С точки зрения архитектуры ключевым вопросом является вопрос о том, осуществляется ли аппаратная поддержка конкретного типа данных. Под аппаратной поддержкой подразумевается, что одна или несколько команд ожидают данные в определенном формате и пользователь не может брать другой формат.

Основными типами данных являются байты, слова и двойные слова.

*Типы данных процессора Pentium 4*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тип** | **1 бит** | **8 бит.** | **16 бит.** | **32 бита** | **64 бита** | **128 бита** |
| Бит |  |  |  |  |  |  |
| Целые числа со знаком |  | \* | \* | \* |  |  |
| Целые числа без знака |  | \* | \* | \* |  |  |
| Двоично-десятичные целые числа |  | \* |  |  |  |  |
| Числа с плавающей точкой |  |  |  | \* | \* |  |

**Типы данных FPU**

7 форматов, 3 класса:

* двоичные числа – целые числа (16, 32, 64 разряда); Эти форматы существуют только в памяти, внутри FPU они автоматически преобразуются в 86-разрядный расширяемый вещественный формат
* упакованные десятичные целые числа(1 цифра = 4 разряда);
* двоичные вещественные числа – 32, 64, 80 разрядов. Значащие числа находятся в поле мантиссы, в поле порядка - положение десятичной, бит знака определяет знак числа. Порядок задается в смещенной форме = истинный порядок + смещение.

**Специальные значения:**

* Денормализованные вещественные числа
* Истинный нуль
* Бесконечность
* Нечисла
* Неподдерживаемые форматы

**ММХ** использует четыре новых [типа](http://fondatore.ru/ch43.shtml) данных:

* учетверенное [слов](http://fondatore.ru/ch02_02f.shtml)о [&mdash](http://fondatore.ru/AboutEBooka.shtml); простое 64-битное [число](http://fondatore.ru/ch11_03e.shtml);
* упакованные двойные [слова](http://fondatore.ru/ch05_04c.shtml) — два 32-битных двойных слова, упакованные в 64-[битный](http://fondatore.ru/ch10_01b.shtml) тип данных. Двойное [слово](http://fondatore.ru/ch11_01b.shtml) 1 занимает [биты](http://fondatore.ru/ch06_01.shtml) 63 – 32, и двойное слово 0 занимает биты 31 – 0;
* упакованные слова — четыре 16-битных слова, упакованные в 64-битный тип данных. Слово 3 занимает биты 63 – 48, слово 0 занимает биты 15 – 0;
* упакованные [байт](http://fondatore.ru/ch05_10h.shtml)ы — восемь байт, упакованных в 64-битный тип данных. Байт 7 занимает биты 63 – 56, байт 0 занимает биты 7 – 0.

[Команды](http://fondatore.ru/ch10_07d.shtml) ММХ перемещают упакованные [данные](http://fondatore.ru/ch05_10d.shtml) в память или в обычные [регистр](http://fondatore.ru/ch02_05c.shtml)ы как целое, но выполняют арифметические и логические [операции](http://fondatore.ru/ch01_02e.shtml) над каждым [элемент](http://fondatore.ru/ch05_07b.shtml)ом по отдельности.

Арифметические операции в ММХ [могут](http://fondatore.ru/ch67.shtml) [использовать](http://fondatore.ru/ch10_04b.shtml) специальный способ обработки переполнений и антипереполнений — *насыщение*. Если [результат](http://fondatore.ru/ch05_05b.shtml) операции больше, чем максимальное [значение](http://fondatore.ru/ch05_02a.shtml) для его типа данных (+127 для [байта](http://fondatore.ru/ch49.shtml) со знаком), то результат считают равным этому максимальному значению. Если он меньше минимального [значения](http://fondatore.ru/ch05_01b.shtml) — соответственно его полагают равным минимально допустимому значению. Например, при операциях с цветом насыщение позволяет ему превращаться в чисто белый при переполнении и в чисто черный при антипереполнении, в то [время](http://fondatore.ru/ch73.shtml) как обычная [арифметика](http://fondatore.ru/ch05_04.shtml) привела бы к нежелательной инверсии [цвета](http://fondatore.ru/ch05_10d.shtml).

**SSE**

Основной тип данных – упакованные числа с плавающей запятой одинарной точности. В одном 128-битном регистре размещаются сразу 4 таких числа: в битах 127—96 число 3, в битах 95—64 число 2, в битах 63—32 число 1, в битах 31—0 число 0. [Целочисленные команды SSE](http://osinavi.ru/asm/SSEexpansion/9.html) могут работать с упакованными байтами, словами или двойными словами. Но эти команды оперируют данными, находящимися в регистрах MMX. В командах SSE формата "команда приёмник, источник" приёмник и источник одновременно не могут быть переменными (похоже, нет такой команды во всём процессоре, которая бы могла оперировать двумя операндами из памяти за исключением тех, которые специально для этого предназначены)

