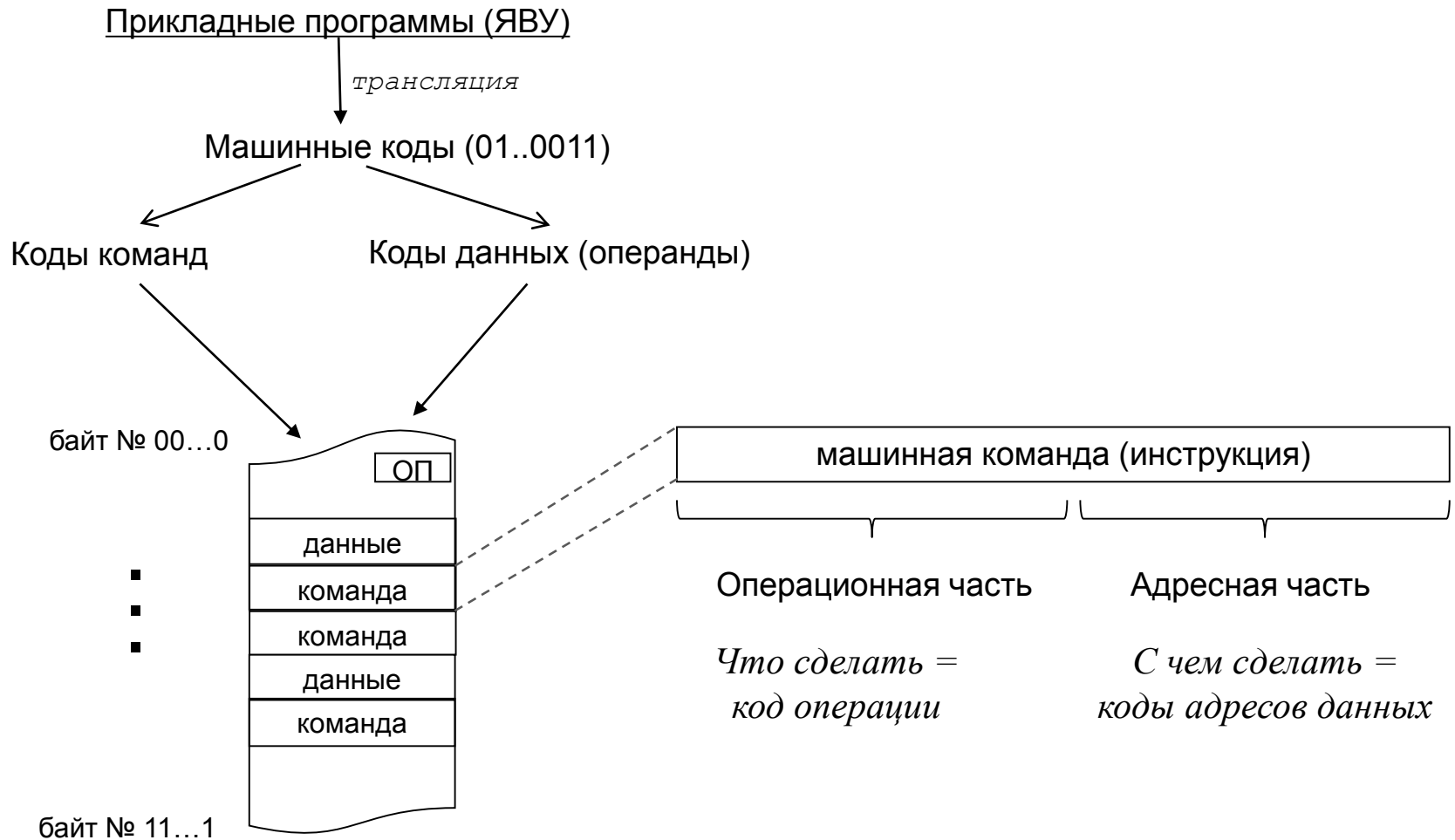


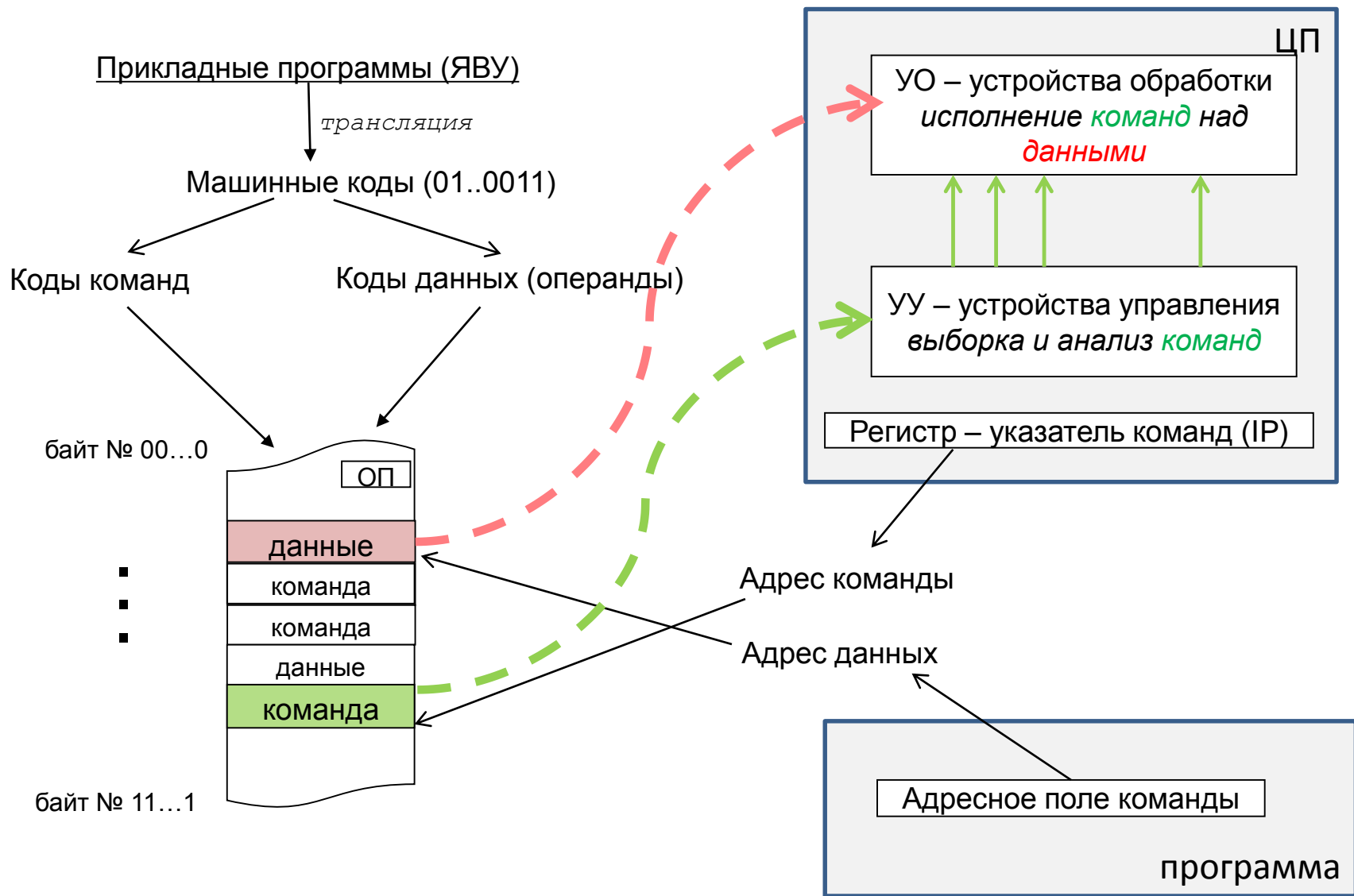
Архитектура ВС



Архитектура ВС. Уровень системы команд



Архитектура ВС. Уровень системы команд.



Побайтное хранение кодов

Размер стандартных машинных элементов данных кратен степеням двойки:

- 1 байт
- 2 байта – слово
- 4 байта – двойное слово
- 8 байт – учетверённое слово
- ...

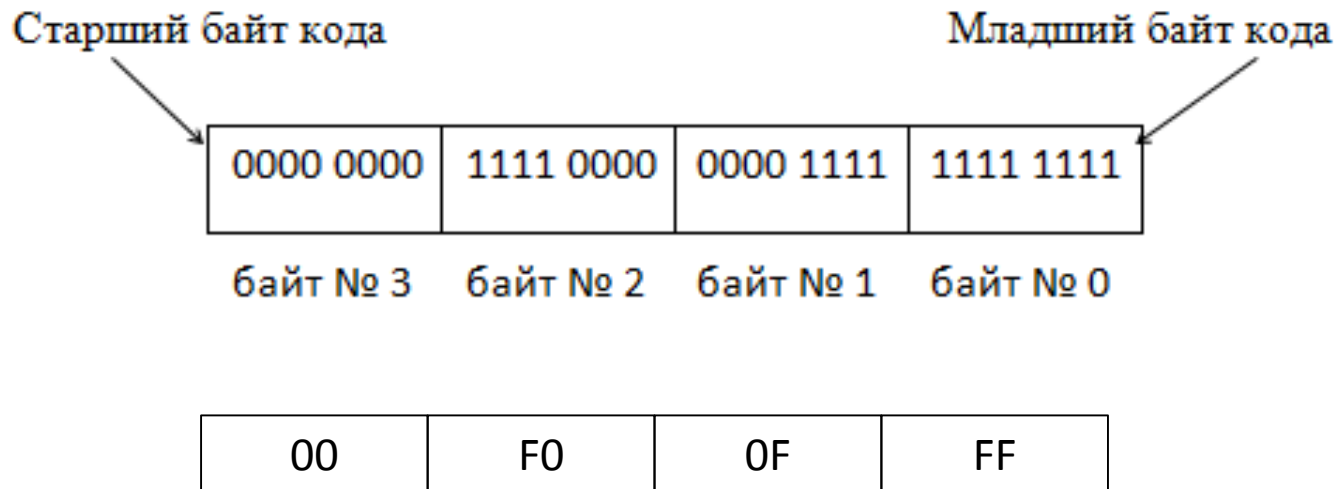
2^0 байт = 8 бит

2^1 байт = 16 бит

2^2 байт = 32 бит

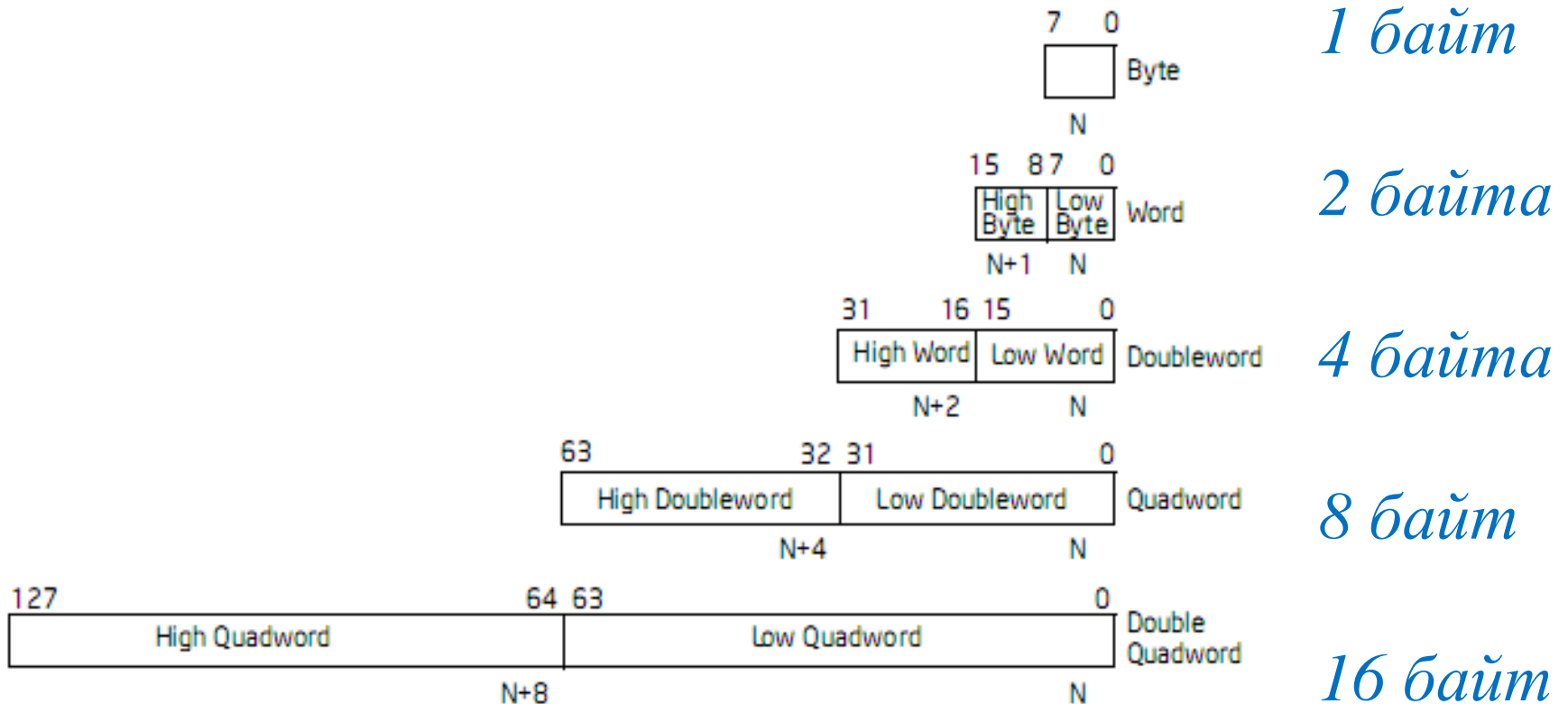
2^3 байт = 64 бит

Двоичное представление числа $15\,732\,735_{10}$ в четырёх байтах

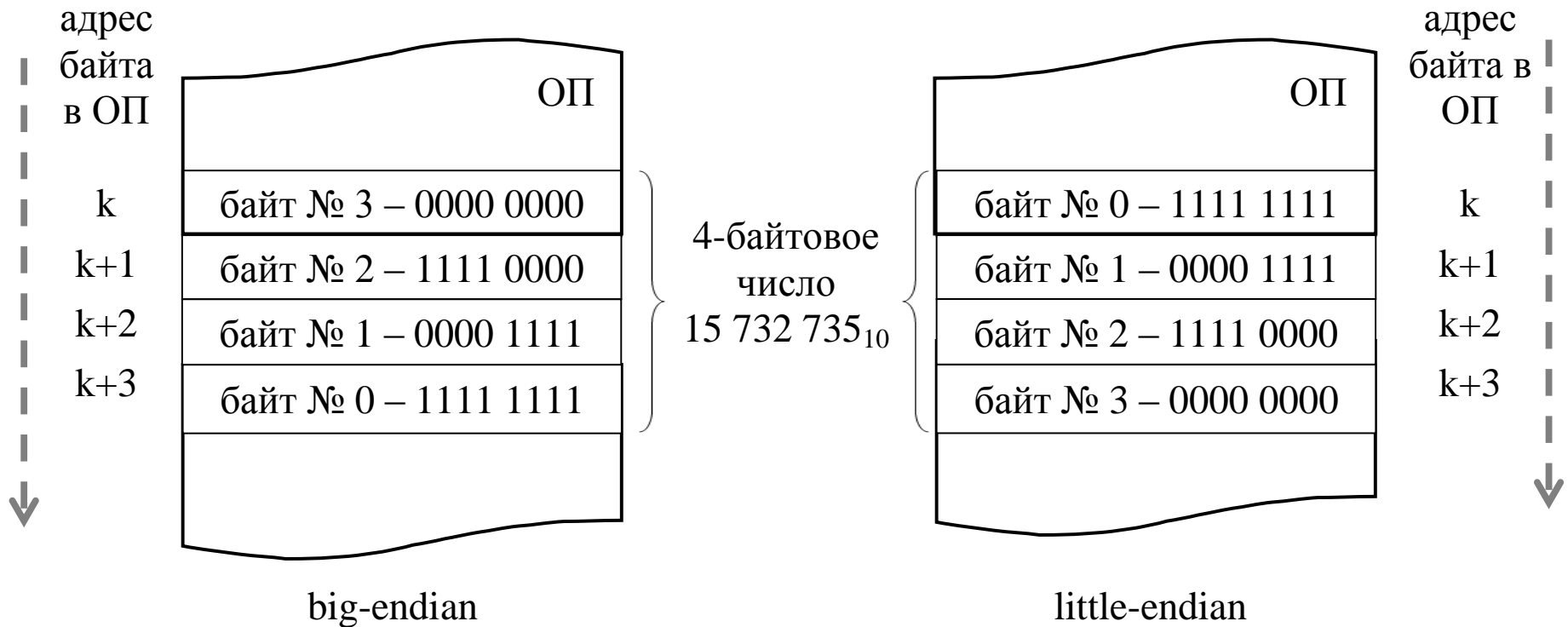


Самостоятельно повторить
двоичную и шестнадцатеричную
систему счисления, правила
перевода целых и дробных чисел.

Основные размеры данных



СПОСОБЫ ХРАНЕНИЯ МНОГОБАЙТОВЫХ ЭЛЕМЕНТОВ ДАННЫХ В ПАМЯТИ КОМПЬЮТЕРА



Прямой: от старшего к младшему
IBM, Motorola, SPARC, MIPS

Обратный: от младшего к старшему
Intel, MIPS

Порядок следования байтов числа в памяти

Типы данных (операнды)

- биты (одна двоичная цифра)
- символьные (единичные символы или строки символов),
- натуральные двоичные числа (целые без знака),
- целые двоичные числа со знаком,
- дробные двоичные числа,
- десятичные цифры (из которых можно программно сформировать число),
- вектора из нескольких целых или нескольких дробных двоичных чисел,
- указатели (ближние и дальние) - адреса,
- теги — признаки,
- дескрипторы — описатели (составные из нескольких типов).

СИМВОЛЬНЫЕ ДАННЫЕ

в виде двоичных кодов символов (целое число без знака)

- *ASCII (каждый символ кодируется однобайтовым двоичным числом)*

Address	Hex dump	ASCII
00403000	28 41 29 3D 33 38 20 21	(A)=38 ↑

- *Unicode (каждый символ кодируется двухбайтовым двоичным числом)*

Используется формат `unsigned char` (символ/байт без знака), в котором могут быть записаны коды от «0000 0000» до «1111 1111» (0÷255 или 00÷FF).

СИМВОЛЬНЫЕ ДАННЫЕ

ASCII-символ (Байт без знака)

Описание для Intel

Symbol db "A"

или

Symbol db 41

Address	Hex dump	ASCII
00403000	41 00 14 00 F0 E5 E7 F3	A. .đâćó

Описание для MIPS

Symbol db 'A'

или

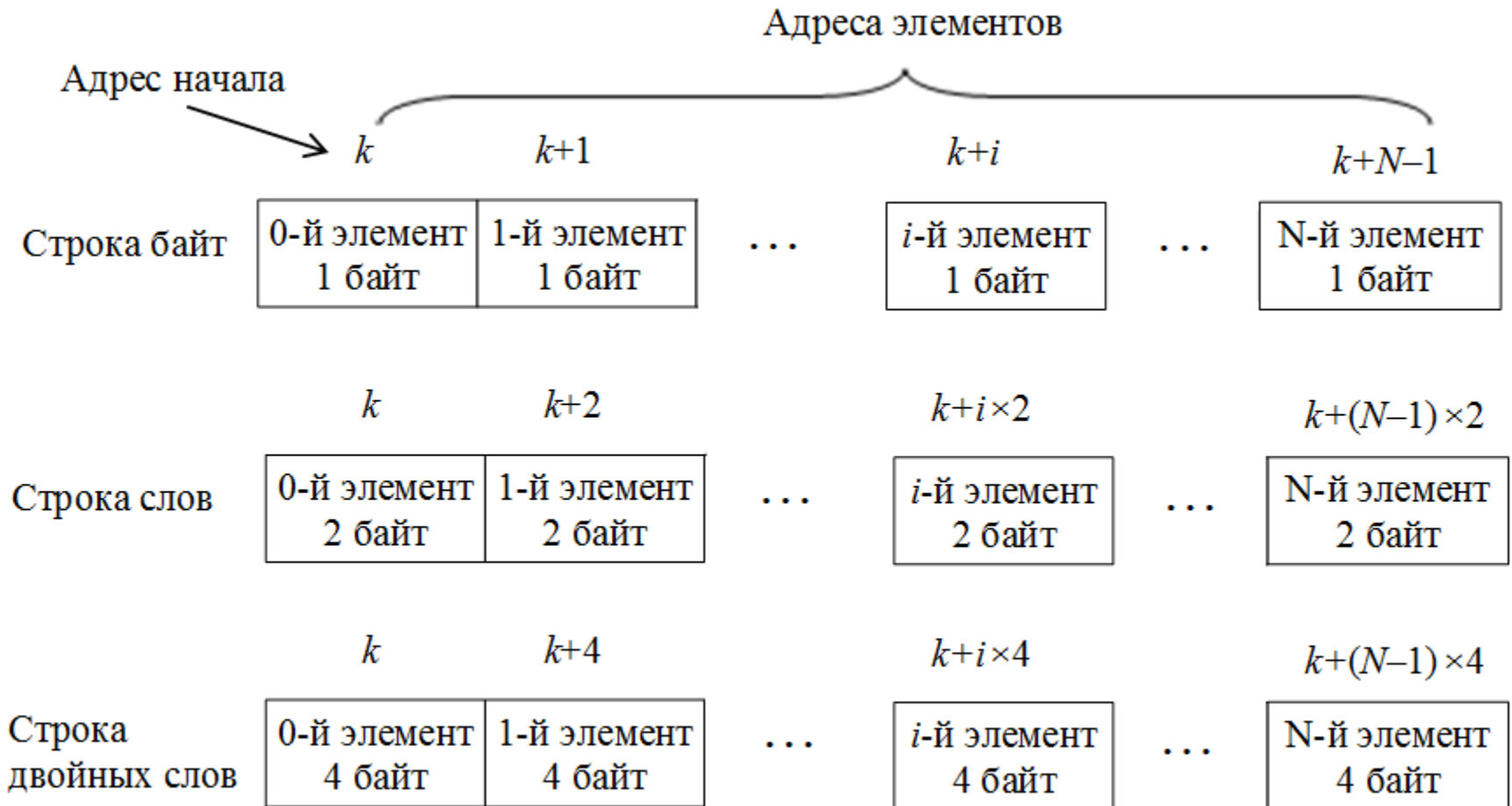
Symbol db 41

```
.data  
simb: .byte 'A', 'B', 'C', 0x41
```

Data Segment	
Address	Value (+0)
0x10010000	0x41434241

Data Segment	
Address	Value (+0)
0x10010000	A C B A

СИМВОЛЬНЫЕ СТРОКИ=одномерные массивы



$$adr(i^{\text{го}} \text{ элемента}) = adr(\text{начала строки/массива}) + i \times (\text{размер элемента})$$

СИМВОЛЬНЫЕ СТРОКИ

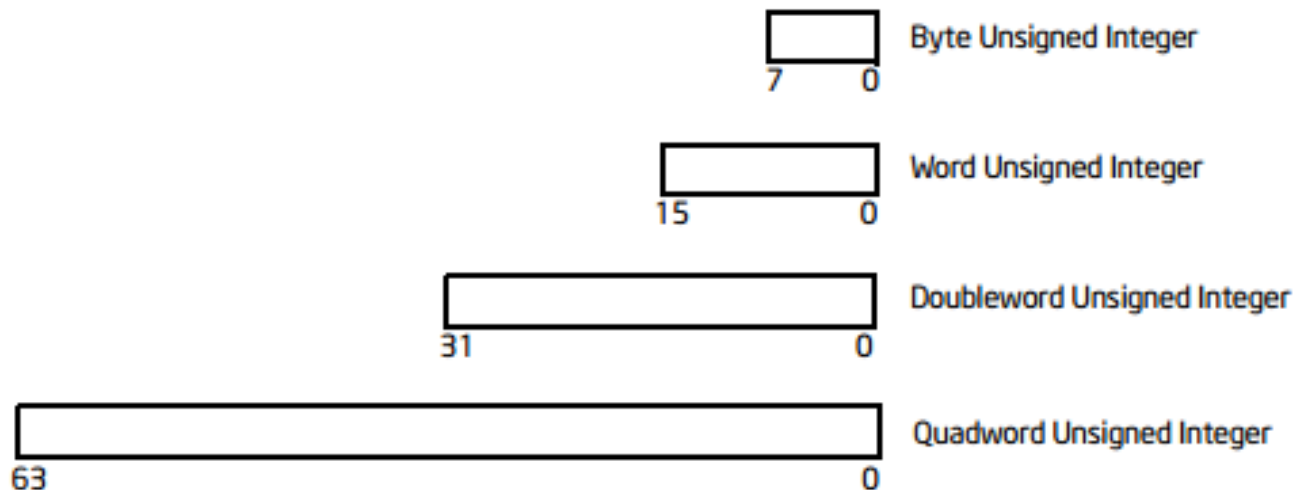
- имя (адрес начала),
- размер элемента,
- длина (количество **БАЙТОВ**),
- конец (завершающий символ «NULL» с кодом 00).



Минимальная длина строки - ноль байт (пустая строка: "", 0),

Максимальная длина строки ограничена максимальным адресом элемента строки, который может быть запрограммирован в данной ВС. Для 32-разрядных процессоров можно задать адрес от 0 до $2^{32}-1$ (32 единицы), значит максимальный размер строки в этом случае = 2^{32} байт = 4 Гбайт.

ЦЕЛЫЕ ДВОИЧНЫЕ ЧИСЛА (БЕЗ ЗНАКА)



Если на число отводится n бит, то диапазон представимых чисел будет $[0; 2^n - 1]$ $n = 8, 16, 32, 64, 128 \dots$

Название формата	Количество байт	Количество разрядов	Диапазон представимых чисел
<i>unsigned byte integer</i>	1	8	$[0; 2^8 - 1]$ или $0 \div 255$
<i>unsigned word integer</i>	2	16	$[0; 2^{16} - 1]$
<i>unsigned doubleword integer</i>	4	32	$[0; 2^{32} - 1]$
<i>unsigned quadword integer</i>	8	64	$[0; 2^{64} - 1]$

Форматы некоторых беззнаковых чисел

ЦЕЛЫЕ ДВОИЧНЫЕ ЧИСЛА (БЕЗ ЗНАКА)

Описание для Intel

- A DB 1,2,3,4,5,6,7,8 ;массив из 8 однобайтовых элементов (byte)
- B DW 1,8,1024,4095 ;массив из 4 двухбайтовых элементов (word)
- C DD 255,15732735 ;массив из 2 четырёхбайтовых элементов (doubleword)
- D DQ 266 ;одно восьмибайтовое число (quadword)

Address	Hex dump							
00403000	01	02	03	04	05	06	07	08
00403008	01	00	08	00	00	04	FF	0F
00403010	FF	00	00	00	FF	0F	F0	00
00403018	0A	01	00	00	00	00	00	00

- 1, 2, 3, 4, 5, 6, 7, 8
- 1, 8, 1024, 4095
- 255, 15732735
- 266₁₀ (=10A₁₆)

Пример хранения кодов целых беззнаковых чисел *перевёрнуто согласно little-endian*

+15 732 735₁₀ = 0000 0000 1111 0000 0000 1111 1111 1111₂ = 00 F0 0F FF₁₆

ЦЕЛЫЕ ДВОИЧНЫЕ ЧИСЛА (БЕЗ ЗНАКА)

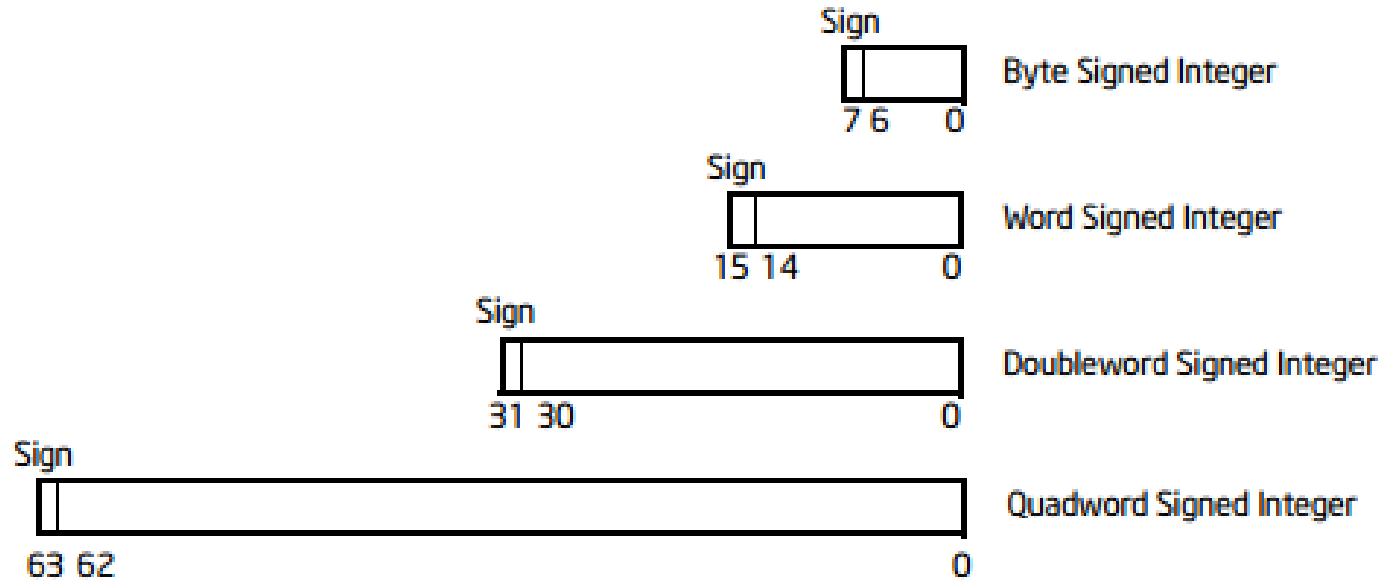
Описание для MIPS

```
        .data  
MEM1:   .byte    0x12  
MEM2:   .half    0x3456  
MEM4:   .word    0x789abcde
```

Labels	
Label	Address ▲
define.asm	
MEM1	0x10010000
MEM2	0x10010002
MEM4	0x10010004

Data Segment		
Address	Value (+0)	Value (+4)
0x10010000	0x34560012	0x789abcde

ЦЕЛЫЕ ДВОИЧНЫЕ ЧИСЛА СО ЗНАКОМ



НОЛЬ

↙ ↘

$[+0]_{\text{доп}} = 0.00\dots 0$ и $[-0]_{\text{доп}} = 1.00\dots 0$

Если оставить только один код нуля ($0.00\dots 0$), то можно использовать второй код для хранения дополнительного отрицательного значения.

ЦЕЛЫЕ ДВОИЧНЫЕ ЧИСЛА СО ЗНАКОМ

Если на число отводится n бит, из которых один – для хранения знака числа, то диапазон представимых чисел будет

$$[-(2^{n-1}-1); -0; +0; 2^{n-1}-1] \longrightarrow [-2^{n-1}; -(2^{n-1}-1); +0; 2^{n-1}-1]$$

$$[-2^{n-1}; 2^{n-1}-1] \quad n = 8, 16, 32, 64, 128 \dots$$

Название формата	Количество байт	Количество разрядов	Диапазон представимых чисел
<i>signed byte integer</i>	1	8	$[-2^7; 2^7-1]$ или $[-128; +127]$
<i>signed word integer</i>	2	16	$[-2^{15}; 2^{15}-1]$
<i>signed doubleword integer</i>	4	32	$[-2^{31}; 2^{31}-1]$
<i>signed quadword integer</i>	8	64	$[-2^{63}; 2^{63}-1]$

Форматы некоторых чисел со знаком

ЦЕЛЫЕ ДВОИЧНЫЕ ЧИСЛА СО ЗНАКОМ

Особенности кодирования отрицательных чисел

Число со знаком:

$+A \rightarrow$ прямой код

$$[+7_{10}]_{\text{пр}} = 111_2 = \mathbf{0}000\ 0111 \text{ (в формате signed byte integer)}$$


знак числа

$-A \rightarrow$ инверсный код (как правило дополнительный код)

$$[-7_{10}]_{\text{пр}} = [-000\ 0111_2 \text{ (в формате signed byte integer)}]_{\text{доп}} = \mathbf{1}111\ 1001$$


знак числа

Самостоятельно

повторить правила получения
дополнительного кода целых
двоичных чисел.

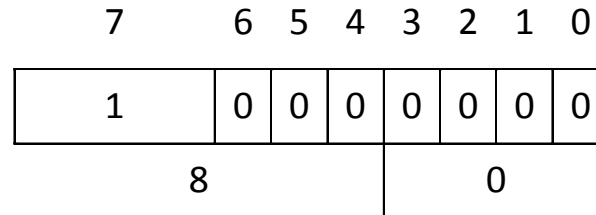
ЦЕЛЫЕ ДВОИЧНЫЕ ЧИСЛА СО ЗНАКОМ

Число Прямой двоичный код

Разряды числа в дополнительном коде

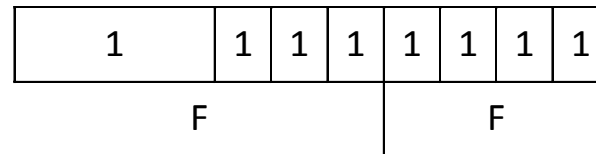
-128

~~-1~~000 0000



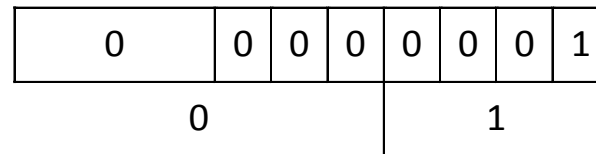
-1

- 000 0001



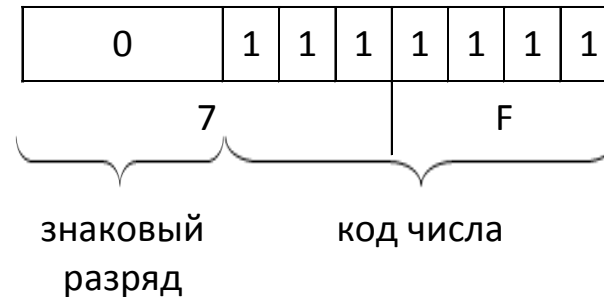
+1

+ 000 0001



+127

+ 111 1111



А так выглядят эти числа в памяти:

```
Hex dump
80 FF 01 7F
```

Самостоятельно изучить правила представления и порядок перевода целых чисел для всех размеров чисел

ЦЕЛЫЕ ДВОИЧНЫЕ ЧИСЛА

Основная сложность работы с целыми числами – возможное переполнение разрядной сетки.

Например (беззнаковые числа):

$$A = 255_{10} = 1111\ 1111_2 = FF_{16}.$$

$$B = 255_{10} = 1111\ 1111_2 = FF_{16}.$$

Должны получить:

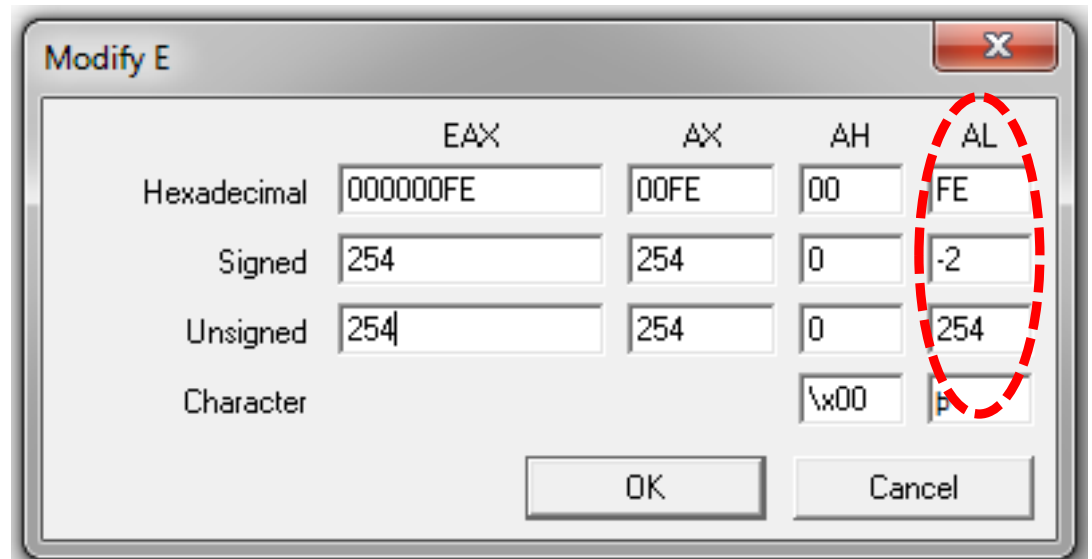
$$A+B=1FE\ (510)$$

Но после операции сложения

MOV AL,255

ADD AL,255

имеем неверный результат:



ЦЕЛЫЕ ДВОИЧНЫЕ ЧИСЛА

Основная сложность работы с целыми числами – возможное переполнение разрядной сетки.

Например (знаковые числа):

$$A = -128_{10} = 1000\ 0000_2 = 80_{16}.$$

$$B = -8_{10} = 1111\ 1000_2 = F8_{16}.$$

Должны получить:

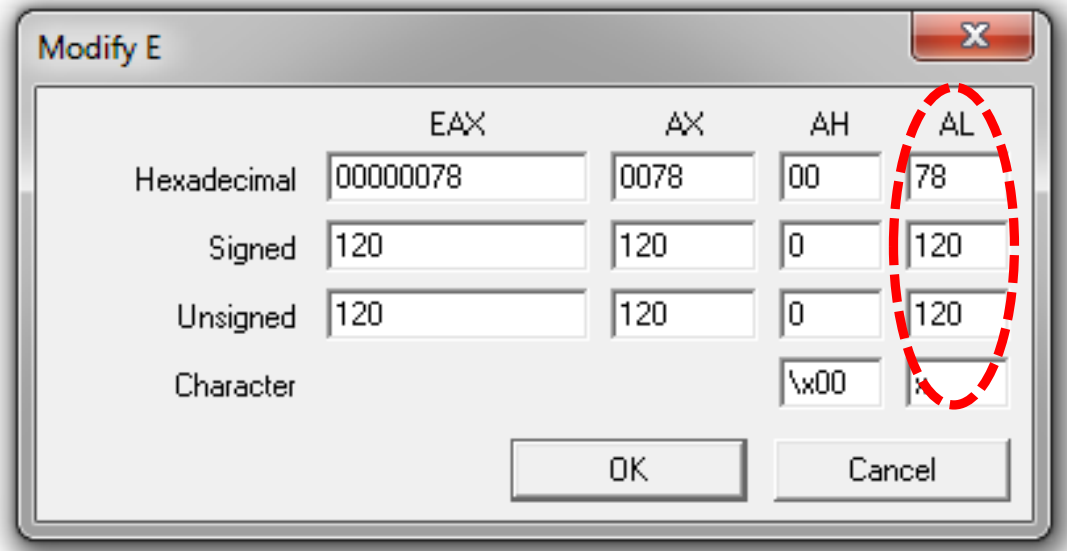
$$A+B=178_{16} (-136_{10})$$

Но после операции сложения

MOV AL, -128

ADD AL, -8

имеем неверный результат:



ДРОБНЫЕ ЧИСЛА В ФОРМАТЕ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (ЧПЗ) или FLOAT

$$A_{10} = \pm m \cdot \alpha^{\pm P},$$

где m – мантисса, записанная с основанием α ; P – порядок числа.

Формат ЧПЗ: $-273,9 = -2739 \cdot 10^{-1} = -2,739 \cdot 10^2 = -0,2739 \cdot 10^3$

Для машинной арифметики $\alpha=2$

$$A_{10} = \pm m \cdot 2^{\pm P}$$

Нормальная форма: мантисса (без учёта знака) находится на полуинтервале $[0;1[$.

$-0,2739 \times 10^3, \quad -0,02739 \times 10^4, \quad -0,002739 \times 10^5, \dots$

Нормализованная форма (мантисса десятичного числа принимает значения от 1 (включительно) до 10 (не включительно), а мантисса двоичного числа принимает значения от 1 (включительно) до 2 (не включительно)).

В такой форме любое число (кроме 0) записывается единственным образом

$$-2,739 \times 10^2.$$

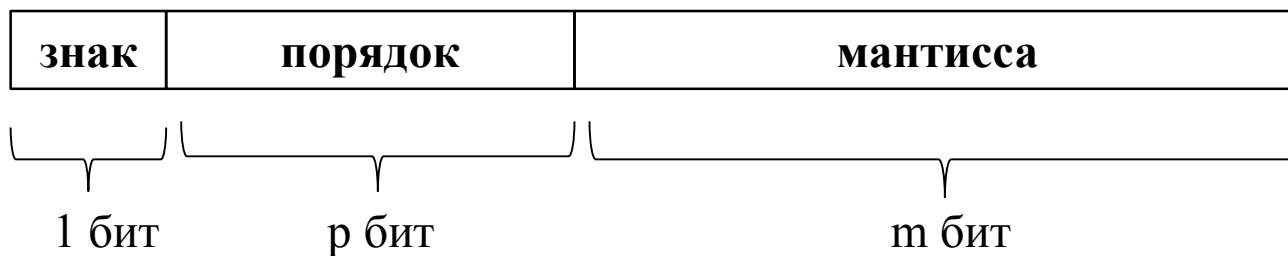
Недостаток заключается в том, что в таком виде невозможно представить 0, это особый случай машинной арифметики.

СТАНДАРТЫ ПРЕДСТАВЛЕНИЯ ЧПЗ В КОМПЬЮТЕРЕ

$$\text{Число} = \pm \text{Мантисса} \times 2^{\pm \text{порядок}}$$

Каждая ячейка памяти с числом соответствующего типа в бинарном виде содержит (последовательно):

- *1 бит знака (S),*
- *p бит, выделяемых на хранение порядка числа,*
- *m бит для хранения мантиссы.*



СТАНДАРТЫ ПРЕДСТАВЛЕНИЯ ЧПЗ В КОМПЬЮТЕРЕ

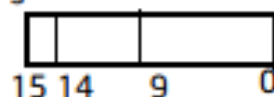
IEEE 754-2008 http://ali.ayad.free.fr/IEEE_2008.pdf

Intel

MIPS

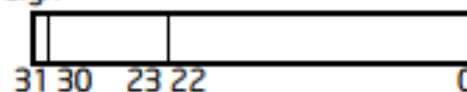
Тип данных	Размер (байт)	Точность	Разрядность (бит)			
			<i>s</i> знак	<i>p</i> порядок	<i>m</i> мантисса	всего о
binary16	2	половинная	1	5	10	16
binary32 (float)	4	одинарная	1	8	23	32
binary64 (double)	8	двойная	1	11	52	64
Не стандартизованный формат Intel	10	расширенная	1	15	64	80
binary128 (long double)	16	учетверённая	1	15	112	128

Sign



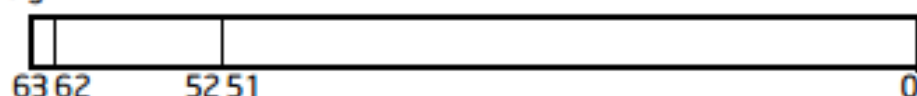
Half Precision
Floating Point

Sign



Single Precision
Floating Point

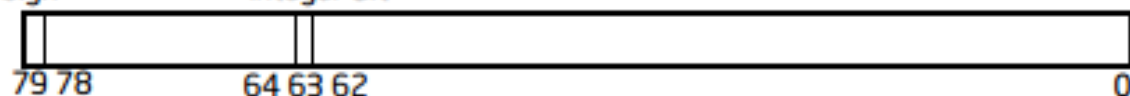
Sign



Double Precision
Floating Point

Sign

Integer Bit



Double Extended Precision
Floating Point

Пример определения ЧПЗ для Intel

Тип данных	Размер (байт)	Точность	Разрядность (бит)			
			<i>s</i> знак	<i>p</i> порядок	<i>m</i> мантисса	всего
binary16	2	половинная	1	5	10	16
binary32 (float)	4	одинарная	1	8	23	32
binary64 (double)	8	двойная	1	11	52	64
Не стандартизованный формат Intel	10	расширенная	1	15	64	80
binary128 (long double)	16	учетверённая	1	15	112	128

AA dw 2,09

A1 dd -23,056

A2 dq 11,0

A3 dt 0,88

Пример определения ЧПЗ для MIPS

Тип данных	Размер (байт)	Точность	Разрядность (бит)			
			<i>s</i> знак	<i>p</i> порядок	<i>m</i> мантисса	всего
binary16	2	половинная	1	5	10	16
binary32 (float)	4	одинарная	1	8	23	32
binary64 (double)	8	двойная	1	11	52	64
Не стандартизованный формат Intel	10	расширенная	1	15	64	80
binary128 (long double)	16	учетверённая	1	15	112	128

aa: .float 22.01

bb: .double -33.7

СТАНДАРТЫ ХРАНЕНИЯ ЧПЗ В ПАМЯТИ КОМПЬЮТЕРА

$$\text{Число} = \pm \text{Мантисса} \times 2^{\pm \text{порядок}}$$

- знак числа: 0, (+) число > 0
1, (−) число < 0
- мантисса, представлена в прямом коде; причём, как правило храниться только дробная часть мантиссы, а целая часть=1 не храниться.
- порядок храниться в смещенном виде (без знака): экспонента=порядок+смещение. Значение смещения равно $2^{p-1}-1$

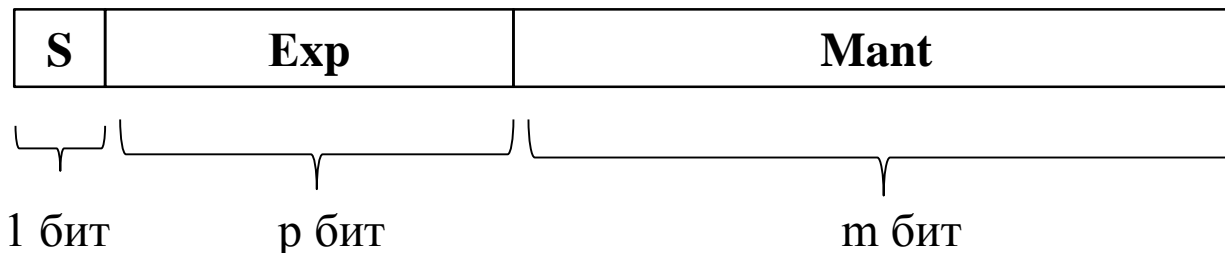
минимальный порядок $P = -\text{смещение} + 1$ (экспонента $E = 000...01$)

максимальный порядок $P = +\text{смещение}$ (экспонента $E = 11...10$)

не число (экспонента $E = 11...11$)

денормализованное число = мантисса без целой части и экспонента $E = 00...0$, но порядок считается минимальным (т.е. $-2^{p-1}-1$)

$$\text{Число ЧПЗ} = (-1)^S \times 1, M \times 2^E$$



ДИАПАЗОНЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Числа с плавающей запятой **с половинной точностью**:



S - знак	P - порядок	M - мантисса
1	5	10

Минимальное/максимальное нормализованное число =

$$\pm \text{максимальная мантисса} = 1,11\ 1111\ 1111_2$$

$$\text{максимальный порядок} = +11110_2 - (2^{p-1} - 1) = 30_{10} - 15 = 15$$

$$\text{Макс/мин.число} = \pm 1,11\ 1111\ 1111 \times 2^{15} = \pm 111\ 1111\ 1111 \times 2^5 = \pm 2047 \times 32 = \pm 65504$$

Минимальное/максимальное ненормализованное число =

$$\pm \text{минимальная мантисса} = 0,00\ 0000\ 0001_2$$

$$\text{минимальный порядок} = 00000_2 - 15 = -15$$

$$\pm 0,00\ 0000\ 0001_2 \times 2^{-15} = \pm 1,0 \times 2^{-25} = \pm 2.98 \cdot 10^{-8}$$

ДИАПАЗОНЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Числа с плавающей запятой **одинарной точности** могут хранить значение в диапазоне $[-3,37 \cdot 10^{38}; -1,17 \cdot 10^{-38}] \cup [1,17 \cdot 10^{-38}; 3,37 \cdot 10^{38}]$,

где

минимальное

максимальное

минимальное

максимальное

вещественное

вещественное

вещественное

вещественное

отрицательное число

отрицательное число

положительное число

положительное число

Для чисел с половинной точностью: $[-65504; -2,98 \cdot 10^{-8}] \cup [2,98 \cdot 10^{-8}; 65504]$.

Для чисел с двойной точностью: $[-1,8 \cdot 10^{308}; -2,2 \cdot 10^{-308}] \cup [2,2 \cdot 10^{-308}; 1,8 \cdot 10^{308}]$.

Для чисел с расшир. точностью: $[-1,18 \cdot 10^{4932}; -3,37 \cdot 10^{-4932}] \cup [3,37 \cdot 10^{-4932}; 1,18 \cdot 10^{4932}]$.

Для чисел с учетверенной точностью – рассчитать самостоятельно

Отдельно рассматривается число ноль (положительный и отрицательный).

Пример перевода дроби в формат ЧПЗ одинарной точности (из руководства Intel)

Notation	Value		
Ordinary Decimal	178.125		
Scientific Decimal	1.78125E ₁₀ 2		
Scientific Binary	1.0110010001E ₂ 111		
Scientific Binary (Biased Exponent)	1.0110010001E ₂ 10000110		
IEEE Single-Precision Format	Sign	Biased Exponent	Normalized Significand
	0	10000110	011001000100000000000000 1. (Implied)

Самостоятельно изучить правила представления и порядок перевода чисел в формате ЧПЗ

Источники

- http://ali.ayad.free.fr/IEEE_2008.pdf
- <http://www.softelectro.ru/ieee754.html>
- Материалы ЛР№3 (FPU)

ОСОБЫЕ СЛУЧАИ ЧПЗ (4 байт)

Положительный ноль	0	00...00	00...00
Отрицательный ноль	1	00...00	00...00
Наименьшее нормализованное положительное число $\approx 1,7 \cdot 10^{-38}$ (при наименее возможном значении $E=1$, т.е. $P=-126$)	0	00...01	00...00
Наименьшее денормализованное положительное число $\approx 1,4 \cdot 10^{-45}$ (без экспоненты, но при наименьшем порядке $P=-126$)	0	00...00	00...01
Наибольшее положительное число ($P=127$)	0	11...10	11...11
Наибольшее нормализованное отрицательное число $\approx 1,8 \cdot 10^{-38}$ (при наименее возможном значении $P=1$)	1	00...01	00...00
Наибольшее денормализованное отрицательное число $\approx -1,4 \cdot 10^{-45}$ (без экспоненты, $P=-126$)	0	00...00	00...01
Наименьшее отрицательное число	1	11...10	11...11
Положительная бесконечность	0	11...11	00...00
Отрицательная бесконечность	1	11...11	00...00
Не число	0	11...11	10...00
Неопределённость	1	11...11	10...00

ТОЧНОСТЬ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ В КОМПЬЮТЕРЕ

Все целые числа представлены без погрешности.

Для действительных чисел:

- вычислительная погрешность
- необходимость перевода дробей из десятичной системы счисления в двоичную.

$$1/5 = 0,2.$$

Двоичное представление:

4 разрядами после запятой - 0.0011_2 , что соответствует десятичному числу 0.1875_{10}

8 разрядами - 0.00110011_2 , что соответствует десятичному числу 0.19921875_{10}

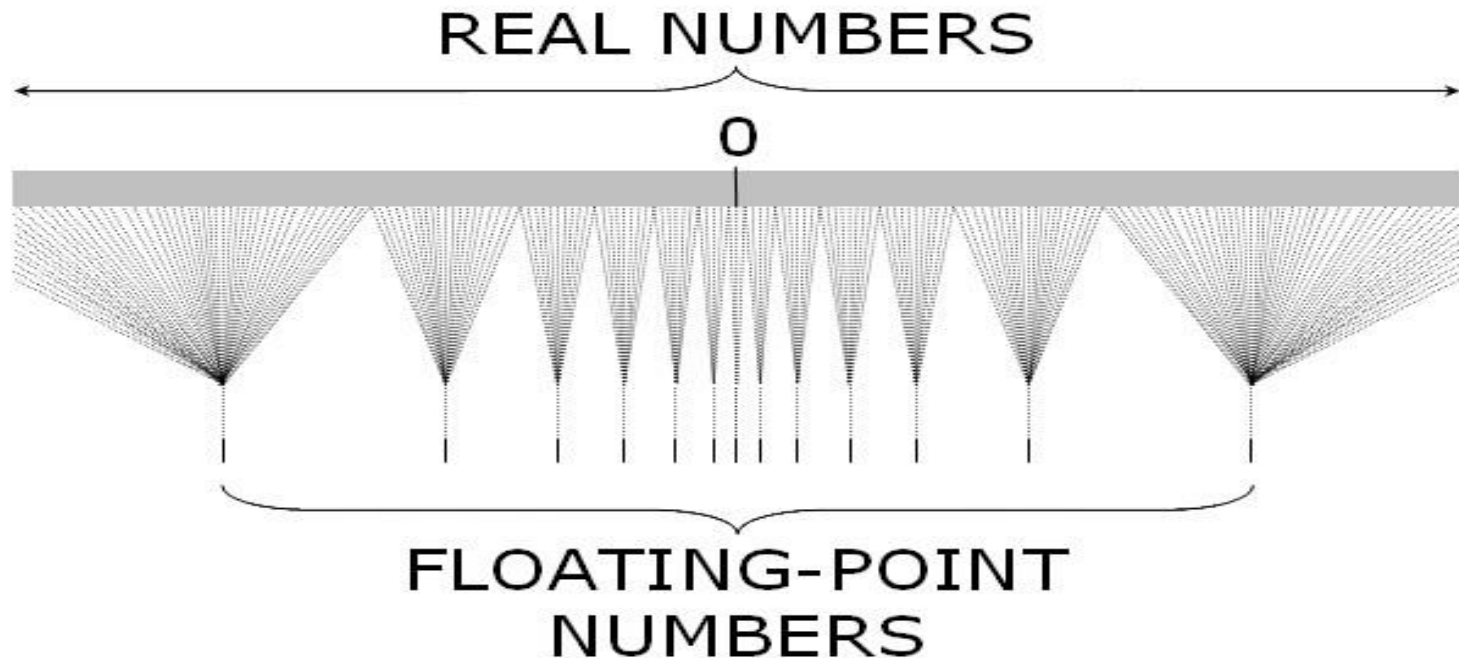
12 разрядами - 0.001100110011_2 , что соответствует десятичному числу 0.199951171875_{10}

28 разрядами - $0.0011001100110011001100110011_2$, что соответствует 0.19999999925494_{10}

Отсюда видим, что точность представления повышается с увеличением разрядной сетки. Кроме того, в формате ЧПЗ не могут быть представлены как очень большие числа, так и очень маленькие.

ДИСКРЕТНОСТЬ И НЕРАВНОМЕРНОСТЬ ПРЕДСТАВЛЕНИЯ ДРОБЕЙ В ВС

Представимые числа на числовой оси расположены дискретно и неравномерно: плотность их возрастает при приближении к нулю и падает при удалении от нуля. Это связано с изменением веса/значимости разрядов мантиссы при изменении порядка P . Чем больше значение порядка, тем больше значимость младшего разряда, равная 2^P .



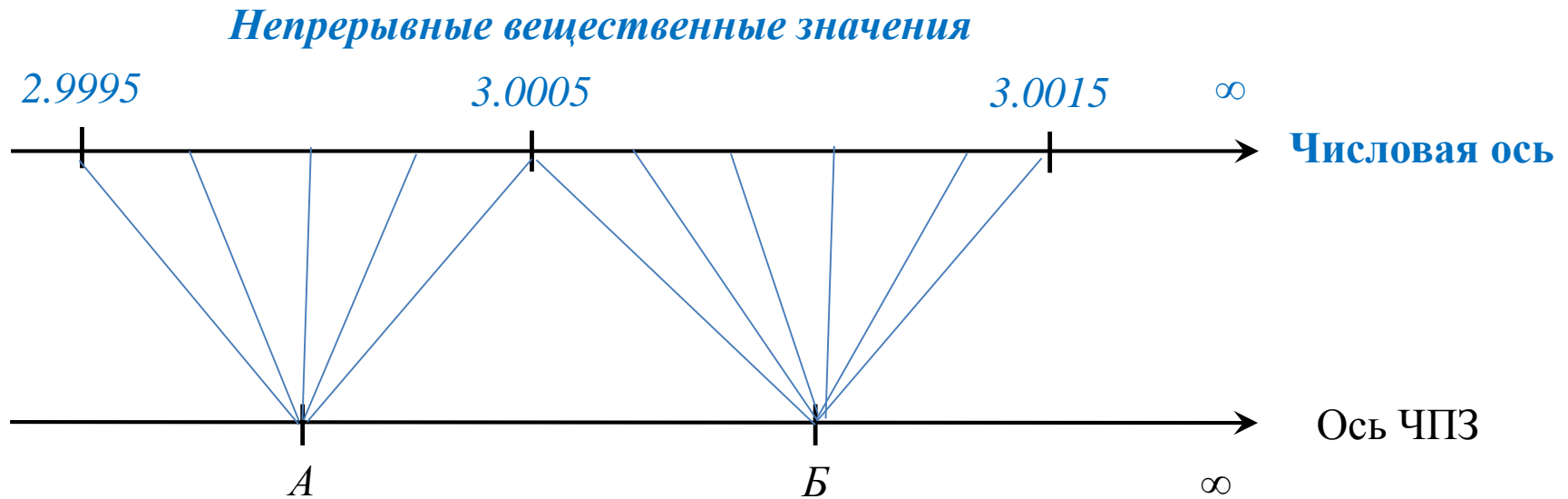
ДИСКРЕТНОСТЬ ПРЕДСТАВЛЕНИЯ ДРОБЕЙ В ВС

S - знак	P - порядок	M - мантисса
1	5	10

Рассмотрим два соседних ЧПЗ числа половинной точности:

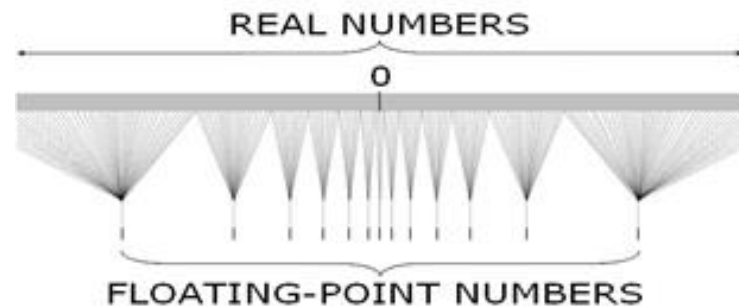
$$A = 0.01111.1000\ 0000\ 00 = 1,1 \times 2^1 = (3_{10})$$

$$B = 0.01111.1000\ 0000\ 01 = 1,10000\ 0000\ 01 \times 2^1 \approx 3.001$$

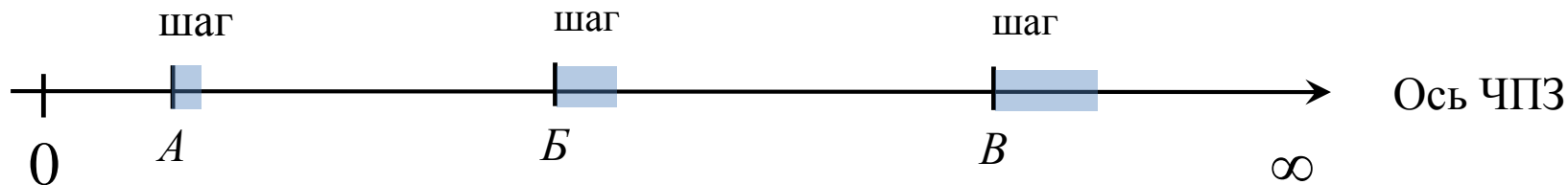


Дискретные приближенные ЧПЗ

НЕРАВНОМЕРНОСТЬ ПРЕДСТАВЛЕНИЯ ДРОБЕЙ В ВС



двоичное число	A= 0,0000 0000 01·2 ⁻¹⁵	B=1,0·2 ⁰	B=1,0·2 ⁺¹⁵									
запись в формате ЧПЗ	<table><tr><td>0</td><td>2⁻¹⁵</td><td>0,0000 ... 01</td></tr></table>	0	2 ⁻¹⁵	0,0000 ... 01	<table><tr><td>0</td><td>2⁰</td><td>1,00...00</td></tr></table>	0	2 ⁰	1,00...00	<table><tr><td>0</td><td>2¹⁵</td><td>1,00...00</td></tr></table>	0	2 ¹⁵	1,00...00
0	2 ⁻¹⁵	0,0000 ... 01										
0	2 ⁰	1,00...00										
0	2 ¹⁵	1,00...00										
ближайшее большее число	0,0000 0000 0001 0 =1,0·2 ⁻¹⁴	1,0000 0000 01 =1,0000 0000 01·2 ⁰	1,0000 0000 01 =1,0000 0000 01·2 ¹⁵									
минимальный шаг представления	2 ⁻¹⁵	0,0000 0000 01·2 ⁰ =2 ⁻¹⁰	0,0000 0000 01·2 ¹⁵ =2 ⁺⁵									



ДВОИЧНО-ДЕСЯТИЧНЫЕ ДАННЫЕ

BCD – binary coded decimal

Одна десятичная цифра BCD-числа кодируется четырьмя двоичными разрядами (тетрадой).

Десятичная цифра	BCD-код 8-4-2-1	BCD-код 2-4-2-1	BCD-код 5-4-2-1	BCD-код 8-4-2-1+3
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 0 0 1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 1
3	0 0 1 1	0 0 1 1	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1
5	0 1 0 1	0 1 0 1	1 0 0 0	1 0 0 0
6	0 1 1 0	0 1 1 0	1 0 0 1	1 0 0 1
7	0 1 1 1	0 1 1 1	1 0 1 0	1 0 1 0
8	1 0 0 0	1 1 1 0	1 0 1 1	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 0 0	1 1 0 0

Использует Intel

MIPS не работает с BCD

ДВОИЧНО-ДЕСЯТИЧНЫЕ ДАННЫЕ

BCD – binary coded decimal

10 → BCD

$$9024,19 = 1001 \text{ } 0000 \text{ } 0010 \text{ } 0100, 0001 \text{ } 1001_{\text{BCD}}$$

9 0 2 4 , 1 9

BCD → 10

$$1 \text{ } 0101 \text{ } 0011, 0111 \text{ } 1_{\text{BCD}} =$$
$$0001 \text{ } 0101 \text{ } 0011, 0111 \text{ } 1000_{\text{BCD}} = 153,78$$



Запись числа в BCD не совпадает с двоичной!

$$10101,1_{\text{BCD}} = 15,8$$

$$10101,1_2 = 16 + 4 + 1 + 0,5 = 21,5$$

ДВОИЧНО-ДЕСЯТИЧНЫЕ ДАННЫЕ

Позволяют оперировать любыми типами и размерами чисел без погрешности (т.к. это формат представления не целого числа, а лишь одного десятичного разряда).

Необходимое количество десятичных разрядов определяет пользователь при написании кода программы, т.е. он сам выбирает точность представления.

Var DB 6, 2, 4, 5, 3 ; = число 35 426

ДВОИЧНО-ДЕСЯТИЧНЫЕ ДАННЫЕ

Неупакованное BCD-число.

	игнорируются				младшая тетрада			
№ разряда	7	6	5	4	3	2	1	0
Значение разряда	Х	Х	Х	Х	1	0	0	1
	Х	Х	Х	Х	0	1	0	1
Вес разрядов					8	4	2	1

Цифра 9

Цифра 5

Var DB 6, 2, 4, 5, 3 ; = число 35 426

В памяти: 06 02 04 05 03

ДВОИЧНО-ДЕСЯТИЧНЫЕ ДАННЫЕ

Упакованное BCD-число.

	старшая тетрада				младшая тетрада				
№ разряда	7	6	5	4	3	2	1	0	
Значение разряда	0	1	0	1	1	0	0	1	Цифры 9 и 5
Вес разрядов	8	4	2	1	8	4	2	1	

Var DB 26, 12, 54, 50 ; = число 50 541 226

В памяти: 26 12 54 50

ДВОИЧНО-ДЕСЯТИЧНЫЕ ДАННЫЕ

Ошибки вычислений над BCD

$$\begin{array}{r} 7 \\ + 8 \\ \hline 15 \end{array} \quad \begin{array}{r} 0111 \\ + 1000 \\ \hline 1111 \\ + 0110 \text{ — корректирующая комбинация} \\ \hline \underline{10101} \\ \underline{1} \quad 5 \end{array}$$

Для вычислений над BCD используются обычные арифметические команды (ADD, SUB, MUL, DIV) и дополнительные команды двоично-десятичной коррекции:

AAA

DAA

AAS . . .

ВЕКТОРА (PACKED SIMD DATA TYPES)

Аппаратная поддержка *векторных операций* и *векторного типа данных* появилась в процессорах, использующих потоковое расширение архитектуры системы команд.

Intel:

- технологии MMX (**M**ulti**M**edia **E**xtensions)
- технологии SSE (англ. **S**treaming **S**IMD **E**xtensions)

одновременная обработка нескольких данных целочисленного (для MMX) или ЧПЗ (для SSE) типа за одну машинную инструкцию.

Registers (MMX)				
MM0	0105	0104	0000	0000
MM1	0000	0000	0000	0000
MM2	0000	0000	0000	0000
MM3	0000	0000	0000	0000
MM4	0000	0000	0000	0000
MM5	0000	0000	0000	0000
MM6	0000	0000	0000	0000
MM7	8000	0000	0000	0000

Registers (3DNow!)				
XMM0	00000000	00000000	00000000	00000000
XMM1	00000000	00000000	00000000	00000000
XMM2	00000000	00000000	00000000	00000000
XMM3	00000000	00000000	00000000	00000000
XMM4	00000000	00000000	00000000	00000000
XMM5	00000000	00000000	00000000	00000000
XMM6	00000000	00000000	00000000	00000000
XMM7	00000000	00000000	00000000	00000000

MIPS не работает с векторами

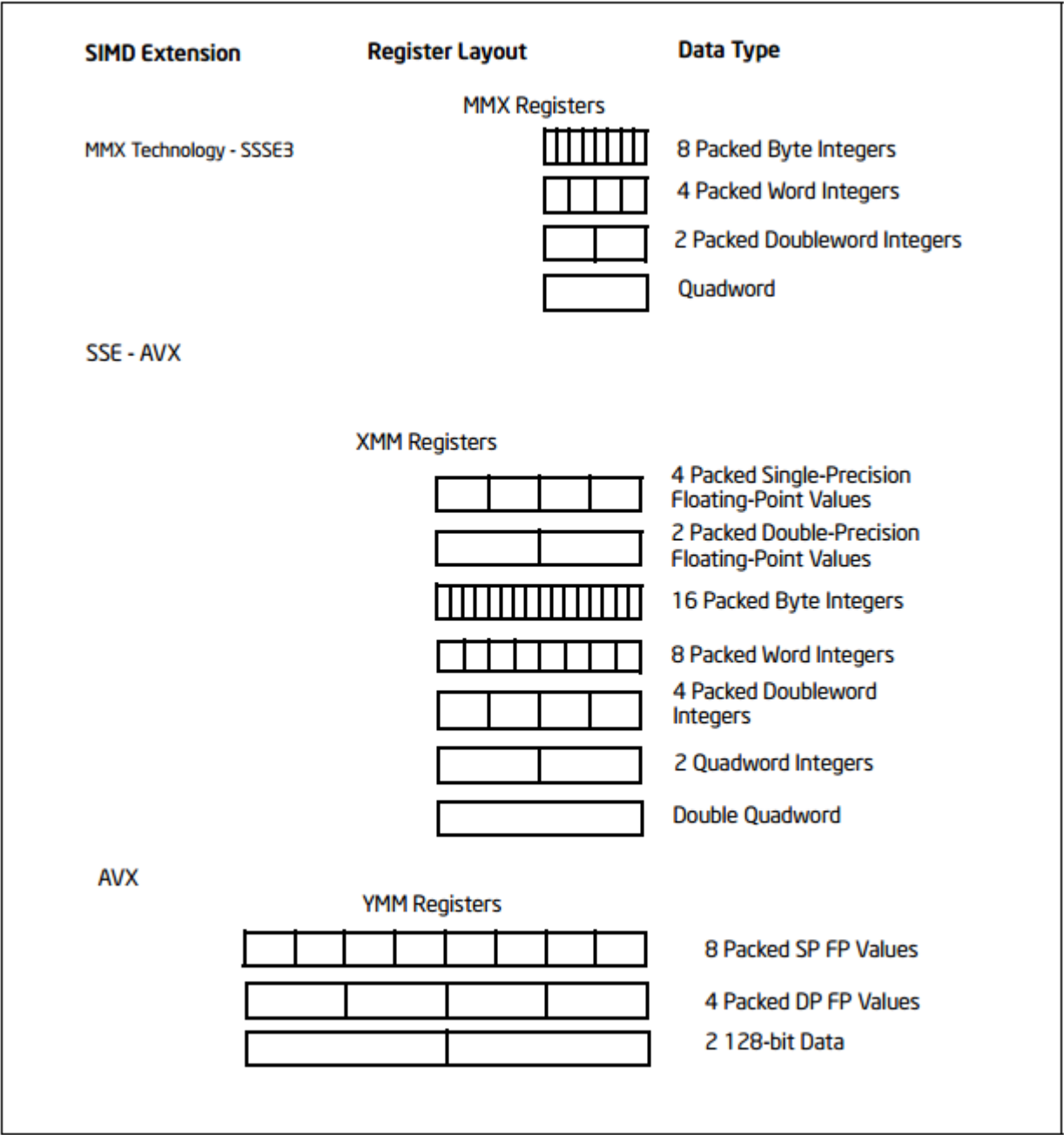


Figure 2-4. SIMD Extensions, Register Layouts, and Data Types

ДАННЫЕ ТИПА УКАЗАТЕЛЬ

= адрес фрагмента кода/данных.

Ближний указатель

Адрес внутри текущего сегмента

Дальний указатель

Ссылка на сегмент

Адрес внутри сегмента

всегда 16 бит

16-32-64 бит

ТЕГИ

может определять:

- *тип данных (целое двоичное число, число с плавающей запятой, десятичное число, адрес, строка символов, дескриптор и т.д.)*
- *формат,*
- *размер,*
- *местоположение,*
- *способы обработки/доступа*

Registers (FPU)		
ST0	valid	-1.00000000000000000000
ST1	valid	-2509847362.0000000000
ST2	empty	0.0
ST3	bad	+UNORM 1000 00000000 00000000
ST4	zero	0.0
ST5	zero	-0.0
ST6	bad	+INF 7FFF 80000000 00000000
ST7	empty	0.0

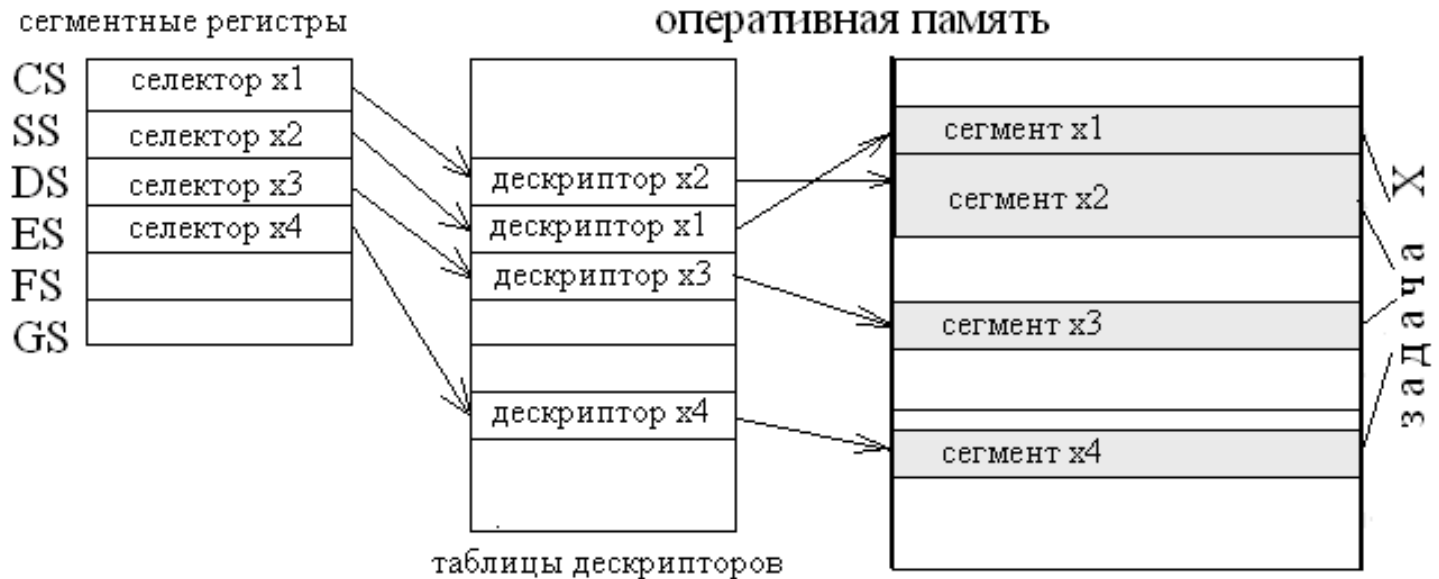


Теги регистров блока FPU

- 00 (valid) – регистр содержит конечное число (достоверное значение),
- 01 (zero) – регистр содержит нулевое значение (\pm ноль),
- 10 (bad) – не число (неопределенное значение, ненормализованное число, бесконечность, неподдерживаемый формат...),
- 11 (empty) – пустой (незаполненный) регистр.

ДЕСКРИПТОРЫ

описатели (составные из нескольких типов данных).



Например, дескриптор сегмента

- *Базовый адрес сегмента (тип – ближний указатель),*
- *Размер сегмента (тип – целое беззнаковое),*
- *Атрибуты сегмента: несколько полей разного размера и назначения.*



- Добавить про типы инструкций и режимы адресации