CISC

x86

MMX (Multimedia Extensions)

Расширение базового набора команд IA-32, впервые появившееся в микропроцессорах компании Intel на базе микроахитектуры P5 , "Pentium with MMX Technology" в 1996 году. Направлено на повышение эффективности обработки звука и видео. Технология интересна объединением различных подходов: фактически Pentium MMX представляет собой классический процессор Pentium (CISC) с добавлением набора команд параллельной обработки данных (SIMD).

Основные нововведения:

Восемь новых 64-битных регистров (MMX регистры)

Три новых упакованных типа данных:

- 64-битные упакованные байты (целые знаковые и беззнаковые)

- 64-битные упакованные слова (целые знаковые и беззнаковые)

- 64-битные упакованные двойные слова (целые знаковые и беззнаковые)

Инструкции для работы с новыми типами и управления состоянием MMX.

Регистры

Существует два способа доступа к регистрам MMX: 64-битный режим и 32-битный.

64-битный режим используется для:

- 64-битного доступа к памяти

- перемещения информации между MMX регистрами

- всех логических арифметических инструкций и инструкций упаковывания

- некоторых инструкций распаковки

32-битный режим используется для:

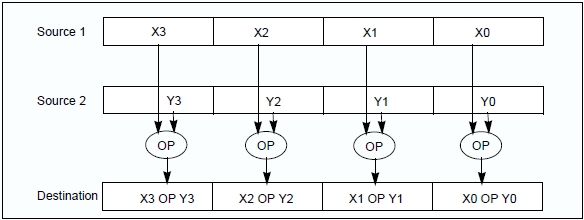
- 32-битного доступа к памяти

- перемещения информации между регистрами общего назначения и MMX регистрами

- некоторых инструкций распаковки

Модель исполнения SIMD

Новые команды и типы данных позволяют увеличить производительность программ за счёт возможности параллельного исполнения нескольких арифметических или логических инструкций параллельно.



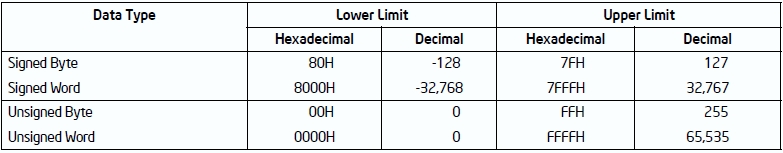
Модель исполнения SIMD

Режимы обработки переполнения.

Технология MMX предоставляет три режима обработки переполнения

- Циклическая (wraparound) арифметика, при использовании которой в случае переполнения в результат попадают только младшие значащие биты.

- Арифметика с насыщением (saturation): знаковая и беззнаковая, когда в случае переполнения в результат попадает наибольшее (либо наименьше) число, которое может быть представлено в конкретном типе.



Границы некоторых типов при насыщении

Новые инструкции

Для работы с новыми типами данных в MMX введены 47 инструкций, которые можно разделить на следующие логические блоки:

-Перемещение данных

-Арифметика

- Сравнение

- Конвертирование

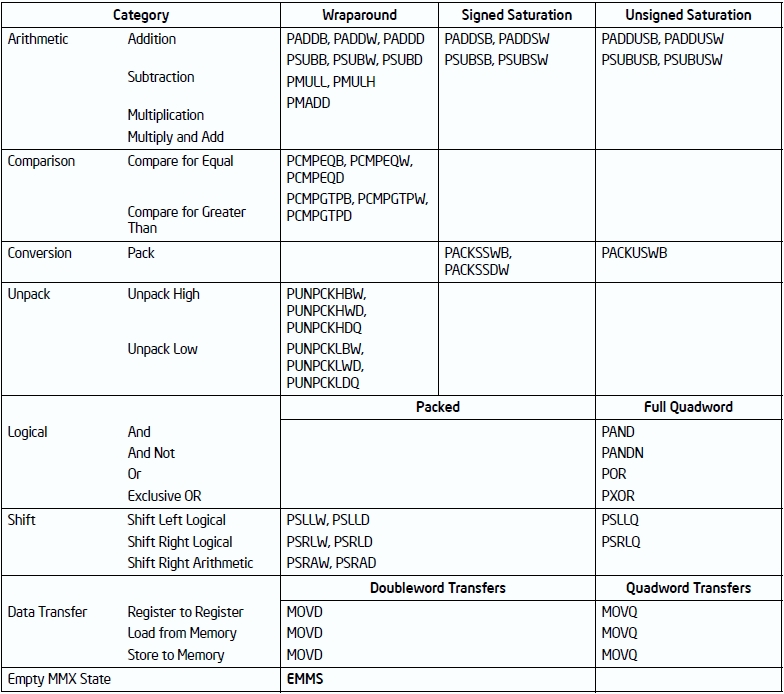
- Распаковывание

- Логические операции

- Сдвиги

- Empty MMX state instruction

Ниже приведена таблица с кратким описанием новых команд.

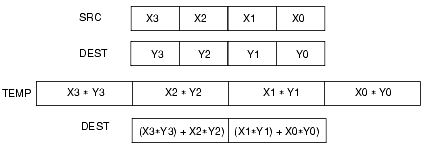


Во многих командах присутствует суффикс, определяющий тип данных и используемую арифметику:

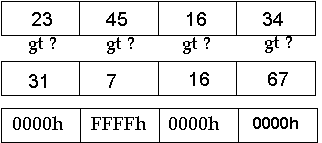
- US, S(S) указывают на использование знаковой и беззнаковой арифметики с насыщением

- B, W, D, Q указывают типы данных

Отдельно следует выделить команду PMADD, которая позволяет вычислять произведение комплексных чисел, компоненты которых представляют собой целочисленное слово (используется циклическая арифметика)



Операции сравнения так же выполняются параллельно и записывают в результат последовательность из нулей в случае выполнения сравнения и последовательность единиц в обратном случае. (Длины последовательностей зависят от используемого типа данных, к примеру, для слова они составят 16 бит.)



Одной из важных особенностей MMX является совпадение регистров MMX и регистров математического сопроцессора. При выполнении любой из MMX-команд происходит установка «режима MMX» с отметкой этого в слове состояния сопроцессора (FPU Tag Word). С этого момента стек регистров сопроцессора рассматривается как набор MMX-регистров; завершает работу в режиме MMX команда EMMS (End MultiMedia State). Такая реализация даёт обратную совместимость, но вместе с тем исключает возможность одновременного использования инструкций сопроцессора и MMX. Кроме того переход между режимами занимает значительное время, что может при неоптимальном использовании ощутимо замедлить выполнение программы.

Согласно данным Intel, микропроцессор с поддержкой MMX ускоряет приложения мультимедиа до 60%. В остальных приложениях ускорение составляет порядка 10%. В первую очередь это связано с применением SIMD инструкций.

Примеры использования.

Умножение комплексного числа на комплексную константу

Вход: MM0: компоненты комплексного числа Dr, Di в двух младших словах

MM1: компоненты комплексной константы, расположенные следующим образом

[Cr, -Ci, Ci, Cr]

Выход: MM0: две 32-битные компоненты произведения [Pr, Pi]

PUNPCKLDQ MM0, MM0 ; Копирует два младших слова MM0 в два старших

PMADDWD MM0, MM1 ; Считает два скалярных произведение векторов из двух компонент

;[(Dr\*Cr-Di\*Ci), (Dr\*Ci+Di\*Cr)]

Наложение картинки ведущего прогноза погоды на фон



Movq mm3,mem1 ; загрузка восьми пикселей из картинки с ведущим

Movq mm4,mem2 ; загрузка восьми пикселе из картинки с картой

Pcmpeqb mm1,mm3

Pand mm4,mm1

Pandn mm1,mm3

Por mm4,mm1

PCMPEQ применяется к изображению ведущего и синему экрану и возвращает битовую маску силуэта ведущего.



PAND применяется к изображению карты и оставляет силуэт ведущего (на самом деле чёрный)



PANDN применяет инвертированную битовую маску к изображению ведущего, удаляя синий фон



POR совмещает две полученные картинки и получает необходимое наложение



SSE (Streaming SIMD Extensions)

Второе расширение набора команд микроархитектуры x86 от Intel, представленное в 1999 в линейке процессоров Pentium III (Ядро Katmai), как ответ на аналогичное расширение от AMD 3DNow!. Новая технология позволила преодолеть две основные проблемы MMX – невозможность одновременного использования MMX и инструкции сопроцессора и возможность MMX работать только с целыми числами.

Основные нововведения:

Восемь (в 64-битном режиме шестнадцать) 128-битных регистров XMM

32-битный регистр MXCSR, предоставляющий биты контроля и статуса для операций, производимых в XMM регистрах

128-битный тип упакованных чисел с плавающей точкой одинарной точности (4x32 бит)

70 новых инструкций для SIMD-операций с упакованными числами с плавающей точкой и целыми, для сохранения и восстановления регистра MXCSR, инструкции для явной предвыборки данных и управления кэшем

Инструкции для работы с числами делятся на два типа:

Single scalar (суффикс ss), которые оперируют с 32 младшими битами

Parallel scalar (суффикс ps), которые оперируют со всеми четырьмя частями регистра параллельно

Кроме очевидных преимуществ при однотипной обработке больших массивов данных, в некоторых случаях могут оказаться полезными операции для работы с кэшем. В качестве примера можно привести оптимизацию функции копирования памяти.

mov esi, [src] // source array

mov edi, [dst] // destination array

mov ecx, [len] // number of QWORDS (8 bytes)

lea esi, [esi+ecx\*8]

lea edi, [edi+ecx\*8]

neg ecx

emms

copyloop:

prefetchnta [esi+ecx\*8 + 512]

movq mm0, qword ptr [esi+ecx\*8]

movq mm1, qword ptr [esi+ecx\*8+8]

movq mm2, qword ptr [esi+ecx\*8+16]

movq mm3, qword ptr [esi+ecx\*8+24]

movq mm4, qword ptr [esi+ecx\*8+32]

movq mm5, qword ptr [esi+ecx\*8+40]

movq mm6, qword ptr [esi+ecx\*8+48]

movq mm7, qword ptr [esi+ecx\*8+56]

movntq qword ptr [edi+ecx\*8], mm0

movntq qword ptr [edi+ecx\*8+8], mm1

movntq qword ptr [edi+ecx\*8+16], mm2

movntq qword ptr [edi+ecx\*8+24], mm3

movntq qword ptr [edi+ecx\*8+32], mm4

movntq qword ptr [edi+ecx\*8+40], mm5

movntq qword ptr [edi+ecx\*8+48], mm6

movntq qword ptr [edi+ecx\*8+56], mm7

add ecx, 8

jnz copyloop

sfence

emms

Основные идеи:

Использовать MMX регистры для копирования больших последовательностей данных

Ускорять чтение, записывая очередной блок данных в кэш инструкцией preftchnta

Ускорять запись, используя инструкцию movntq, которая позволяет писать данные напрямую в память, минуя кэш.

При использовании такой функции скорость копирования увеличивается почти на 80% (по сравнению со стандартным memcpy)

SSE2

Очередное SIMD-расширение от Intel, впервые появившееся в процессорах Pentium 4 в 2001 году. SSE2 расширяет более ранний набор инструкций SSE и вводится с целью вытеснения MMX.

Основные нововведения:

Шесть новых типов данных: 4 запакованных 128-битных целочисленных типа и 128-битные запакованные числа с плавающей точкой двойной точности

Команды для поддержки новых типов и новые версии MMX инструкций (+ расширение возможностей обработки арифметических ошибок)

Дополнительные команды для работы с кэшем

Сложные команды конвертирования **ч**исел.

SSE3

Третья версия SIMD-расширения Intel, впервые представлено в 2004 году в ядре Prescott процессоре Pentium 4. SSE3 содержит 13 новых инструкций. Наиболее заметным изменением стало появление ассиметричных и горизонтальных инструкций.

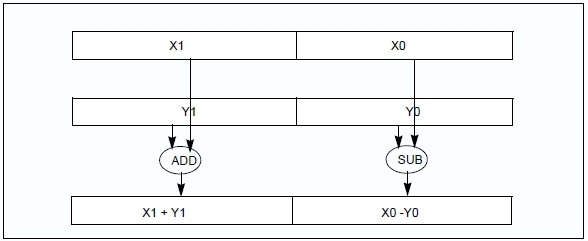


Схема выполнения инструкции ADDSUBPD

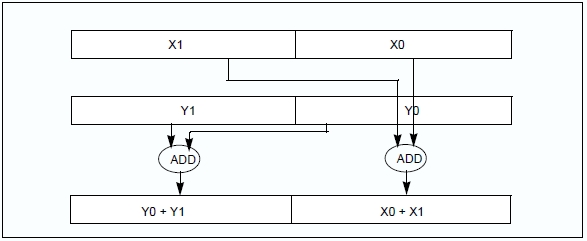


Схема выполнения инструкции HADDPD

Кроме того появилось несколько инструкций, улучшающих процессы перемещения чисел с плавающей точкой, и две инструкции для синхронизации потоков - MONITOR и MWAIT.

SSSE3

Дополнение технологии SSE3, появившееся в 2006 году. Включает 16 команд для работы с упакованными целыми (знаковыми и беззнаковыми). Стоит отметить добавление нескольких новых инструкций горизонтальных операций и инструкций вычисления абсолютного значения.

SSE4

Новый набор команд микроархитектуры Intel Core, впервые появившийся в 2007 году в процессорах серии Penryn. SSE4 состоит из 54 инструкций, 47 из них относят к SSE4.1 (они есть в процессорах Penryn). Полный набор команд (SSE4.1 и SSE4.2, то есть 47 + оставшиеся 7 команд) доступен в процессорах Intel с микроархитектурой Nehalem, которые были выпущены в середине ноября 2008 года и более поздних редакциях. Краткое описание некоторых новых инструкций можно увидеть в таблице ниже

|  |  |
| --- | --- |
| **Инструкция** | **Описание** |
| MPSADBW | Вычисляет восемь смещённых сумм абсолютных разностей, четыре за раз. (, |x0−y0|+|x1−y1|+|x2−y2|+|x3−y3|, |x0−y1|+|x1−y2|+|x2−y3|+|x3−y4|, …, |x0−y7|+|x1−y8|+|x2−y9|+|x3−y10|). Эта операция важна для некоторых кодеков, и позволяет считать разность блоков 8×8 меньше, чем за семь циклов. |
| PHMINPOSUW | В два младших слова выхода пишет соответственно минимальное слово из входа и его индекс. |
| DPPS, DPPD | Вычисление скалярных произведений. |
| PMINSB, PMAXSB, PMINUW,  PMAXUW, PMINUD, PMAXUD,  PMINSD, PMAXSD | Вычисление минимумов и максимумов для различных упакованных типов. |
| ROUNDPS, ROUNDSS, ROUNDPD, ROUNDSD | Различные типы округления чисел с плавающей точкой до целого. |
| INSERTPS, PINSRB, PINSRD/PINSRQ,  EXTRACTPS, PEXTRB, PEXTRD/PEXTRQ | Инструкции, упрощающие обмен данными между памятью, регистрами общего назначения и XMM. |
| PMOVSXBW, PMOVZXBW, PMOVSXBD,  PMOVZXBD, PMOVSXBQ, PMOVZXBQ,  PMOVSXWD, PMOVZXWD, PMOVSXWQ,  PMOVZXWQ, PMOVSXDQ, PMOVZXDQ | Преобразование упакованного целочисленного значения в целочисленное значение более широкого типа. |
| PCMPEQQ | Сравнение qword на равенство |
| PACKUSDW | Convert signed DWORDs into unsigned WORDs with saturation. |
| MOVNTDQA | Efficient read from write-combining memory area into SSE register; this is useful for retrieving results from peripherals attached to the memory bus. |