного представления чисел в формате с плавающей запятой введено понятие нормализованное число.

Для более существенного увеличения точности для представления числа с плавающей запятой отводят несколько слов, например 2.

Действительное число одинарной точности



$$|X|_{max} = 2^{-|E|_{max}} \times M_{max} = 2^{+127} \times 1,11 \dots 1 = 2^{+128} \approx 10^{+38}$$

63	62	52	51	0
S	E_{CM}		F	

$$|X|_{min} = 2^{-1024} \times 1,00 \dots 0 = 2^{-1024} \approx 10^{-308}$$

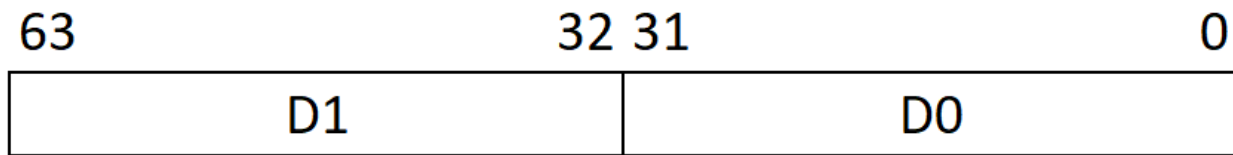
$$|X|_{max} = 2^{+1023} \times 1,11 \dots 1 = 2^{+1024} \approx 10^{+308}$$

79	78	64	63	0
S	E_{CM}	F		

Используются внутри FPU для выполнения арифметических операций.

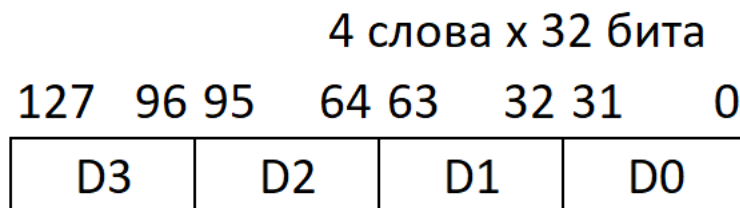
- Форматы упакованных чисел с плавающей запятой -

Технология 3DNow предложенная компанией **AMD** использует два числа одинарной точности.

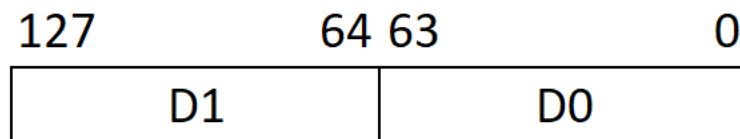


В процессорах **Intel** начиная с **Pentium 3** также реализована технология, ориентированная на параллельную обработку упакованных чисел с плавающей запятой.

Технология названа **SSE (Streaming SIMD Extensions)**.



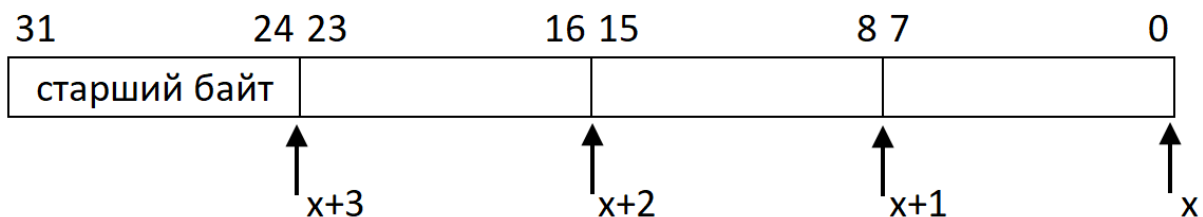
Дальнейшее развитие получила в **Pentium 4** и названа **SSE2**.



- Размещение числовых данных в памяти -

Минимальной адресуемой единицей в памяти является байт. Большая часть форматов подразумевает использование нескольких байт для представления числа. Встаёт вопрос какой из адресов считать за адрес числа.

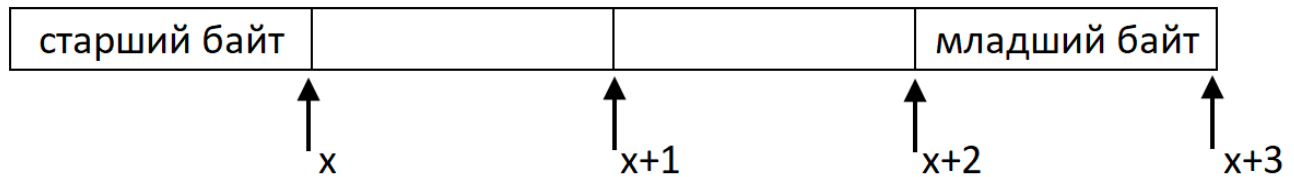
Возможны следующие варианты. За адрес числа применяется адрес младшего байта.



Такой подход используется в ЭВМ **Intel** и **DEC**.

Байты адресуются справа налево и такой порядок адресации называется прямым порядком байтов (Little endian).

За адрес слова применяется адрес старшего байта



Такой подход используется в ЭВМ компании **IBM** и **Motorola**. Байты адресуются слева направо и такой порядок адресации называется обратным порядком байтов (big-endian).

Для современной машины предусматривается использование обоих вариантов при чём выбор может быть произведен программным путём за счёт соответствующей установки регистра конфигурации.

- Выравнивание данных -

Помимо порядка размещения байтов, существенным бывает и выбор адрес, с которого может начинаться запись числа. Связано это с физической реализацией оперативной памяти, которая выполняет запись или считывание одновременно нескольких байт информации. Причём данная операция выполняется быстрее если адрес первого байта пересылаемого слова будет кратным длине слова в байтах.

Если разрядность шины данных 16 разрядов, то адреса выровненных слов должны быть четными.

Если разрядность шины данных 32 разряда, то адреса выровненных двойных слов должны быть кратны 4.

Если шина 64-разрядная, то выровненные адреса должны быть кратны 8.

- Размещение чисел в памяти с выравниванием -

	байт	0
		4
	слово	8
		ch
двойное слово		10h
		14h
четверённое слово		18h

Архитектура в **80x86** не требует выравнивания, однако обращение в память обычно быстрее если операнды выровнены.

Некоторые архитектуры, подобные **ARM** и **MIPS** требуют чтобы объекты в памяти были выровнены.

- Символьная информация -

В эту группу входят буквы, цифры, знаки препинания, математические и другие символы.

Совокупность возможных символов и назначенных им двоичных кодов образуют таблицу кодировки.

До недавнего времени распространёнными были кодировки таблицы в которых символы кодируются с помощью байта:

- Расширенный двоично-кодированный код EBCDIC (Extension Binary Coded Decimal Interchange Code);

- Американский стандартный код для обмена информацией ASCII (American Standard Code for Information Interchange);

Первый вариант используется в качестве внутреннего кода в универсальных вычислительных машинах фирмы **IBM**. Он же известен под названием ДКОИ (Двоичный Код для Обработки Информации).

Стандартный код **ASCII** – 7 разрядный, 8 разряд используется для записи бита четности. Это обеспечивает представление 128 символов, включая латинские буквы, цифры, знаки математических операций и знаки пунктуации.

Позже появилась европейская модификация **Latin 1** в которой используются все 8 разрядов. Дополнительные 128 символов отводится для представления специфических букв западно-европейских языков, символов псевдографики, ряда математических и финансовых символов. Именно эта кодировочная таблица считается мировым стандартом де-факто, который применяется с различными модификациями во всех странах.

- Unicode -

В 1993 году консорциум компании **Apple, Microsoft, Hewlett-Packard, DEC, и IBM** разработал 16 битовый стандарт **ISO10646** определяющий универсальный набор символов (UCS, Universal Character Set).

Новый код, известный под названием **UNICODE**, позволяет задать до 65536 символов, включая символы всех основных «живых» и «мертвых» языков. Для букв русского языка выделены коды с 1040 по 1093.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
03D	б	э	у	у	у	ф	ш	ж	о	о	с	с	ф	ф	ж	ж
03E	э	э	ш	ш	ч	ч	б	э	э	э	х	х	б	б	т	т
03F	х	е	с	ј	е	е	э	р	р	с	м	м	р	о	е	э
040	ё	ё	т	г	е	с	і	ї	ј	л	н	т	к	й	у	ц
041	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
042	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
043	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
044	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
045	ё	ё	т	г	е	с	і	ї	ј	л	н	т	к	й	у	ц
046	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
047	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ
048	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ

- Логические данные -

Элементом логических данных является булева переменная, которая может принимать лишь два значения: «истина» или «ложь». Единицей кодируется истинное значение, нулём – ложное.

ЭВМ оперируют наборами логических переменных длиной в машинное слово. Обработываются такие слова с помощью команд логических операций (И, ИЛИ, НЕ, и т. д.). При этом все биты обрабатываются одинаково, но независимо друг от друга.

- Строки -

Строки - это непрерывная последовательность битов, байтов, слов или двойных слов. Если байты байтовой строки представляют собой коды символов, то говорят о текстовой строке. Длина строки может меняться в очень широких пределах. Существует два способа определения длины строки.

Первый из них заключается в использовании специального управляющего символа, обозначающего конец строки и являющегося его последним символом. Обычно это нули во всех разрядах байта.

Второй способ состоит в использовании отдельного слова памяти или регистра процессора, содержащего число, которое определяет длину строки в байтах.