6. Команды MMX, SSE, SSE-2, SSE-3, AVX

Идея **MMX** (первоначальное название этой технологии Multi-Media eXtension было изменено фирмой INTEL на Matrix-Math eXtension) родилась как средство, решающее часть проблем с нехваткой быстродействия центрального процессора для задач мульти-медиа приложений, характеризующихся следующими факторами:

-небольшие целочисленные данные(например, 8-ми битовые графические пиксели или 16-ти битовые элементы звукового потока), которые обрабатываются в значительных объемах;

-небольшие, многократно повторяющиеся циклы;

-частые умножения и накапливания результата (выражения типа:

«a = a + b»);

-постоянные циклические обращения к памяти.

Параллельность вычислений – ключевой момент в технологии ММХ. Поэтому большинство новых ММХ-команд построены по принципу «ОДНА КОМАНДА – МНОЖЕСТВО ДАННЫХ» (SINGLE INSTRUCTION MULTIBLE DATA – SIMD). Так как процессоры PENTIUM имеют 64-х разрядную шину данных, базовая длина слова данных новых ММХ-инструкций выбрана тоже 64 разрядной.

ММХ-регистры хранят только ММХ-данные, остальные восемь РОН-ов (EAX, EBX и др.) используются для адресации памяти, работы с циклами или любых других операций.

MMX-данные хранятся в младших 64-х битах 80-ти битовых FP-

регистров (на месте мантиссы FP-данных). Поле порядка и знака (старшие 16 бит) устанавливаются в состояние «все единицы», что соответствует состоянию «минус бесконечность» или «нечисло» для FP-данных. Поэтому FP- и MMX-данные невозможно случайно неверно интерпретировать.

Таблица 2.35 – *Команды MMX.*

КОМАНДЫ ПЕРЕСЫЛКИ	
EMMS	Очистка стека регистров – установка всех единиц в слове тегов
MOVD mm, m32/ir32	Пересылка данных в младшие 32 бита регистра MMX с заполнением старших бит нулями
MOVD m32/ir32, mm	Пересылка данных из младших 32-х бит регистра MMX
MOVQ mm, mm/m64	Пересылка данных в регистр MMX
MOVQ mm/m64, mm	Пересылка данных из регистра MMX
PACKSSDW mm, mm/m64	Упаковка со знаковым насыщением двух двойных слов, расположенных в mm, и двух двойных слов mm/m64 в четыре слова, помещаемых в mm
PACKSSWB mm, mm/m64	Упаковка со знаковым насыщением четырех слов, расположенных в mm, и четырех слов mm/m64 в восемь байт, помещаемых в mm
PACKUSWB mm, mm/m64	Упаковка с насыщением четырех знаковых слов, расположенных в mm, и четырех слов mm/m64 в восемь беззнаковых байт, помещаемых в mm
PUNPCKHBW mm, mm/m64	Чередование в регистре-приемнике байтов старшей половины операнда-источника с байтами старшей половины операнда-приемника
PUNPCKHWD mm, mm/m64	Чередование в регистре-приемнике слов старшей половины операнда-источника со словами старшей половины операнда-приемника
PUNPCKHDQ mm, mm/m64	Чередование в регистре-приемнике двойного слова старшей половины операнда-источника с двойным словом старшей половины операнда-приемника
PUNPCKLBW mm, mm/m64	Чередование в регистре-приемнике байтов младшей половины операнда-источника с байтами младшей половины операнда-приемника
PUNPCKLWD mm, mm/m64	Чередование в регистре-приемнике слов младшей половины операнда-источника со словами младшей половины операнда-приемника
PUNPCKLDQ mm, mm/m64	Чередование в регистре-приемнике двойного слова младшей половины операнда-источника с двойным словом младшей половины операнда-приемника

АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ КОМАНДЫ	
PADDB mm, mm/m64 PADDW mm, mm/m64 PADDD mm, mm/m64	Сложение упакованных байт (слов или двойных слов) без насыщения (с циклическим переполнением)
PADDSB mm, mm/m64 PADDSW mm, mm/m64	Сложение упакованных байт (слов) со знаковым насыщением
PADDUSB mm, mm/m64 PADDUSW mm,mm/m64	Сложение упакованных байт (слов) с беззнаковым насыщением
PSUBB mm, mm/m64 PSUBW mm, mm/m64 PSUBD mm, mm/m64	Вычитание упакованных байт (слов, двойных слов) без насыщения (с циклическим переполнением)
PSUDSB mm, mm/m64 PSUDSW mm, mm/m64	Вычитание упакованных знаковых байт (слов) с насыщением
PSUBUSB mm, mm/m64 PSUBUSW mm,mm/m64	Вычитание упакованных беззнаковых байт (слов) с насыщением
PMULHW mm, mm/m64	Умножение упакованных знаковых слов с сохранением только старших 16 бит элементов результата
PMULLW mm, mm/m64	Умножение упакованных знаковых или беззнаковых слов с сохранением только младших 16 бит результата
PMADDWD mm, mm/m64	Умножение четырех знаковых слов операнда-источника на четыре знаковых слова операнда приемника. Два двойных слова результатов умножения младших слов суммируются и записываются в младшие 32 бита операнда-приемника. Два двойных слова результатов умножения старших слов суммируются и записываются в старшие 32 бита операнда-приемника
PAND mm, mm/m64	Логическое "И"
PANDN mm, mm/m64	Логическое "И-НЕ"
POR mm, mm/m64	Логическое "ИЛИ"
PXOR mm, mm/m64	Исключающее "ИЛИ"
PCMPEQB mm,mm/m64 PCMPEQW mm,mm/m64 PCMPEQD mm,mm/m64	Сравнение (на равенство) упакованных байт (слов, двойных слов). Все биты элемента результата будут единичными (True) при равенстве соответствующих элементов операндов и – нулевыми (False) при неравенстве
PCMPGTB mm,mm/m64 PCMPGTW mm,mm/m64 PCMPGTD mm,mm/m64	Сравнение (по величине) упакованных знаковых байт (слов, двойных слов). Все биты элемента результата будут единичными (True), если соответствующий элемент операнда-получателя больше элемента операнда-источника, и нуле-

SSE. Основное отличие новой технологии «MMX-2», так называемое, потоковое расширение – SSE (Streaming SIMD Extensions), получившей офи-

циальное название «Новый набор инструкций KATMAI» – KNI (Katmai New Instruction), – использование параллельных вычислений над числами с плавающей запятой одинарной точности (SIMD-FP, Single Instruction Multiple Data — Floating Point). В добавление к 57 MMX-командам процессор PENTIUM-3 (более раннее название – KATMAI) имеет еще 70 команд, ориентированных на SIMD-FP (направленных на обработку 3D-приложений), а также дополнительные целочисленные инструкции с регистрами MMX и команды управления кэшированием. Добавлены специализированные команды для ускорения трансформации трехмерных объектов, эффектов освещения, атмосферных эффектов и др.

Расширение SSE-2. В новом процессоре PENTIUM-4 (2000 г.) основные нововведения направлены на ускорение обработки потоковых данных, что обеспечивает максимальную производительность при обработке видео, графических изображений, работе с мультимедиа и в других сложных задачах. Поток в данном контексте подразумевает, что с его данными должны выполняться однотипные операции. Кроме того, данные, уже прошедшие обработку, в дальнейшем этим вычислительным процессом использоваться не будут и ими не следует засорять КЭШ.

Добавлены дополнительные 144 команды (SSE-2), ускоряющие работу широкого спектра потоковых ресурсоемких приложений.

В процессоре PENTIUM-4 добавлены новые форматы данных:

-упакованная пара чисел с плавающей точкой двойной точности (2 * 64 = 128 бит);

-упакованные целые числа: 16 байт (16 * 8 = 128 бит), 8 слов (8 * 16 = 128 бит), 4 двойных слова (4 * 32 = 128 бит) и 2 счетверенных слова (2 * 64 = 128 бит).

Расширение SSE-3. В процессор PENTIUM-4 на ядре PRESCOTT (который появился в конце 2003 г.) добавлены еще 13 новых команд SIMD-FP, получившие название SSE-3.

Advanced Vector Extensions (AVX) – расширение системы команд x86 для микропроцессоров Intel и AMD, предложенное Intel в марте 2008.

Новые возможности AVX:

-Новая схема кодирования инструкций VEX;

-Ширина векторных регистров SIMD увеличивается со 128 (XMM) до 256 бит (регистры YMM0 – YMM15). Существующие 128-битовые SSE инструкции будут использовать младшую половину новых YMM регистров, не изменяя старшую часть. Для работы с YMM регистрами добавлены новые 256-битовые AVX инструкции. В будущем возможно расширение векторных регистров SIMD до 512 или 1024 бит. Например,

процессоры с архитектурой Larrabee уже имеют векторные регистры (ZMM) шириной в 512 бит, и используют для работы с ними SIMD команды с MVEX и VEX префиксами, но при этом они не поддерживают AVX;
-Неразрушающие операции. Набор AVX инструкций использует трёхоперандный синтаксис. Например, вместо a = a + b можно использовать c = a + b, при этом регистр a остаётся неизменённым. В случаях, когда значение a используется дальше в вычислениях, это повышает производительность, так как избавляет от необходимости сохранять перед вычислением и восстанавливать после вычисления регистр, содержавший a, из другого регистра или памяти;
-Для большинства новых инструкций отсутствуют требования к выравниванию операндов в памяти. Однако, рекомендуется следить за выравниванием на размер операнда, во избежание значительного снижения производительности.
-Набор инструкций AVX содержит в себе аналоги 128-битовых SSE инструкций для вещественных чисел. При этом, в отличие от оригиналов, сохранение 128-битового результата будет обнулять старшую половину YMM регистра. 128-битовые AVX инструкции сохраняют прочие преимущества AVX, такие как новая схема кодирования, трехоперандный синтаксис и невыровненный доступ к памяти. Рекомендуется отказаться от старых SSE и MMX инструкций в пользу новых 128-битовых AVX инструкций, даже если достаточно двух операндов.