КОНСТРУИРОВАНИЕ ПРОГРАММ

Лекция № 19

Арифметические расширения команд процессора. Технология SSE2/SSE3

+375 17 293 8039 (505a-5)

+375 17 320 7402 (ОИПИ НАНБ)

prep@lsi.bas-net.by

ftp://student:2ok*uK2@Rwox@lsi.bas-net.by/

Кафедра ЭВМ, 2022

2022.05.13

Оглавление

Технология SSE2

Технология **SSE2** (**S**treaming **S**IMD **E**xtensions **2**) разработана для применения в процессорах Intel Pentium 4.

Ее назначение — **повысить эффективность операций со 128-разрядными данными в формате плавающей точки с двойной точностью** и с целочисленными данными.

Позволяет разрабатывать высокопроизводительные приложения для 3D-графики и 3D-геометрии, моделирования и симуляции процессов, обработки сигналов, 3D-анимации, кодирования/декодирования, распознавания речи и т. д.

SSE2 расширяет возможности MMX за счет использования 128-разрядных регистров вместо 64-разрядных, обеспечивая большую эффективность параллельных вычислений.

В SSE2 используются новые типы данных:

- 128-разрядные операнды с плавающей точкой двойной точности;
- 128-разрядные упакованные целые числа.

Технология SSE2 позволяет улучшить вычислительные возможности благодаря:

- улучшению управления данными в кэше;
- повышению производительности операций, требующих более высокой точности;
- расширению до 128 бит диапазона обрабатываемых 64-разрядными командами операндов.

Новые функции, предоставляемые расширениями SSE2, расширяют модель программирования SIMD архитектуры IA-32 по трем важным направлениям:

1) обеспечивают возможность выполнения SIMD-операций над парами упакованных значений двойной точности с плавающей запятой.

2) позволяют выполнять более точные вычисления в регистрах XMM, что повышает производительность процессора в научных и инженерных приложениях, а также в приложениях, использующих передовые методы 3-D геометрии (такие как трассировка лучей). Дополнительная гибкость обеспечивается инструкциями, которые работают с одинарными (скалярными) значениями с плавающей запятой двойной точности, расположенными в младшем квадрослове регистра XMM.

3) обеспечивают возможность работы со 128-битными упакованными целыми числами (байты, слова, двойные слова и четверные слова) в регистрах XMM. Это обеспечивает большую гибкость и большую пропускную способность при выполнении SIMD-операций над упакованными целыми числами. Данная возможность особенно полезна для таких приложений, как RSA-аутентификация и шифрование RC5. Используя полный набор SIMD-регистров, типов данных и инструкций, предоставляемых технологией MMX и расширениями SSE/SSE2, программисты могут разрабатывать алгоритмы, точно сочетающие упакованные данные с плавающей запятой одинарной и двойной точности и 64- и 128-битные упакованные целочисленные данные.

Кеширование

Расширения SSE2 улучшают поддержку, представленную в расширениях SSE, для управления возможностью кэширования SIMD-данных. Инструкции по управлению кешем SSE2 обеспечивают возможность потоковой передачи данных в регистры XMM и из них без загрязнения кешей, а также возможность предварительной выборки данных до их фактического использования.

Совместимость расширений SSE/SSE2/SSE3/MMX и x87 FPU

Регистры XMM и регистр MXCSR, введенные в среду исполнения IA-32 в рамках расширения SSE, совместно используется с расширениями SSE2 и SSE3.

Расширения SSE2 полностью совместимы со всем программным обеспечением, написанным для процессоров IA-32. Все существующее программное обеспечение продолжает работать правильно, без изменений, на процессорах, включающих расширения SSE2, а также при наличии приложений, включающих эти расширения.

Инструкции SSE/SSE2/SSE3 полностью совместимы — они могут выполняться вместе в одном потоке команд без необходимости сохранять состояние при переключении между наборами команд.

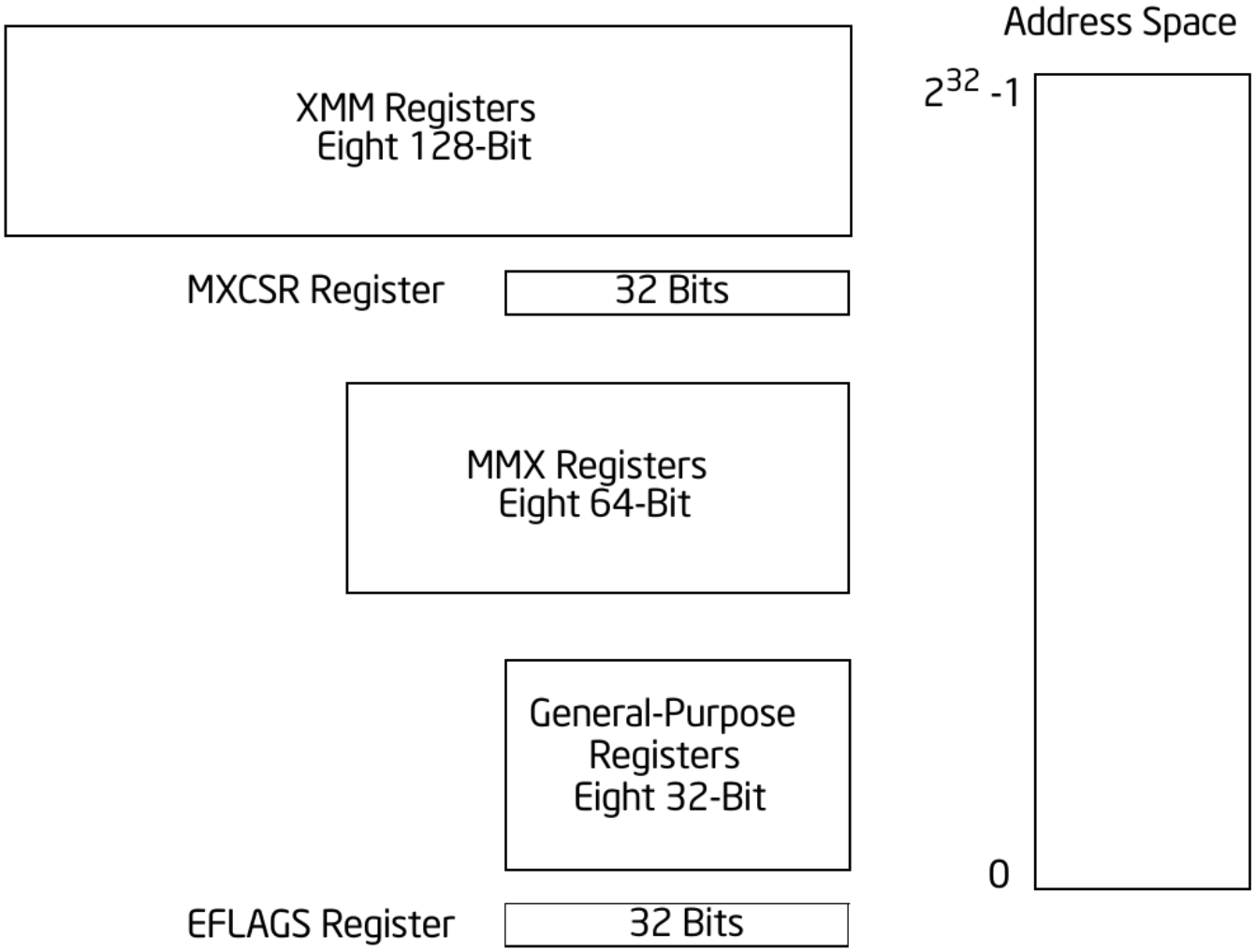
Усовершенствования инструкции CPUID позволяют обнаруживать расширения SSE2.

Регистры XMM не зависят от регистров x87 FPU и MMX, поэтому операции SSE/SSE2/SSE3, выполняемые с регистрами XMM, могут выполняться параллельно с операциями над регистрами x87 FPU и MMX.

Кроме того, поскольку расширения SSE2 используют те же регистры, что и расширения SSE, для сохранения и восстановления состояния программы во время переключения контекста не требуется никакой новой поддержки операционной системы, помимо той, которая предусмотрена для расширений SSE — инструкции FXSAVE и FXRSTOR сохраняют и восстанавливают состояния расширений SSE/SSE2/SSE3 вместе с состояниями x87 FPU и MMX.

Расширения SSE2 доступны из всех режимов выполнения IA-32 — защищенный режим, режим реального адреса, виртуальный режим 8086.

Среда программирования SSE2



Все инструкции SSE работают с XMM регистрами, регистрами MMX и/или памятью:

Регистры XMM — эти восемь регистров используются для работы с упакованными или скалярными данными с плавающей точкой одинарной точности.

Скалярные операции — это операции, выполняемые с отдельными (неупакованными) значениями с плавающей запятой одинарной точности, хранящиеся в младшем двойном слове регистра XMM. На регистры XMM можно ссылаться по именам **XMM0 ... XMM7**.

Данные могут загружаться в регистры XMM или записываться из регистров в память с шагом 32, 64 и 128 бит. При хранении всего содержимого регистра XMM в памяти (128-бит) данные сохраняются в 16 последовательных байтах, причем младший байт регистра хранится в первом байте в памяти.

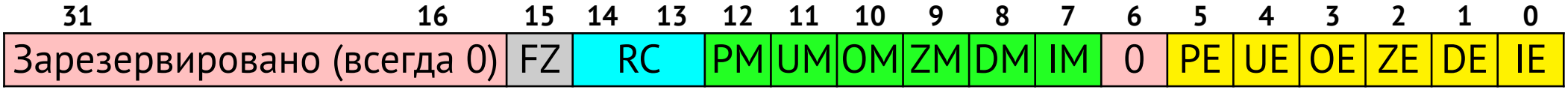
Регистр MXCSR — этот 32-битный регистр содержит биты состояния и управления, используемые в операциях с плавающей запятой SIMD.

MMX регистры — восемь регистров используются для выполнения операций над 64-битными упакованными целочисленными данными. Они также используются для хранения операндов для некоторых операций, выполняемых между регистрами MMX и XMM. На регистры MMX можно ссылаться по именам **MM0 ... MM7**.

Регистры общего назначения — восемь регистров общего назначения (EAX, EBX, ECX, EDX, EBP, ESI, EDI и ESP) используются вместе с существующими режимами адресации IA-32 для адресации операндов в памяти. Регистры MMX и XMM не могут использоваться для адресации памяти. РОНтакже используются для хранения операндов для некоторых инструкций SSE.

Регистр EFLAGS — 32-битный регистр используется для записи результатов некоторых операций сравнения.

Регистр управления/состояния — MXCSR (32 бит)



Биты 0..5 — флаги исключений:

IE (0) — запрошена невыполнимая операция (команда).

DE (1) — денормализованный операнд — запрошена операция над денормализованным числом;

ZE (2) — деление на 0 — запрошено деление на 0;

OE (3) — переполнение — результат слишком большой;

UE (4) — антипереполнение — результат слишком маленький;

PE (5) — неточный результат — результат не может быть представлен точно;

Биты 7..12 — маски исключений.

IM (7) — маска исключения IE;

DM (8) — маска исключения DE;

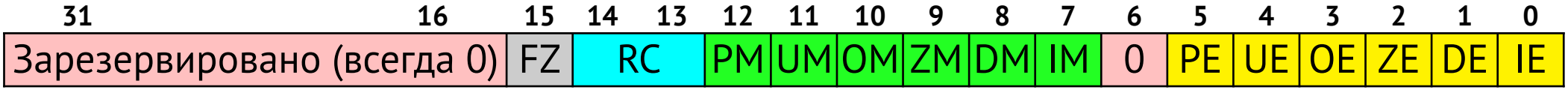
ZM (9) — маска исключения ZE;

OM (10) — маска исключения OE;

UM (11) — маска исключения UE;

PM (12) — маска исключения PE.

По умолчанию все маскирующие биты устанавливаются в 1, так что никакие исключения не обрабатываются.



RC (14..13) — управление округлением

0 (00) — к ближайшему целому (устанавливается по умолчанию);
1 (01) — к отрицательной бесконечности;
2 (10) — к положительной бесконечности;
3 (11) — к нулю;

FZ (15) — режим сброса в ноль (flush-to-zero)

По умолчанию выключен.

В этом режиме команды SSE не превращают слишком маленькое число с плавающей запятой в денормализованное, как этого требует IEEE 754, а возвращают 0.

Знак нуля соответствует знаку получившегося бы денормализованного числа, а также устанавливаются флаги PE и UE.

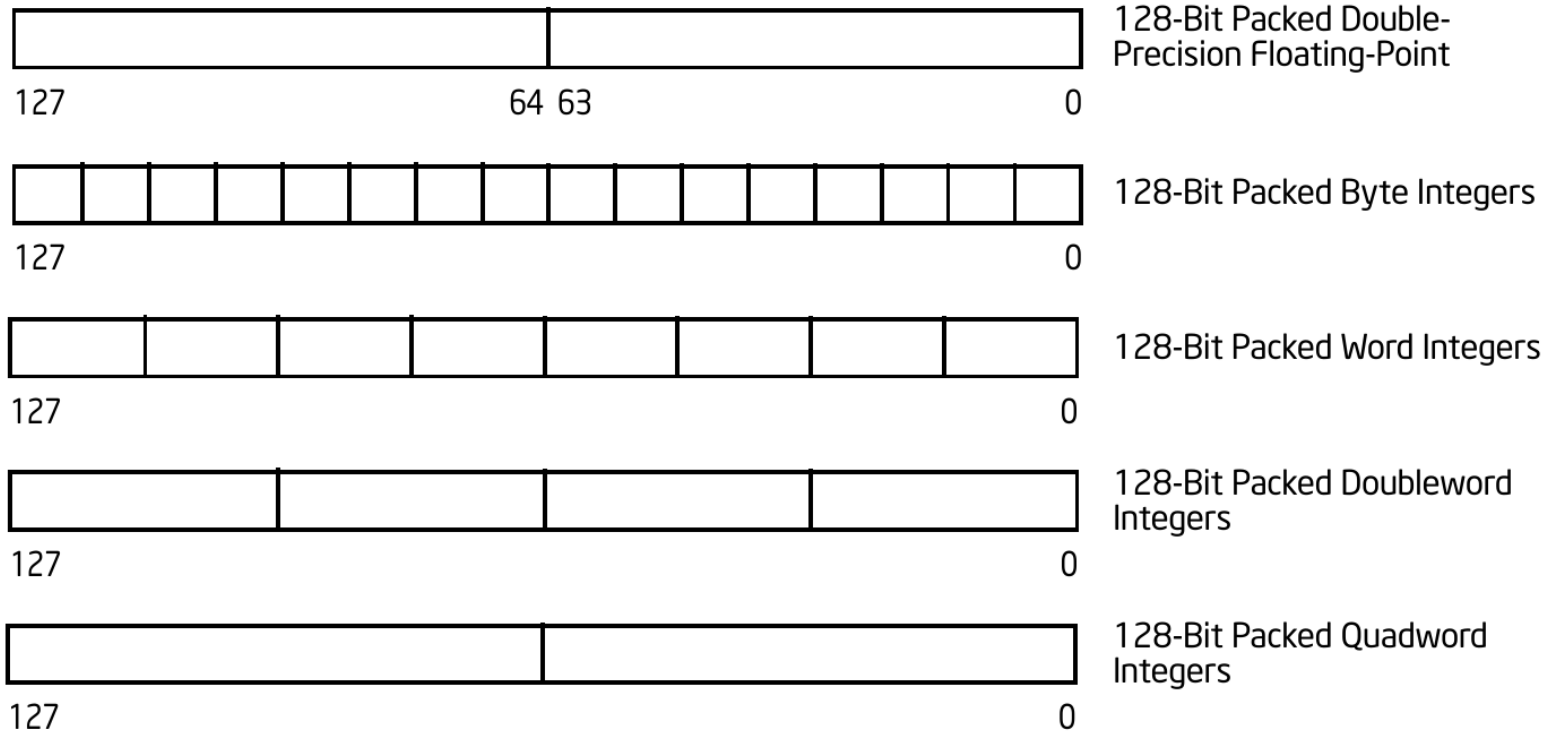
Типы данных SSE2

Основное преимущество SSE2 связано с применением 64-разрядных чисел с плавающей точкой двойной точности.

SSE2-команды при выполнении операций используют восемь 128-разрядных регистров (XMMO - XMM7) и могут работать в скалярном или параллельном режиме.

SSE2-команды оперируют с такими типами данных, как:

- упакованные и скалярные числа с плавающей точкой одинарной точности (SSE);
- упакованные и скалярные числа с плавающей точкой двойной точности (++);
- упакованные и скалярные **целые числа** размером 128 бит (++).



Команды 128-разрядной целочисленной арифметики используют тот же набор регистров (XMM0 - XMM7), что и команды, оперирующие с числами с плавающей точкой.

Инструкции SSE2 не требуют применения команды EMMS, поскольку выполняются вне зависимости от сопроцессора.

SSE2-команды позволяют:

- разрабатывать алгоритмы, в которых одновременно можно обрабатывать смешанные типы данных — упакованные числа с плавающей точкой в коротком формате и указанные с двойной точностью, а также целые 64- и 128-разрядные числа;

- работать с данными различной размерности: байтом, словом, двойным словом, учетверенным словом и двойным учетверенным словом.

С помощью параллельных команд можно одновременно обрабатывать все упакованные операнды, в то время как с помощью скалярных — только младший операнд.

Команды SSE2-расширения в большинстве случаев требуют выравнивания адресов операндов в памяти по 16-байтовой границе.

Есть исключения, например команда загрузки, или сохранения, операнда в невыровненной области памяти (**MOVUPD, MOVUPS**).

Есть скалярная команда, работающей с переменной размером в 8 байт и не требующей выравнивания.

Инструкции SSE2

Инструкции SSE2 разделены на четыре функциональные группы:

- упакованные и скалярные инструкции двойной точности с плавающей запятой
- 64-битные и 128-битные целочисленные инструкции SIMD
- 128-битные расширения целочисленных инструкций SIMD, представленные с технологией MMX и расширениями SSE.
- инструкции по управлению кэшированием и порядку исполнения инструкций.

Команды обработки 128-разрядных данных с плавающей точкой

В группу инструкций обработки 128-разрядных данных с плавающей точкой входят следующие инструкции:

- перемещения (пересылки, передачи) данных;
- арифметические (сложения, вычитания, умножения, деления, извлечения квадратного корня и поиска максимума/минимума);
- сравнения;
- логических операций;
- распаковки и распределения данных;
- преобразования форматов данных;
- управления состоянием вычислений;
- управления кэшированием данных.

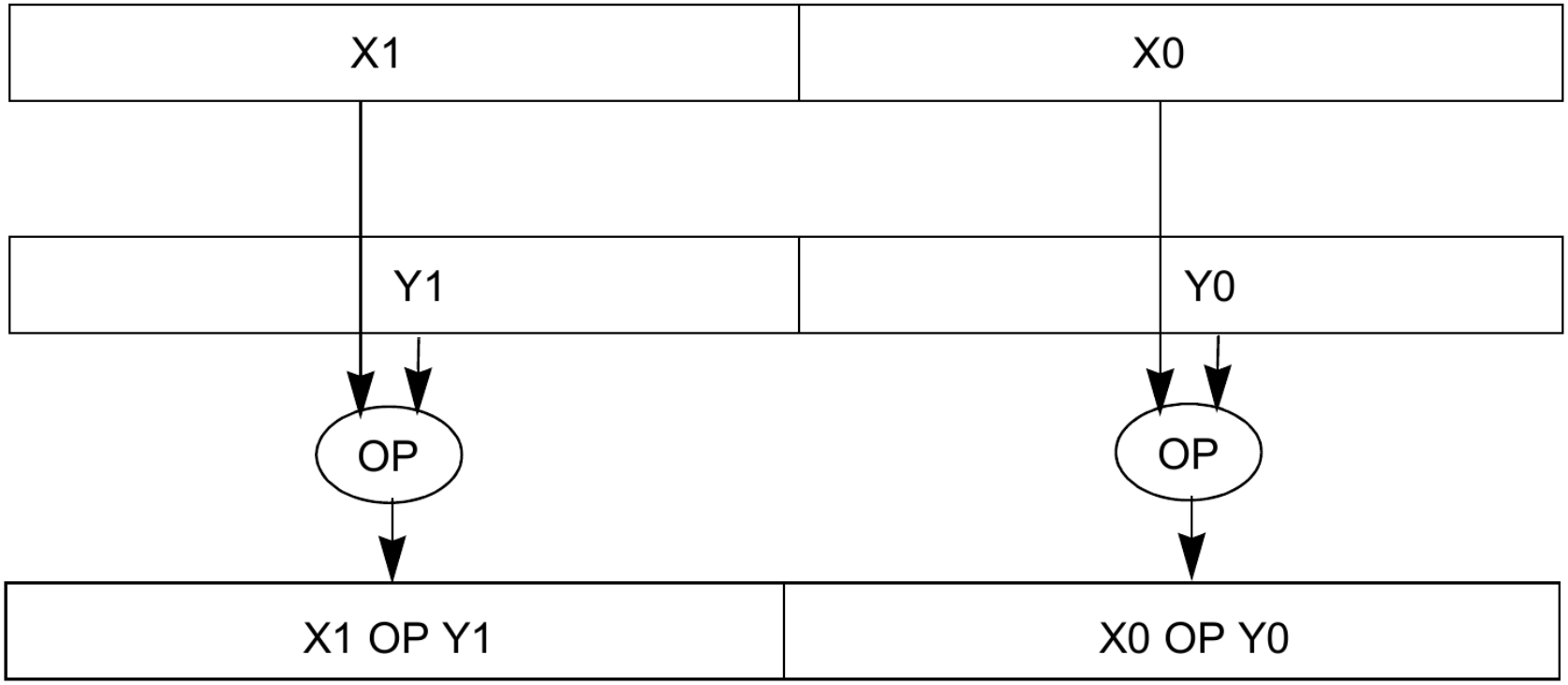
Инструкции подразделяются на параллельные и скалярные.

Мнемонические обозначения параллельных команд содержат суффикс **pd** (packed double), а скалярные — суффикс **sd** (scalar double), или **ps**, **ss** — packed/scalar single.

При разработке ассемблерных процедур с SSE2-командами в большинстве случаев требуется выравнивание адресов данных по 16-байтовой границе.

Параллельные инструкции с плавающей запятой двойной точности выполняют операции SIMD над упакованными операндами с плавающей запятой двойной точности.

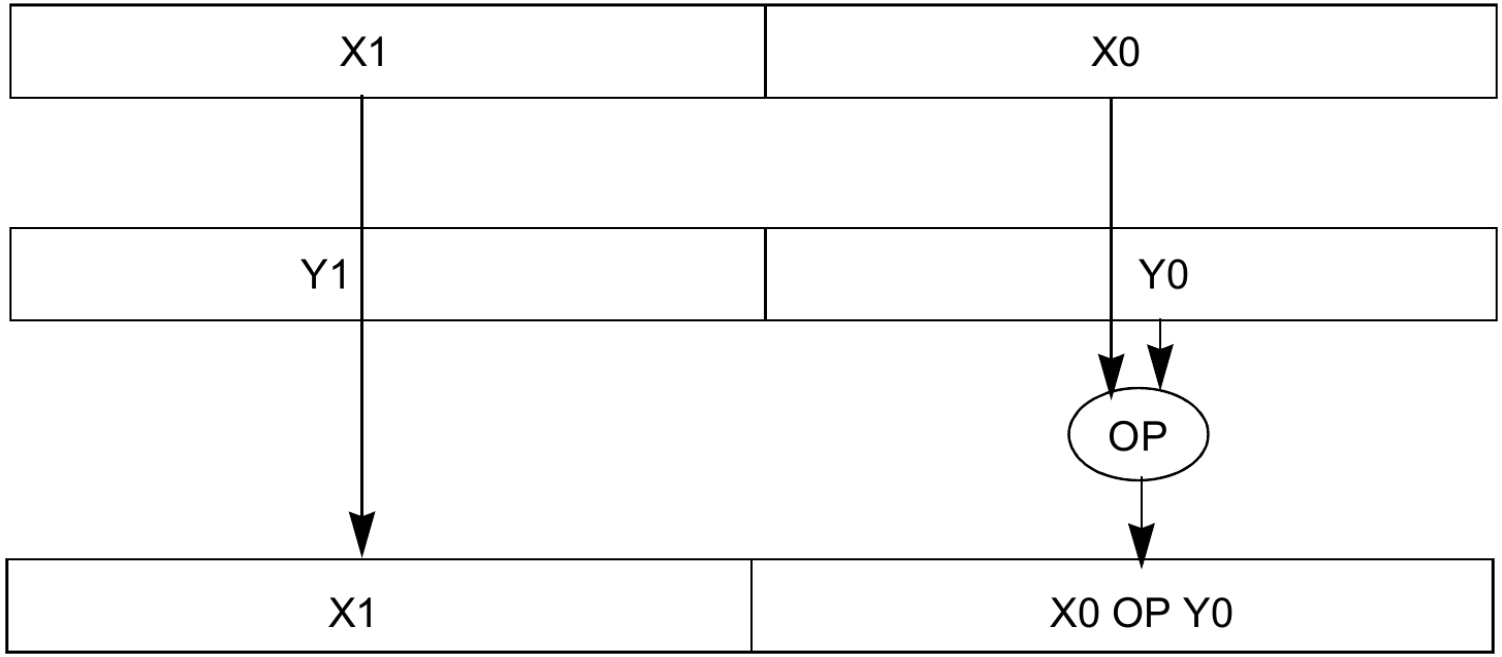
Каждый исходный операнд содержит два значения с плавающей запятой двойной точности, а целевой операнд содержит результаты операции (OP), выполненной параллельно с соответствующими значениями (X0 и Y0, X1 и Y1) в каждом операнде.



Скалярные инструкции с плавающей запятой двойной точности работают с младшими (наименее значимыми) квадрословами из двух исходных операндов (X0 и Y0). Наиболее значимое двойное слово (X1) первого исходного операнда просто передается в операнд назначения.

Скалярные операции аналогичны операциям с плавающей запятой, выполняемым в регистрах данных x87 FPU с установленным полем **PC** (управление точностью) в регистре управления x87 FPU в двойную точность (53-битовое значение).

Однако есть исключение — в операциях со стеком x87 для результата используется 15-битный диапазон экспонент, в то время как операции SSE используют 11-битный диапазон экспонент.



Команды перемещения данных

Команды перемещения данных перемещают данные с плавающей запятой двойной точности между регистрами XMM и между регистрами XMM и памятью.

MOVAPD — пересылка выравненных упакованных данных с плавающей запятой двойной точности — передает 128-битный упакованный операнд с плавающей запятой двойной точности из памяти в регистр XMM или наоборот, или между регистрами XMM. **Адрес памяти должен быть выровнен по 16-байтовой границе; если нет, генерируется исключение общей защиты (GP #).**

MOVUPD — пересылка невыровненных 128-разрядных упакованных данных с плавающей точкой двойной точности из входного операнда (источника) в выходной операнд (приемник).

В качестве источника и приемника могут выступать XMM-регистр или 128-разрядная ячейка памяти, при этом хотя бы один из операндов должен быть XMM-регистром. Выравнивание данных в памяти не требуется.

MOVSD — пересылка скалярных данных размером 64 бит из младшей части XMM-регистра в память и наоборот. Данные пересылаются из 64-разрядной ячейки памяти в младшую часть XMM-регистра. В качестве операнда-источника и операнда-приемника могут выступать XMM-регистр или ячейка памяти.

Если оба операнда являются XMM-регистрами, то пересылаются младшие части регистров.

Если выполняется пересылка данных из памяти в XMM-регистр, то старшие 64 бита регистра устанавливаются в 0;

MOVHPD — пересылка старших 64 битов XMM-регистра в память и наоборот.

Данные пересылаются из 64-разрядной ячейки памяти в старшую часть XMM-регистра.

В качестве операнда-источника и операнда-приемника могут выступать XMM-регистр или ячейка памяти.

Если выполняется пересылка данных в XMM-регистр, то младшая часть регистра не изменяется;

MOVLPD — пересылка младших 64 битов XMM-регистра в память и наоборот.

Данные пересылаются из 64-разрядной ячейки памяти в младшую часть XMM-регистра.

В качестве операнда-источника и операнда-приемника могут выступать XMM-регистр или ячейка памяти.

Если выполняется пересылка данных в XMM-регистр, то старшая часть регистра не изменяется;

MOVMSKPD — сохранение знаковых битов каждого 64-разрядного операнда с плавающей точкой двойной точности в младших битах 32-разрядного регистра общего назначения.

Это 2-разрядное значение может быть использовано для организации ветвлений в программе.

Арифметические инструкции SSE2

Арифметические инструкции SSE2 выполняют операции сложения, вычитания, умножения, деления, квадратного корня и получения максимального/минимального значений для упакованных и скалярных значений с плавающей запятой двойной точности.

ADDPD/SUBPD (сложение/вычитание упакованных значений с плавающей запятой двойной точности) складывают и вычитают, соответственно, два упакованных операнда с плавающей запятой двойной точности.

ADDSD/SUBSD (сложение/вычитание скалярных значений с плавающей запятой двойной точности) складывают и вычитают, соответственно, младшие компоненты двух операндов с плавающей запятой двойной точности и сохраняют результат в младшее квадрослово операнда назначения.

MULPD (умножение упакованных значений с плавающей запятой двойной точности) умножает два упакованных операнда с плавающей запятой двойной точности.

MULSD (умножение скалярных значений с плавающей запятой двойной точности) умножает два младших операнда с плавающей запятой двойной точности и сохраняет результат в младшем квадрослове операнда назначения.

DIVPD (деление упакованные значения с плавающей запятой двойной точности) делит два упакованных операнда с плавающей запятой двойной точности.

DIVSD (деление скалярных значений с плавающей запятой двойной точности) делит младшие значения с плавающей запятой двойной точности двух операндов и сохраняет результат в младшем квадрослове операнда назначения.

SQRTPD (вычислить квадратные корни упакованных значений с плавающей запятой двойной точности) вычисляет квадратные корни значений в упакованном операнде с плавающей запятой двойной точности.

SQRTSD (вычислить квадратный корень из скалярного значения с плавающей запятой двойной точности) вычисляет квадратный корень младшего значения с плавающей запятой двойной точности исходного операнда и сохраняет результат в нижнем квадрослове операнда назначения.

MAXPD/MINPD (вернуть максимальное/минимальное из упакованных значений с плавающей запятой двойной точности) сравнивает соответствующие значения в двух упакованных операндах с плавающей запятой двойной точности и возвращает численно большее/меньшее значение для каждой пары сравнения.

MAXSD/MINSD (вернуть максимальное/минимальное из скалярных значений с плавающей запятой двойной точности) сравнивает младшие компоненты с плавающей запятой двойной точности из двух упакованных операндов с плавающей запятой двойной точности и возвращает численно большее/меньшее значение в младшем квадрослове операнда назначения.

Логические инструкции SSE2

Логические инструкции SSE2 выполняют операции И, И НЕ, ИЛИ и исключающее ИЛИ для упакованных значений с плавающей запятой двойной точности.

ANDPD (побитовое логическое И упакованных значений с плавающей запятой двойной точности) возвращает логическое И для двух упакованных операндов с плавающей запятой двойной точности.

ANDNPD (побитовое логическое И НЕ упакованных значений с плавающей запятой двойной точности) возвращает логическое И НЕ двух упакованных операндов с плавающей запятой двойной точности.

ORPD (побитовое логическое ИЛИ упакованных значений с плавающей запятой двойной точности) возвращает логическое ИЛИ двух упакованных операндов с плавающей запятой двойной точности.

XORPD (побитовое логическое исключающее ИЛИ упакованных значений с плавающей запятой двойной точности) возвращает логическое исключающее ИЛИ двух упакованных операндов с плавающей запятой двойной точности.

Инструкции сравнения SSE2

Инструкции сравнения SSE2 сравнивают упакованные и скалярные значения с плавающей запятой двойной точности и возвращают результаты сравнения либо в целевом операнде, либо в регистр **EFLAGS**.

CMPPD (сравнить упакованные значения с плавающей запятой двойной точности) сравнивает соответствующие значения двух упакованных операндов с плавающей запятой двойной точности, используя третий непосредственный операнд в качестве предиката, и возвращает 64-битный результат со всеми устанавленными битами в 1 (сравнение верно) или в 0 (сравнение ложно) для каждого сравнения с операндом назначения.

```
CMPPS xmm1, xmm2/m128, imm8
CMPPD xmm1, xmm2/m128, imm8
```

Значение непосредственного операнда позволяет выбрать любое из восьми условий сравнения.

Предикат		Проверяемое утверждение	NAN
0	EQ	приемник равен источнику	0
1	LT	приемник строго меньше источника	0
2	LE	приемник меньше либо равен источнику	0
3	UNORD	приемник или источник являются NAN	1
4	NEQ	приемник неравен источнику	1
5	NLT	приемник больше либо равен источнику	1
6	NLE	приемник строго больше источника	1
7	ORD	ни приемник ни источник не являются NAN	0

Компиляторы и ассемблеры могут реализовывать псевдооперации с двумя операндами в дополнение к форме с тремя операндами.

Pseudo-Op	CMPPD Реализация
CMP**EQ**PD xmm1, xmm2	CMPPS xmm1, xmm2, 0
CMP**LT**PD xmm1, xmm2	CMPPS xmm1, xmm2, 1
CMP**LE**PD xmm1, xmm2	CMPPS xmm1, xmm2, 2
CMP**UNORD**PD xmm1, xmm2	CMPPS xmm1, xmm2, 3
CMP**NEQ**PD xmm1, xmm2	CMPPS xmm1, xmm2, 4
CMP**NLT**PD xmm1, xmm2	CMPPS xmm1, xmm2, 5
CMP**NLE**PD xmm1, xmm2	CMPPS xmm1, xmm2, 6
CMP**ORD**PD xmm1, xmm2	CMPPS xmm1, xmm2, 7

CMPSD (сравнить скалярные значения с плавающей запятой двойной точности) сравнивает младшие значения из двух упакованных операндов с плавающей запятой двойной точности, используя непосредственный операнд в качестве предиката, и возвращает 64-битный результат со всеми устанавленными битами в 1 (сравнение верно) или в 0 (сравнение ложно) для младшего сравнения с операндом назначения.

Непосредственный операнд устанавливает условие сравнения, как для **CMPPD**.

COMISD/UCOMISD – сравнение/неупорядоченное сравнение скалярных значений с плавающей запятой двойной точности с установкой EFLAGS.

Инструкции сравнивают младшие компоненты двух упакованных операндов с плавающей запятой двойной точности и устанавливают флаги ZF, PF и CF в регистре EFLAGS для отображения результата (больше, меньше, равно или неупорядочено).

Флаги **OF**, **SF**, **AF** обнуляются.

Если одно из сравниваемых чисел — не-число, все три флага (**ZF**, **PF**, **CF**) устанавливаются в 1.

Если сравниваемые числа равны, **ZF=1**, **PF=0**, **CF=0**.

Если **dst < src**, **ZF=0**, **PF=0**, **CF=1**.

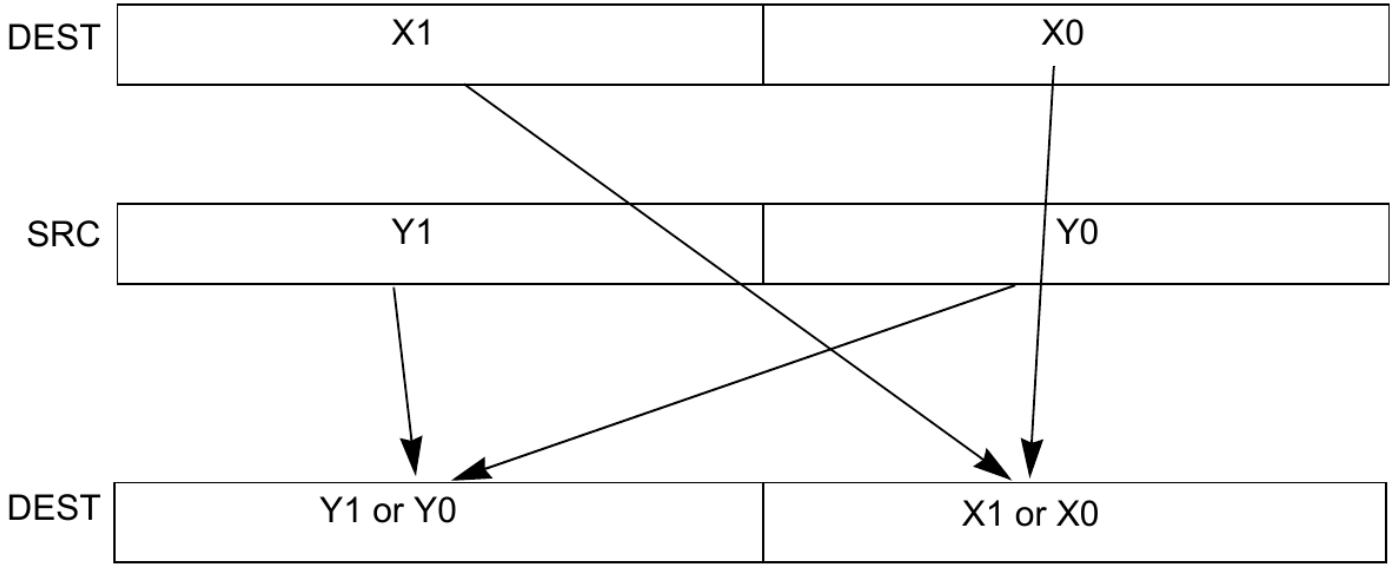
Если **dst > src**, **ZF=0**, **PF=0**, **CF=0**.

Эти две инструкции различаются следующим образом — инструкция **COMISD** генерирует исключение недопустимой операции (**#I**) с плавающей точкой, когда исходным операндом является либо **QNaN**, либо **SnaN**, а инструкция **UCOMISD** сигнализирует об исключении недопустимой операции, только если исходным операндом является **SnaN**.

Инструкции перемешивания и распаковки SSE2

Инструкции перемешивания SSE2 перемешивают содержимое двух упакованных значений с плавающей запятой двойной точности и сохраняют результаты в операнде-адресате.

SHUFPD (Packed Shuffle Operation — перемешать упакованные значения с плавающей запятой двойной точности) помещает одно из двух упакованных значений с плавающей запятой двойной точности из операнда назначения в младшее квадрослово операнда назначения, а одно из двух упакованных значений с плавающей запятой двойной точности из операнда-источника в старшее квадрослово операнда назначения.



Мнемоника

```
SHUFPD xmm1, xmm2/m128, imm8
```

Используя один и тот же регистр для операндов источника и назначения, **SHUFPD** может обменять два упакованных значения с плавающей запятой двойной точности.

Всего есть четыре варианта обмена компонент. Выбор варианта осуществляется двумя младшими битами третьего операнда **imm8**.

Бит 0 отвечает за содержимое **dst0**, бит 1 — за содержимое **dst1**.

В **dst1** помещается квадрослово из **src**, номер которого определяется старшим битом **imm**.

В **dst0** помещается квадрослово из **dst**, номер которого определяется младшим битом **imm**. , в частности, если **imm[0]=0** , то происходит горизонтальный перенос старшего квадрослова в младшее (**dst1→dst0**), в противном случае (**imm[0]=1**) содержимое младшего квадрослова остается без изменения.

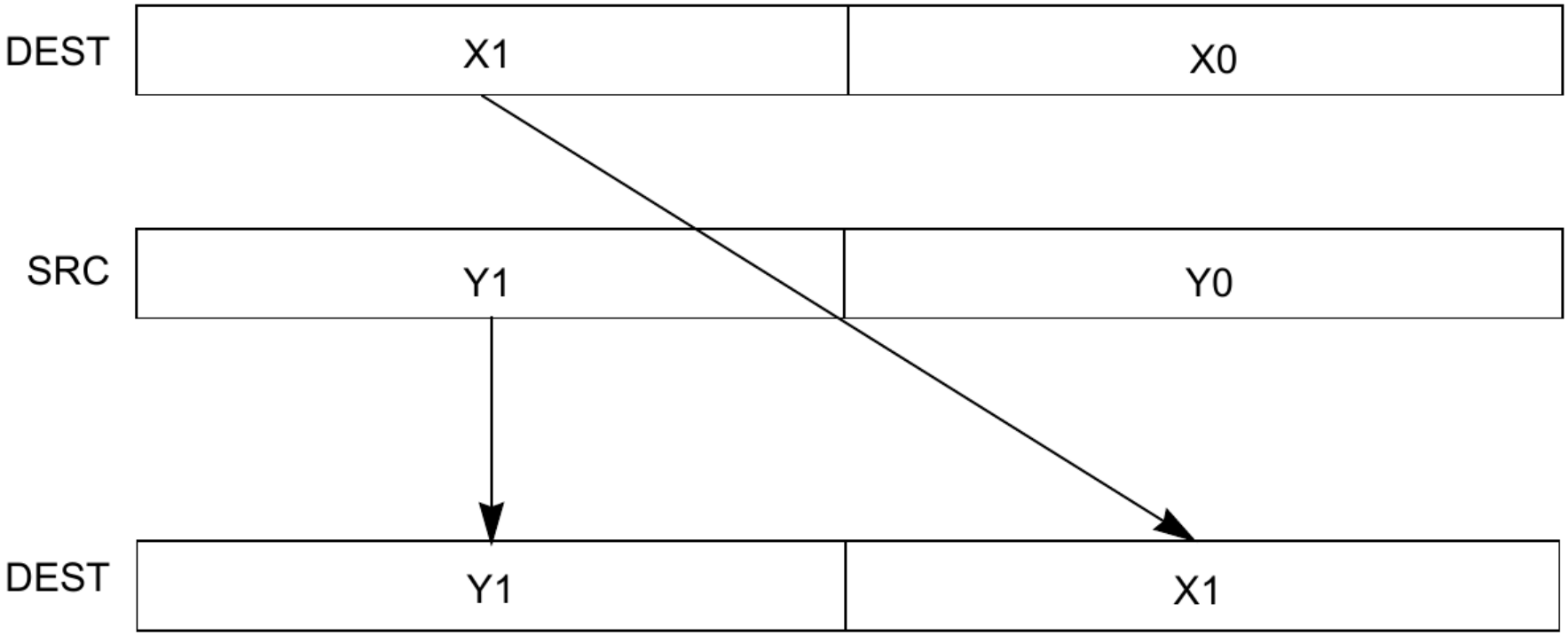
imm	dst1	dst0	Примечание	Алиас
00	src0	dst0	**dst0** остается на месте	**UNPCKLPD**
01	src0	dst1	**dst1 → dst0** (горизонтальное копирование)	
10	src1	dst0	**dst0** остается на месте	
11	src1	dst1	**dst1 → dst0** (горизонтальное копирование)	**UNPCKHPD**

```
;=================================
; sse.asm
;=================================
%include "syscalls_32.inc"
%include "syscalls.mac"

global  _start

;---------------------------------
section         .data
;---------------------------------
a               dq      1.0
b               dq      2.0
c               dq      3.0
d               dq      4.0
;---------------------------------
section         .bss
;---------------------------------
_xmm0           reso 2
_xmm1           reso 2
_xmm2           reso 2
_xmm3           reso 2
_xmm4           reso 2
_xmm5           reso 2
_xmm6           reso 2
_xmm7           reso 2
```

```
;---------------------------------
section .text
;---------------------------------
_start:
            movdqa  xmm1, [c]

            movdqa  xmm0, [a]
            shufpd  xmm0, xmm1,0x0
;---------------------------------
            movdqa  xmm0, [a]
            unpcklpd xmm0, xmm1
;---------------------------------
            movdqa  xmm0, [a]
            shufpd  xmm0, xmm1,0x1
;---------------------------------
            movdqa  xmm0, [a]
            shufpd  xmm0, xmm1,0x2
;---------------------------------
            movdqa  xmm0, [a]
            shufpd  xmm0, xmm1,0x3
;---------------------------------
            movdqa  xmm0, [a]
            unpckhpd xmm0, xmm1
;---------------------------------
            SYS_EXIT 0
;---------------------------------
```

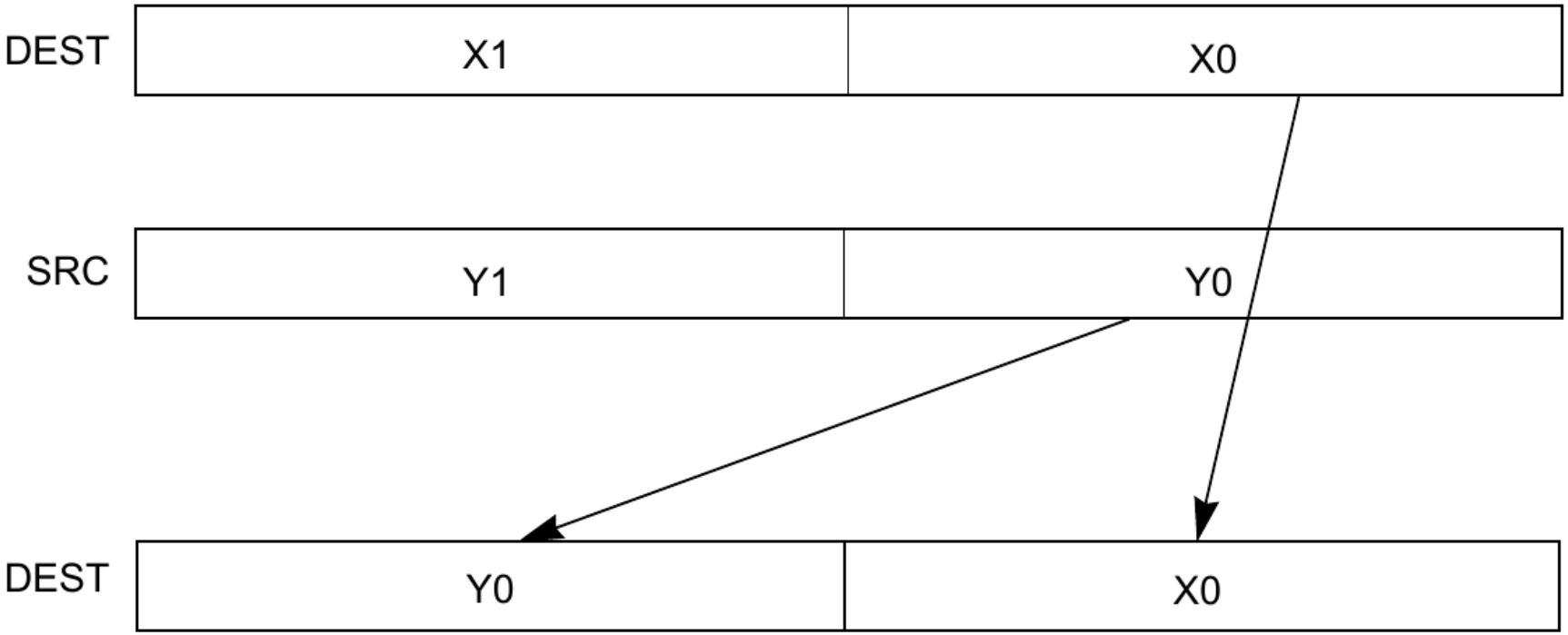
UNPCKHPD (распаковывать и чередовать старших упакованных значений с плавающей запятой двойной точности) выполняет чередующуюся распаковку старших значений из операндов источника и назначения и сохраняет результат в операнде назначения.

```
UNPCKHPD xmm1, xmm2/m128
```



UNPCKLPD (распаковывать и чередовать младшие упакованные значения с плавающей запятой двойной точности) выполняет чередующуюся распаковку младших значений из операндов источника и назначения и сохраняет результат в операнде назначения.

```
UNPCKLPD xmm1, xmm2/m128
```



Инструкции преобразования

Инструкции преобразования SSE2 поддерживают упакованные (**P**) и скалярные (**S**) преобразования между:

- форматами с плавающей запятой двойной (**D**) и одинарной (**S**) точности;
- целочисленным форматом размером в двойное слово и форматом с плавающей точкой двойной точности (**I D**);
- целочисленным форматом размером в двойное слово и форматом с плавающей точкой одинарной точности (**I S**).

Символьные компоненты в мнемониках инструкций имеют следующее значение:

CVT – преобразование в соответствии с режимом округления (если преобразование оказывается неточным, результат округляется в соответствии с режимом округления, выбранным в регистре MXCSR);

CVTT – преобразование с усечением (результат округляется в направлении нуля);

PD – упакованные с плавающей запятой двойной точности;

PS – упакованные с плавающей запятой одинарной точности;

SD – скалярные с плавающей запятой двойной точности;

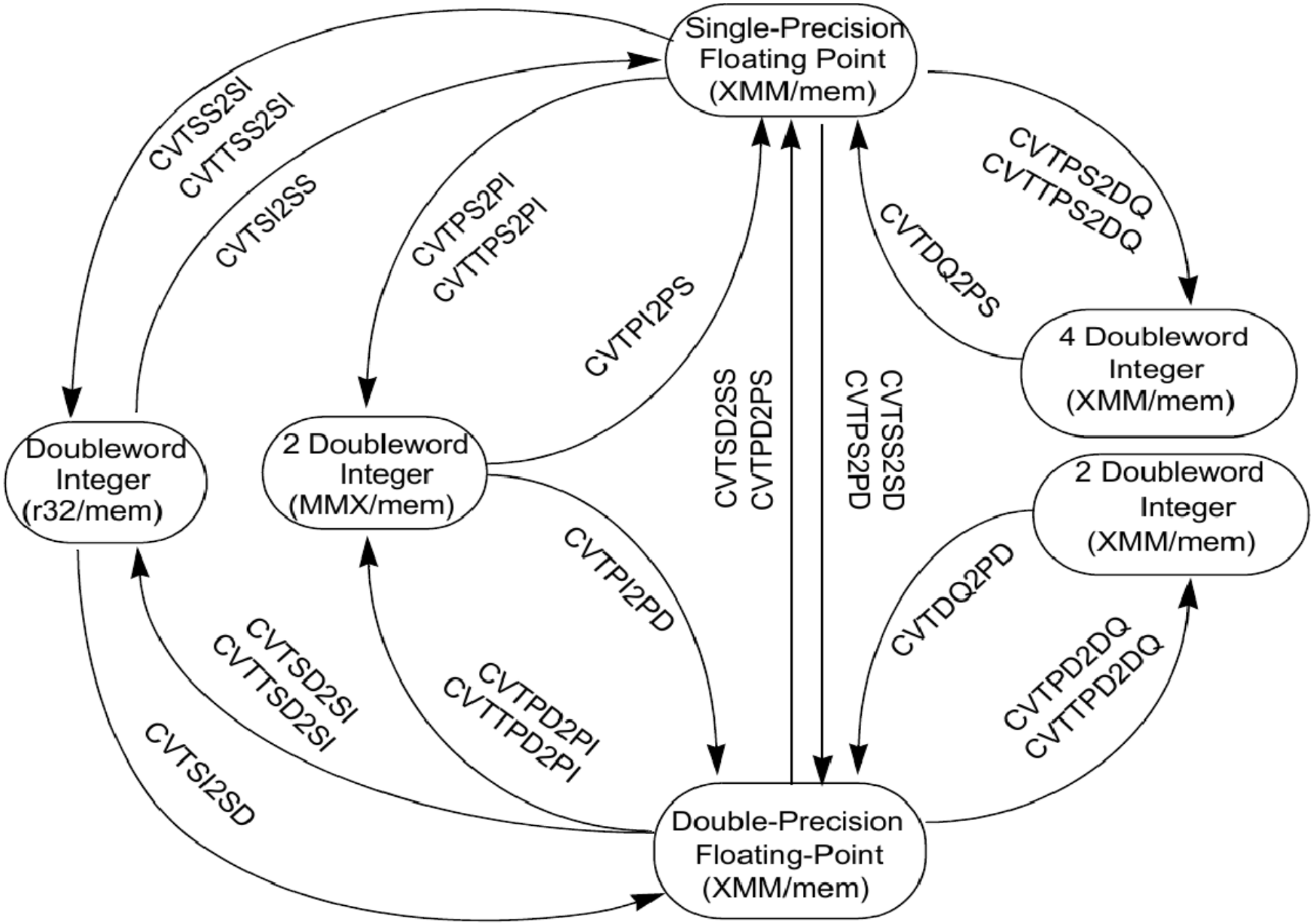
SS – скалярные с плавающей запятой одинарной точности;

DQ – упакованные целые числа в формате квадрослова слова;

PI – упакованные целые числа в формате двойного слова;

SI – скалярные целые числа в формате двойного слова.

Схема выполнения преобразований



Преобразования между значениями с плавающей запятой

CVTPD2PS — преобразование упакованных значений с плавающей запятой двойной точности в упакованные значения с плавающей запятой одинарной точности

```
CVTPD2PS xmm1, xmm2/m128 ; double --> float
```

Преобразует два (четыре или восемь) упакованных значений с плавающей запятой двойной точности в исходном операнде (второй операнд) в два (четыре или восемь) упакованных значений с плавающей запятой одинарной точности в операнде назначения (первый операнд).

Старшая часть операнда назначения очищается.

CVTPS2PD — преобразование упакованных значений с плавающей запятой одинарной точности в упакованные значения с плавающей запятой двойной точности

```
CVTPS2PD xmm1, xmm2/m64  ; float --> double
```

Преобразует два (четыре или восемь) упакованных значений с плавающей запятой одинарной точности в исходном операнде (второй операнд) в два (четыре или восемь) упакованных значений с плавающей запятой двойной точности в операнде назначения (первый операнд).

Преобразования между упакованными значениями с плавающей запятой и целыми числами

CVTPD2DQ — преобразование упакованных значений с плавающей запятой двойной точности в упакованные целые числа в формате двойного слова

```
CVTPD2DQ xmm1, xmm2/m128 ; double -> dword
```

Преобразует упакованные значения с плавающей запятой двойной точности из исходного операнда (второй операнд) в упакованные целые числа в формате двойного слова (32 бита) со знаком в операнд назначения (первый операнд).

Когда преобразование является неточным, возвращаемое значение округляется в соответствии с битами управления округлением в регистре MXCSR или встроенными битами управления округлением.

Если преобразованный результат не может быть представлен в целевом формате, возникает исключение недопустимой операции с плавающей запятой, а если это исключение маскируется, возвращается неопределенное целочисленное значение (2w -1, где w представляет количество битов в целевом формате).

CVTDQ2PD — преобразование упакованных целых чисел в формате двойного слова в упакованные значения с плавающей запятой двойной точности.

```
CVTDQ2PD xmm1, xmm2/m64 ; dword -> double
```

CVTPS2DQ — преобразование упакованных значений с плавающей запятой одинарной точности в упакованные целые значения в формате двойного слова со знаком

```
CVTPS2DQ xmm1, xmm2/m128
```

Преобразует четыре (восемь или шестнадцать) упакованных значений с плавающей запятой одинарной точности из исходного операнда в четыре (восемь или шестнадцать) целых чисел в формате двойного слова (32 бит) со знаком в операнде назначения.

Когда преобразование является неточным, возвращаемое значение округляется в соответствии с битами управления округлением в регистре MXCSR или встроенными битами управления округлением.

Если преобразованный результат не может быть представлен в целевом формате, возникает исключение недопустимой операции с плавающей запятой, а если это исключение маскируется, возвращается неопределенное целочисленное значение (2w -1, где w представляет количество битов в целевом формате).

CVTDQ2PS — преобразование упакованных целых чисел в формате двойного слова в упакованные значения с плавающей запятой одинарной точности.

```
CVTDQ2PS xmm1, xmm2/m128
```

Преобразует четыре (восемь или шестнадцать) целых чисел в формате двойного слова (32 бит) со знаком из исходного операнда в четыре (восемь или шестнадцать) упакованных значения с плавающей запятой одинарной точности в операнде назначения.

CVTPD2PI — преобразование упакованных значений с плавающей запятой двойной точности в упакованные двойные целые числа

```
CVTPD2PI mm, xmm/m128
```

Преобразует упакованные значения с плавающей запятой двойной точности из исходного операнда (второй операнд) в упакованные целые числа в формате двойного слова (32 бита) со знаком в операнд назначения (первый операнд), которым является регистр MMX.

Если преобразование является неточным, возвращаемое значение округляется в соответствии с битами управления округлением в регистре MXCSR. Если преобразованный результат больше, чем максимальное целое двойное слово со знаком, возникает исключение недопустимой операции с плавающей запятой, и если это исключение маскируется, возвращается неопределенное целое значение (80000000H).

Эта инструкция вызывает переход от операции x87 FPU к операции технологии MMX (то есть указатель вершины стека x87 FPU устанавливается в 0, а теговое слово x87 FPU устанавливается на все 0 [действительно]). Если эта инструкция выполняется во время ожидания исключения x87 FPU с плавающей запятой, исключение обрабатывается до выполнения инструкции CVTPD2PI.

CVTPI2PD — преобразование упакованных целых чисел двойного слова в упакованные значения с плавающей запятой двойной точности

```
CVTPI2PD xmm, mm/m64*
```

Преобразует два упакованных целых числа в формате двойного слова со знаком из исходного операнда (второй операнд) в два упакованных значения двойной точности с плавающей запятой в операнде назначения (первый операнд).

Исходным операндом может быть технологический регистр MMX или 64-битная ячейка памяти. Операнд назначения — это регистр XMM. Кроме того, в зависимости от конфигурации операнда:

- для операндов (xmm, mm) инструкция вызывает переход от x87 FPU к операции технологии MMX (то есть указатель вершины стека x87 FPU устанавливается на 0, а теговое слово x87 FPU устанавливается на все 0 [действительно]). Если эта инструкция выполняется во время ожидания исключения x87 FPU с плавающей запятой, исключение обрабатывается до выполнения инструкции CVTPI2PD.

- для операндов (xmm, m64) инструкция не вызывает перехода на технологию MMX и не принимает исключений x87 FPU.

CVTPS2PI – преобразование упакованных значений с плавающей запятой одинарной точности в упакованные двойные целые числа

```
CVTPS2PI mm, xmm/m64 ; два float --> два dword
```

CVTPI2PS – преобразование упакованных целых чисел двойного слова в упакованные значения с плавающей запятой одинарной точности

```
CVTPI2PS xmm, mm/m64 ; два dword --> два float
```

Преобразования между скалярными значениями с плавающей запятой и целыми числами

CVTSD2SS – преобразование скалярного значения с плавающей запятой двойной точности в скалярное значение с плавающей запятой одинарной точности

```
CVTSD2SS xmm1, xmm2/m64
```

CVTSS2SD – преобразование скалярного значения с плавающей запятой одинарной точности в скалярное значение с плавающей запятой двойной точности

```
CVTSS2SD xmm1, xmm2/m32
```

CVTSD2SI – преобразование скалярного числа с плавающей запятой двойной точности в целое число в формате двойного слова

```
CVTSD2SI r32, xmm1/m64
```

CVTSI2SD – преобразование целого числа в формате двойного слова в скалярное значение с плавающей запятой двойной точности

```
CVTSI2SD xmm1, r32/m32
```

CVTSI2SS – преобразование целого числа в формате двойного слова или квадрослова в скалярное значение с плавающей запятой одинарной точности

```
CVTSI2SS xmm1, r/m32 ; dword
CVTSI2SS xmm1, r/m64 ; qword
```

CVTSS2SI – преобразование скалярного значения с плавающей запятой одинарной точности в целое число в формате двойного слова или квадрослова

```
CVTSS2SI r32, xmm1/m32 ; dword
CVTSS2SI r64, xmm1/m32 ; qword
```

Преобразования между значениями с плавающей запятой и целыми числами с усечением

CVTTPD2DQ — преобразование с усечением упакованных значений с плавающей запятой двойной точности в упакованные целые числа в формате двойного слова

```
CVTTPD2DQ xmm1, xmm2/m128
```

Преобразует два (четыре или восемь) упакованных значений двойной точности с плавающей запятой из исходного операнда (второй операнд) в два (четыре или восемь) упакованных целых чисел двойного слова со знаком в операнде назначения (первый операнд).

Когда преобразование является неточным, возвращается усеченное (округленное до нуля) значение. Если преобразованный результат больше, чем максимальное целое двойное слово со знаком, возникает исключение недопустимости с плавающей запятой, и если это исключение маскируется, возвращается неопределенное целое значение (80000000H).

CVTTPD2PI — преобразование с усечением упакованных значений с плавающей запятой двойной точности в упакованные целые числа в формате двойного слова

```
CVTTPD2PI mm, xmm/m128    ; два double --> два dword
```

CVTTPS2DQ — преобразование с усечением упакованных значений с плавающей запятой одинарной точности в упакованные целые в формате двойного слова со знаком

```
CVTTPS2DQ xmm1, xmm2/m128 ; четыре float --> четыре dword
```

CVTTPS2PI — преобразование с усечением упакованных значений с плавающей запятой одинарной точности в упакованные целые числа в формате двойного слова

```
CVTTPS2PI mm, xmm/m64
```

CVTTSD2SI — преобразование с усечением скалярного значения с плавающей запятой двойной точности в целое число со знаком

```
CVTTSD2SI r32, xmm1/m64 ; dword
CVTTSD2SI r64, xmm1/m64 ; qword
```

CVTTSS2SI — преобразование с усечением скалярного значения с плавающей запятой одинарной точности в целое число

```
CVTTSS2SI r32, xmm1/m32
CVTTSS2SI r64, xmm1/m32
```

Расширение SSE3

Расширение SSE3 предлагает 13 инструкций, которые повышают производительность технологий Streaming SIMD Extensions, Streaming SIMD Extensions 2 и математических возможностей x87-FP. Эти инструкции могут быть сгруппированы в следующие категории:

- одна инструкция x87FPU, используемая в целочисленном преобразовании;
- одна целочисленная инструкция SIMD, которая обращается к не выровненным загрузкам данных;
- две SIMD упакованные инструкции ADD/SUB с плавающей точкой;
- четыре SIMD горизонтальные инструкции ADD/SUB с плавающей точкой;
- три SIMD инструкции с плавающей точкой LOAD/MOVE/DUPLICATE;
- инструкция по синхронизации двух потоков.

Инструкции SSE3 могут выполняться только на процессорах Intel 64 и IA-32, которые поддерживают расширения SSE3. Поддержка этих инструкций может быть обнаружена с помощью инструкции CPUID.

FISTTP — инструкция целочисленного преобразования x87-FP. Ведет себя как инструкция **FISTP**, но использует усечение, независимо от режима округления, указанного в управляющем слове с плавающей запятой (FCW).

```
FISTTP m16int ; ST(0) → m16int с обрезанием
FISTTP m32int ; ST(0) → m32int с обрезанием
FISTTP m64int ; ST(0) → m64int с обрезанием
```

LDDQU — специальная 128-битная инструкция загрузки данных без выравнивания, предназначенная для предотвращения расщепления строк кэша.

ADDSUBPS — выполняет сложение с одинарной точностью для второй и четвертой пары 32-битных элементов данных в операндах и вычитание с одинарной точностью первой и третьей пар.

ADDSUBPD — выполняет сложение с двойной точностью второй пары и вычитание с двойной точностью первой пары квадрослов.

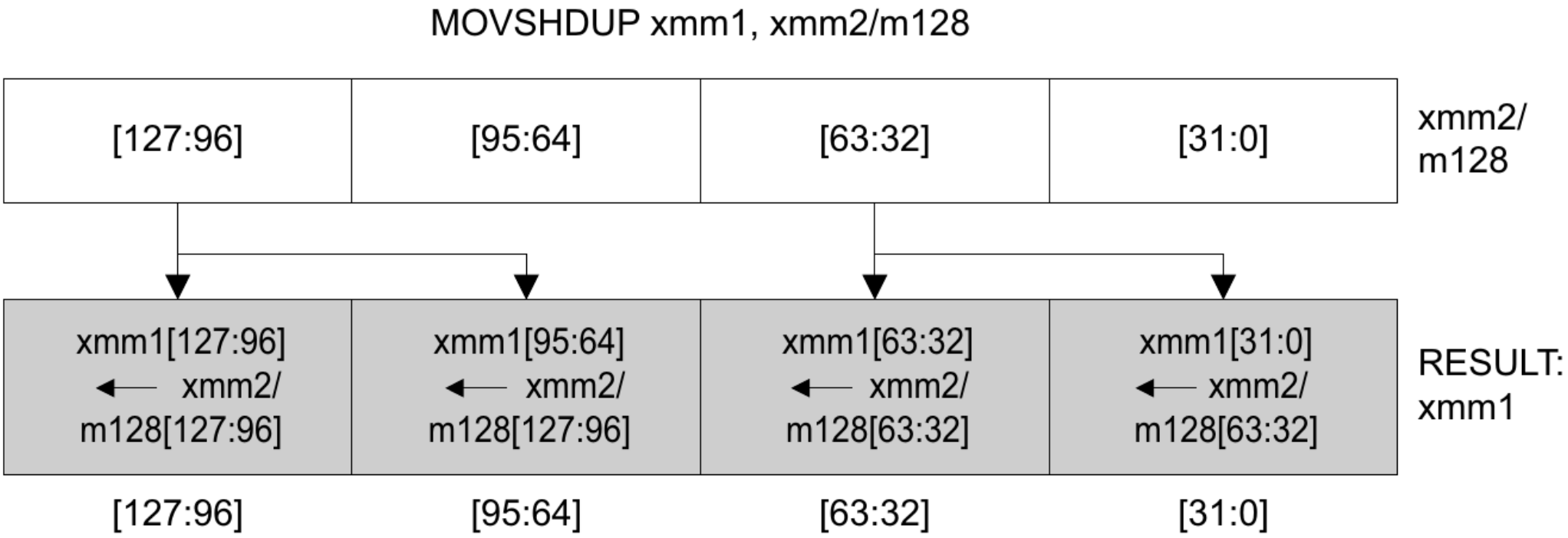
HADDPS — выполняет сложение с одинарной точностью для смежных элементов данных. Первый элемент данных результата получается путем сложения первого и второго элементов первого операнда, второй элемент — путем сложения третьего и четвертого элементов первого операнда, третий — путем сложения первого и второго элементов второго операнда и четвертый — путем сложения третьего и четвертого элементов второго операнда.

HSUBPS — выполняет вычитание с одинарной точностью для смежных элементов данных. Первый элемент данных результата получается путем вычитания второго элемента первого операнда из первого элемента первого операнда, второй элемент — путем вычитания четвертого элемента первого операнда из третьего элемента первого операнда, третий — путем вычитания второго элемента второго операнда из первого элемента второго операнда и четвертый — путем вычитания четвертого элемента второго операнда из третьего элемента второго операнда.

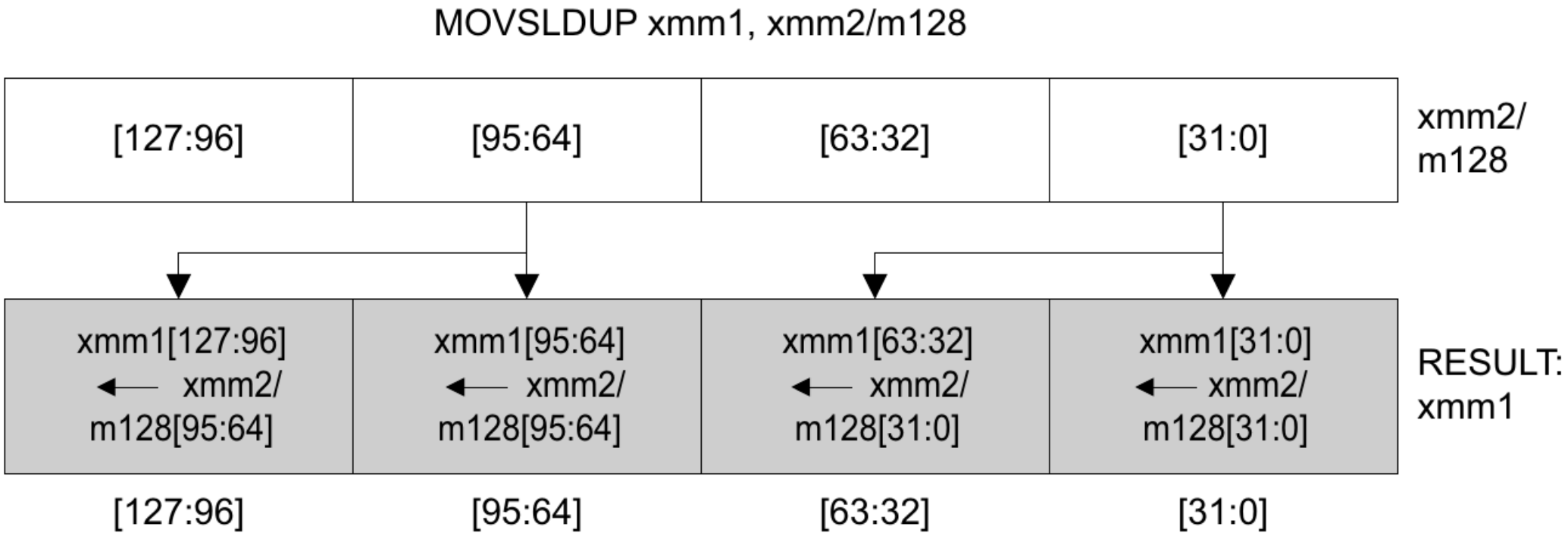
HADDPD — выполняет сложение с двойной точностью для смежных элементов данных. Первый элемент данных результата получается путем сложения первого и второго элементов первого операнда, второй элемент — путем сложения первого и второго элементов второго операнда.

HSUBPD — выполняет вычитание двойной точности для смежных элементов данных. Первый элемент данных результата получается путем вычитания второго элемента первого операнда из первого элемента первого операнда, второй элемент — путем вычитания второго элемента второго операнда из первого элемента второго операнда.

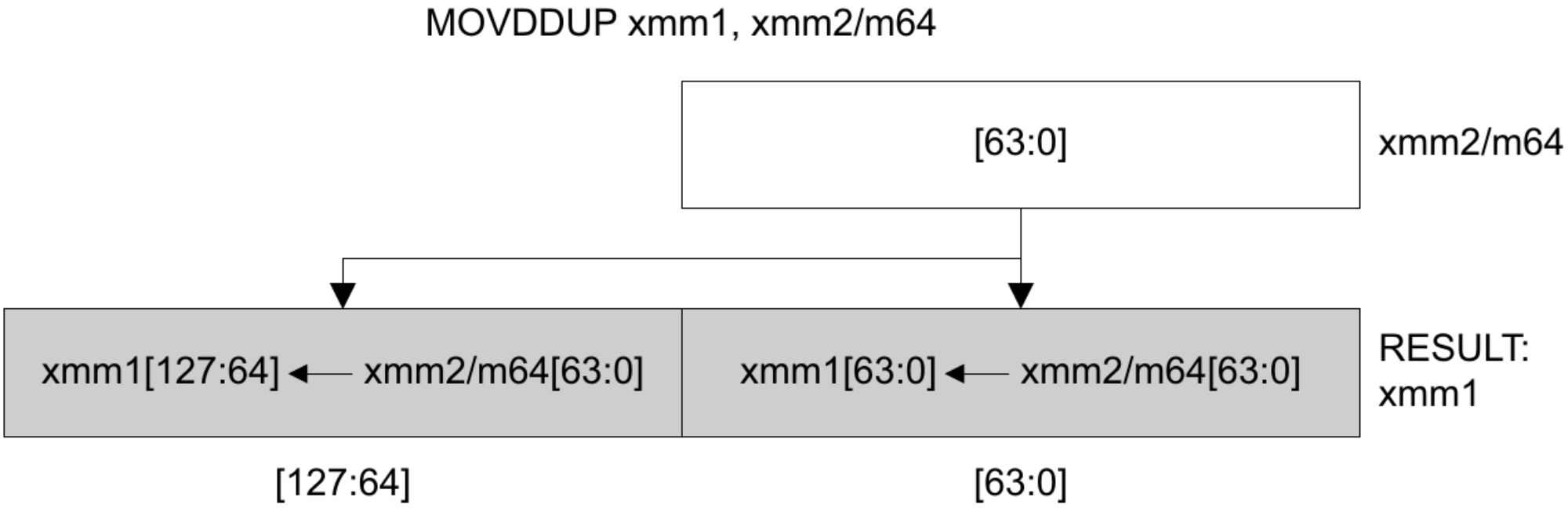
MOVSHDUP — загружает/перемещает 128 бит, дублирует второй и четвертый 32-битные элементы данных.



MOVSLDUP — загружает/перемещает 128 бит, дублирует первый и третий 32-битные элементы данных.



MOVDDUP — загружает/перемещает 64 бита (биты [63: 0], если источником является регистр) и возвращает одинаковые 64 бита как в нижней, так и в верхней половине 128-битного регистра результата. Дублирует 64 бита из источника



Инструкции дополнительного потокового SIMD-расширения 3 (SSSE3)

SSSE3 предоставляет 32 инструкции (представленные 14 мнемониками) для ускорения вычислений на упакованных целых числах.

Они включают:

12 инструкций, которые выполняют горизонтальные операции сложения или вычитания;

6 инструкций, которые оценивают абсолютные значения;

2 инструкции, которые выполняют операции умножения и сложения и ускоряют оценку точечных произведений;

2 инструкции, которые ускоряют операции умножения упакованных целых и производят целочисленные значения с масштабированием;

2 инструкции, которые выполняют побайтовую, тасовку на месте в соответствии со вторым операндом управления перемешиванием;

6 инструкций, которые отменяют упакованные целые числа в операнде-адресате, если знаки соответствующего элемента в операнде-источнике меньше нуля;

2 инструкции, которые выравнивают данные из комбинации двух операндов.

Инструкции SSSE3 могут выполняться только на процессорах Intel 64 и IA-32, которые поддерживают расширения SSSE3. Поддержка этих инструкций может быть обнаружена с помощью инструкции CPUID.

Горизонтальная и асимметричная обработка

Многие инструкции SSE/SSE2/SSE3/SSSE3 ускоряют обработку данных SIMD с помощью модели, называемой вертикальным вычислением. Используя эту модель, поток данных является вертикальным между элементами входных и выходных данных.

На рисунке ниже показаны асимметричная и горизонтальная обработка данных SSE3.

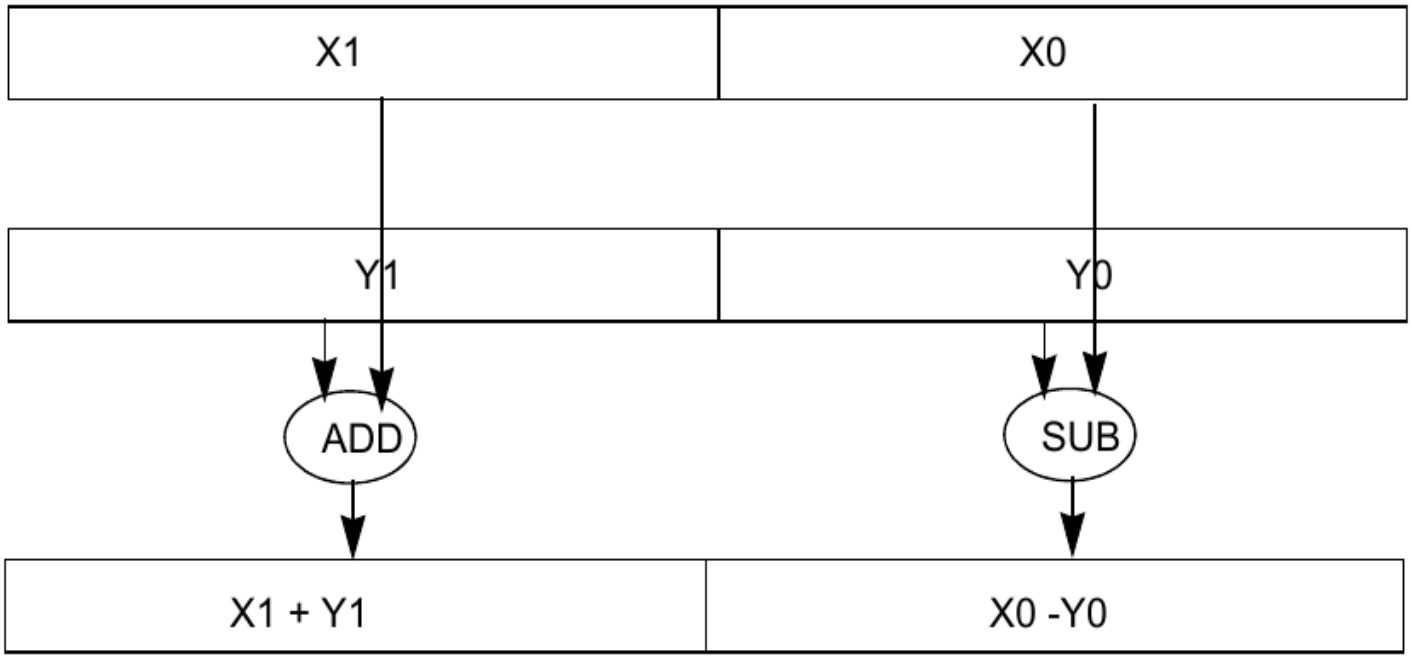


Рисунок — Асимметричная и вертикальная обработка данных инструкцией ADDSUBPD.

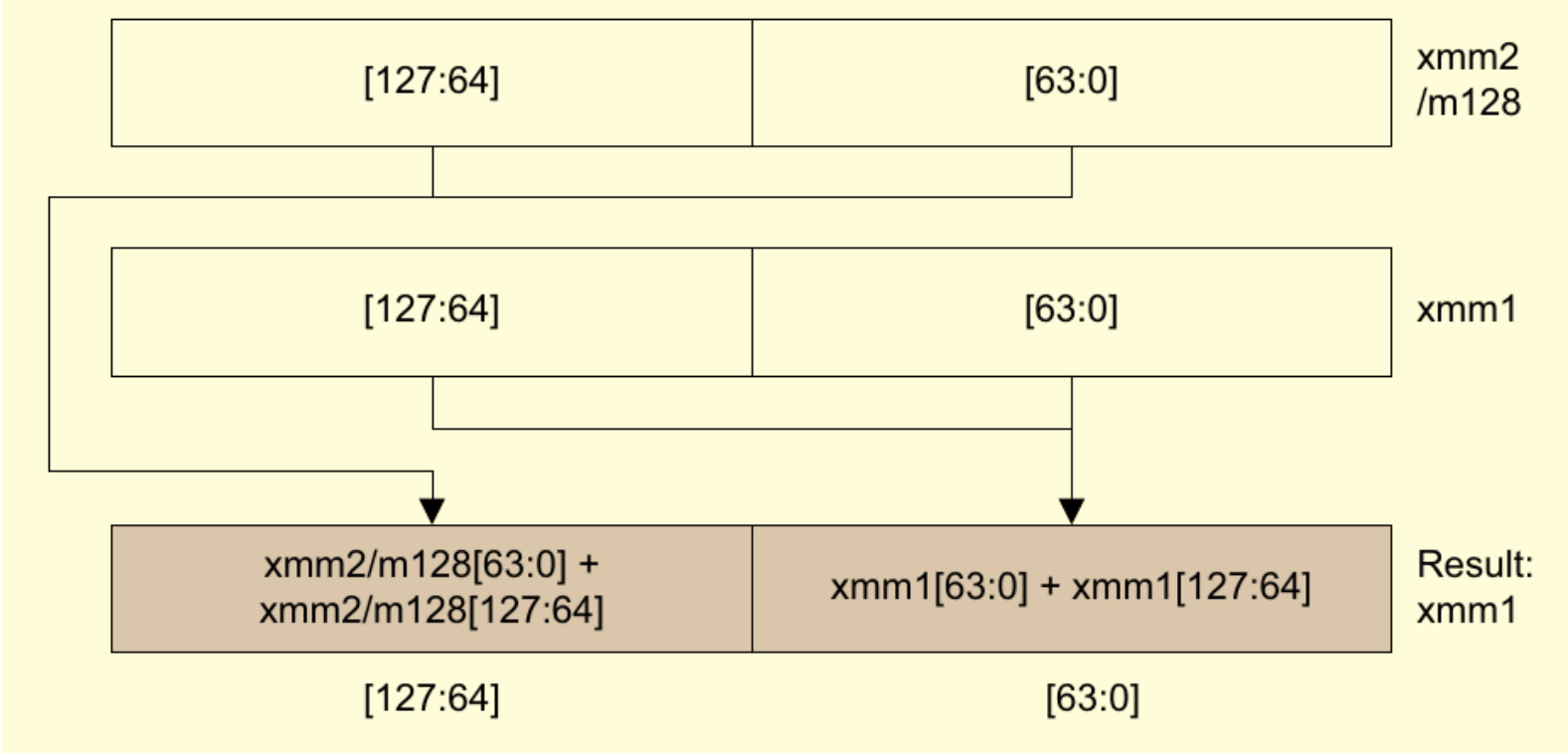
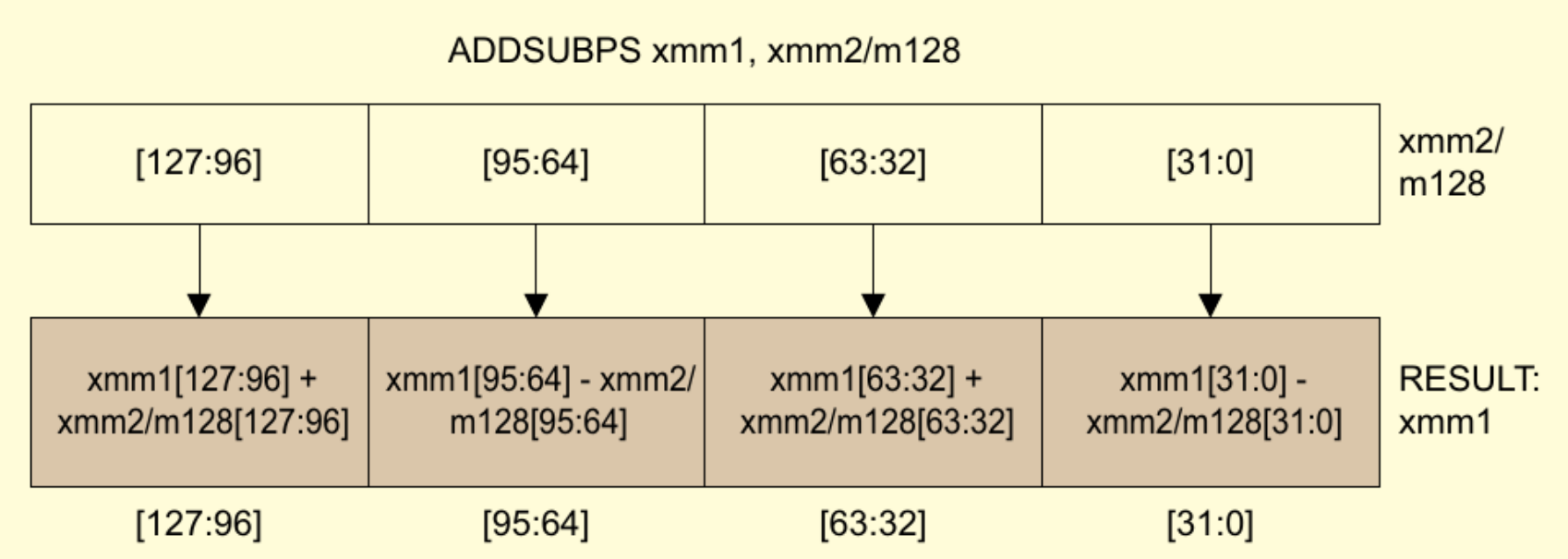


Рисунок – Горизонтальное перемещение данных данных инструкцией **HADDPD**.

Команды SIMD с плавающей запятой обеспечивают упакованное сложение/вычитание

ADDSUBPS — инструкция имеет два 128-битных операнда. Команда выполняет сложение с одинарной точностью для второй и четвертой пар 32-битных элементов данных в операндах, а также вычитание одинарной точности по первой и третьей парам.



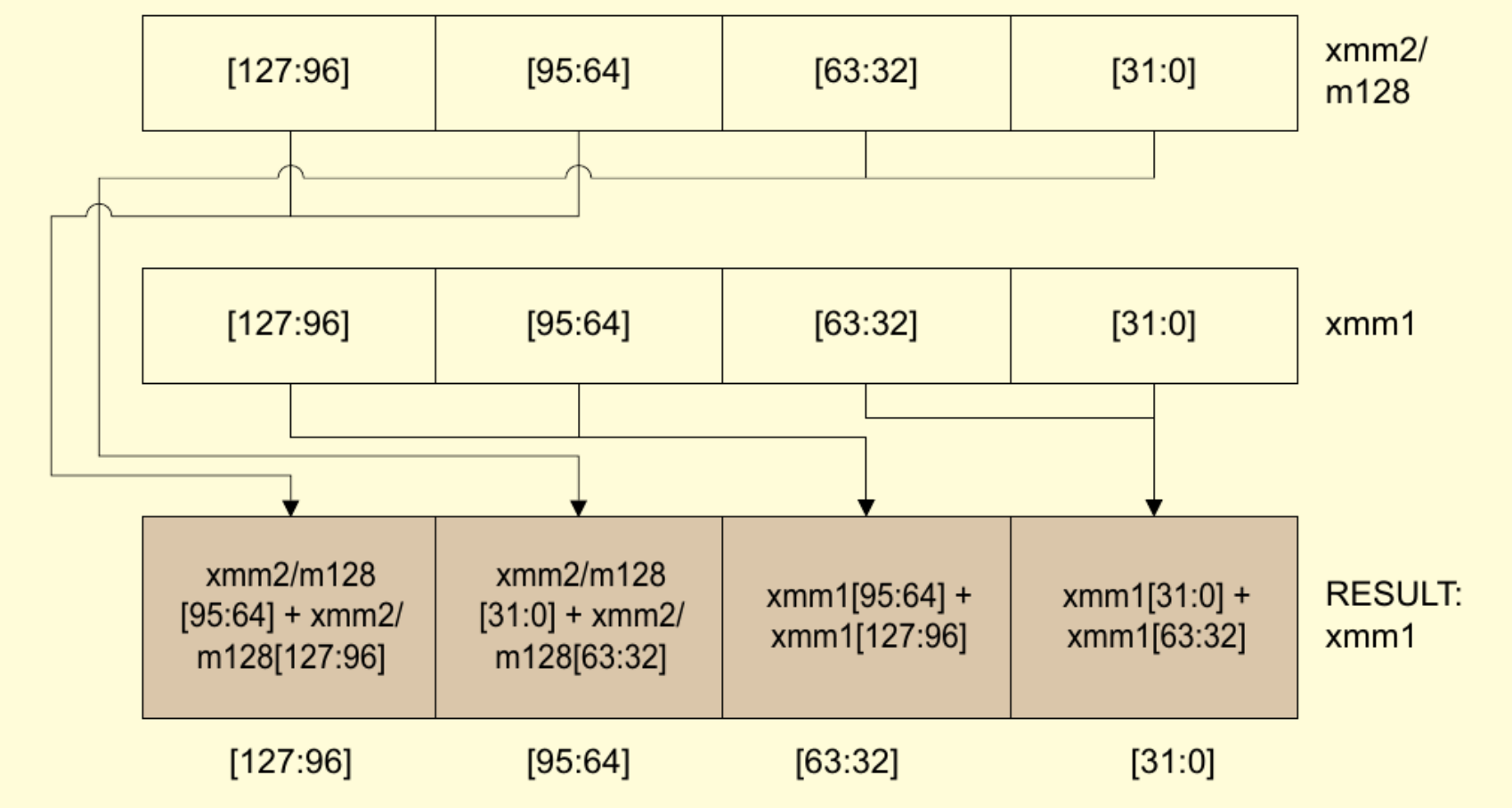
ADDSUBPD — инструкция имеет два 128-битных операнда. Инструкция выполняет сложение с двойной точностью для второй пары четверных слов и вычитание с двойной точностью для первой пары (см.выше).

Команды с плавающей запятой обеспечивают горизонтальное сложение/вычитание

Большинство инструкций SIMD работают вертикально. Это означает, что результат в позиции k является функцией элементов в позиции k обоих операндов.

Горизонтальное сложение/вычитание работает горизонтально. Это означает, что для получения результата используются смежные элементы данных в одном исходном операнде.

HADDPS — инструкция выполняет сложение с одинарной точностью для смежных элементов данных.



Первый элемент данных результата получается путем сложения первого и второго элементов первого операнда.

Второй элемент путем сложения третьего и четвертого элементов первого операнда.

Третий — сложением первого и второго элементов второго операнда.

Четвертый — сложением третьего и четвертого элементов второго операнда.

HSUBPS — инструкция выполняет вычитание с одинарной точностью для смежных элементов данных.

Первый элемент данных результата получается вычитанием второго элемента первого операнда из первого элемента первого операнда. Второй элемент — путем вычитания четвертого элемента первого операнда из третьего элемента первого операнда. Третий — вычитанием второго элемента второго операнда из первого элемента второго операнда. Четвертый — путем вычитания четвертого элемента второго операнда из третьего элемента второго операнда.

HADDPD — инструкция выполняет сложение с двойной точностью для смежных элементов данных.

Первый элемент данных результата получается путем сложения первого и второго элементов первого операнда. Второй элемент — сложением первого и второго элементов второго операнда.

HSUBPD — инструкция выполняет вычитание с двойной точностью для смежных элементов данных.

Первый элемент данных результата получается вычитанием второго элемента первого операнда из первого элемента первого операнда. Второй элемент — путем вычитания второго элемента второго операнда из первого элемента второго операнда.

Обзор инструкций SSSE3

SSSE3 предоставляет 32 инструкции для ускорения различных мультимедийных приложений и приложений обработки сигналов, использующих целочисленные данные SIMD. Они включают:

- Двенадцать инструкций, выполняющих операции горизонтального сложения или вычитания;
- Шесть инструкций, вычисляющих абсолютные значения.
- Две инструкции, которые выполняют операции умножения и сложения и ускоряют вычисление скалярных произведений;
- Две инструкции, которые ускоряют операции упакованного целочисленного умножения и производят целочисленные значения с масштабированием.
- Две инструкции, которые выполняют побайтовое перемешивание на месте в соответствии со вторым операндом управления перемешиванием.
- Шесть инструкций, которые инвертируют упакованные целые числа в операнде-адресате, если знаки соответствующего элемента в исходном операнде меньше нуля.
- Две инструкции, которые выравнивают данные из двух операндов. Операнды этих инструкций представляют собой упакованные целые числа размером в байты, слова или двойные слова. Операнды хранятся в виде 64- или 128-битных данных в регистрах MMX, регистрах XMM или в памяти.

Горизонтальное сложение/вычитание

По аналогии с упакованными командами горизонтального сложения и вычитания с плавающей запятой в SSE3, SSSE3 предлагает аналогичные возможности для упакованных целочисленных данных.

Поддерживаются элементы данных слов и двойных слов со знаком.

Также поддерживается насыщенная версия для горизонтального сложения и вычитания слов со знаком.

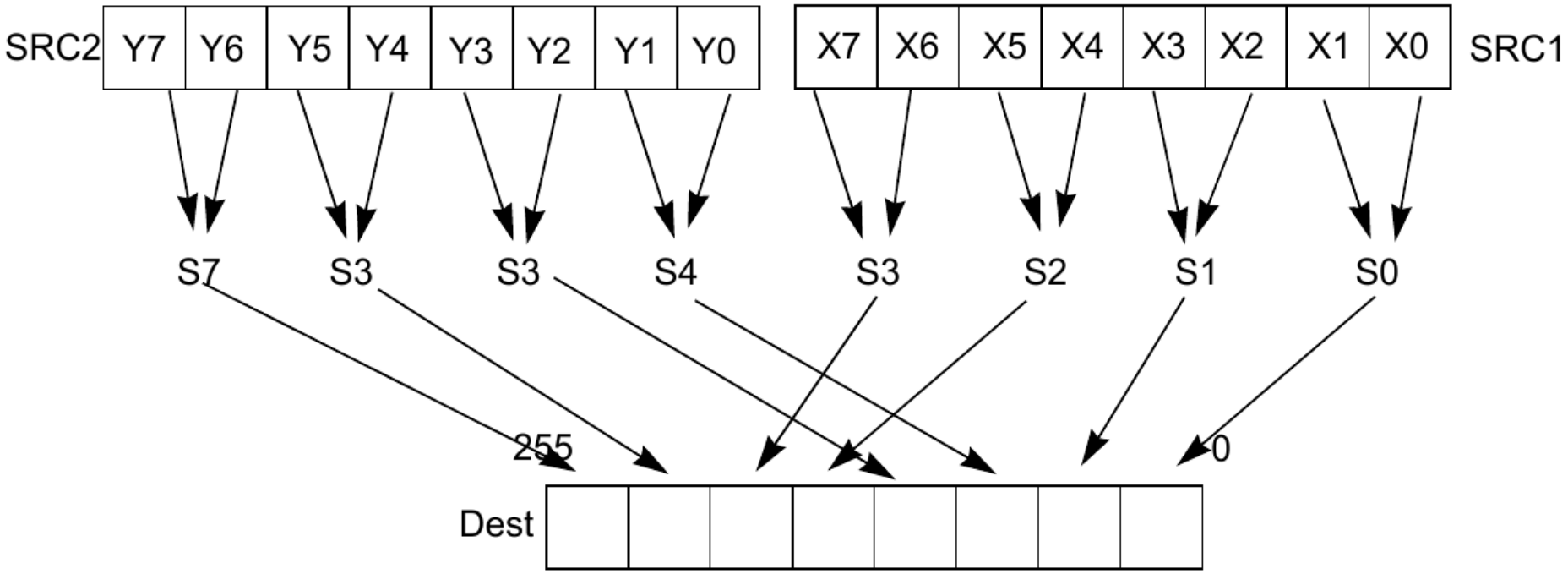


Рисунок — Горизонтальное перемещение данных при выполнении инструкции **PHADDD**

Есть шесть горизонтальных инструкций сложения, представленных тремя мнемониками.
Три работают со 128-битными операндами, а другие три — с 64-битными операндами.
Ширина каждого элемента данных составляет 16 или 32 бита.

PHADDW складывает два соседних 16-битных целых числа со знаком по горизонтали от исходного и целевого операндов и упаковывает 16-битные результаты со знаком в целевой операнд.

PHADDSW складывает два соседних 16-битных целых числа со знаком по горизонтали из исходного и целевого операндов в режиме насыщения и упаковывает 16-битные результаты со знаком в целевой операнд.

PHADDD складывает два смежных 32-разрядных целых числа со знаком по горизонтали из исходного и целевого операндов и упаковывает 32-разрядные результаты со знаком в целевой операнд.

Есть шесть команд горизонтального вычитания, представленных тремя мнемониками.
Три работают со 128-битными операндами, а другие три — с 64-битными операндами.
Ширина каждого элемента данных составляет 16 или 32 бита.

PHSUBW выполняет горизонтальное вычитание для каждой смежной пары 16-битовых целых чисел со знаком, вычитая старшее значащее слово из младшего значащего слова каждой пары в исходном и целевом операндах. 16-битные результаты со знаком упаковываются и записываются в операнд назначения.

PHSUBSW выполняет горизонтальное вычитание для каждой смежной пары 16-битовых целых чисел со знаком, вычитая старшее значащее слово из младшего значащего слова каждой пары в исходном и целевом операндах в режиме насыщения. 16-битные результаты со знаком упаковываются и записываются в операнд-адресат.

PHSUBD выполняет горизонтальное вычитание для каждой смежной пары 32-битных целых чисел со знаком, вычитая старшее двойное слово из младшего значащего двойного слова каждой пары в исходном и целевом операндах. 32-битные результаты со знаком упаковываются и записываются в операнд назначения.

Упакованные абсолютные значения

Имеется шесть инструкций вычисления абсолютных значений упакованных чисел, представленных тремя мнемониками.

Три работают со 128-битными операндами, а три другие — с 64-битными операндами.

Ширина элементов данных составляет 8, 16 или 32 бита.

Абсолютное значение каждого элемента данных исходного операнда сохраняется в качестве UNSIGNED результата в операнде-адресате.

PABSB вычисляет абсолютное значение каждого байта данных со знаком.

PABSW вычисляет абсолютное значение каждого 16-битного элемента данных со знаком.

PABSD вычисляет абсолютное значение каждого 32-битного элемента данных со знаком.

Умножение и сложение упакованных байтов со знаком и без знака

Есть две инструкции умножения и сложения упакованных байт со знаком и без знака, представленные одним мнемоническим символом.

Один работает со 128-битными операндами, а другой — с 64-битными.

Умножения выполняются на каждой вертикальной паре элементов данных.

Элементы данных в исходном операнде являются значениями байтов со знаком, элементы входных данных операнда назначения являются значениями байтов без знака.

PMADDUBSW умножает каждое значение байта без знака на соответствующее значение байта со знаком, в результате получается промежуточное 16-разрядное целое число со знаком.

Каждая смежная пара 16-битных значений со знаком складывается по горизонтали с насыщением.

16-битные результаты со знаком упаковываются в операнд-назначение.

Умножение упакованных с округлением и масштабированием[1]

Имеются две инструкции упакованного умножения с округлением и масштабированием, представленные одним мнемоническим символом.

Одна работает со 128-битными операндами, а другая — с 64-битными.

PMULHRSW умножает по вертикали каждое 16-разрядное целое число со знаком из операнда-назначения на соответствующее 16-разрядное целое число со знаком из исходного операнда, создавая промежуточные 32-разрядные целые числа со знаком.

Каждое промежуточное 32-битное целое число усекается до 18 старших битов.

Округление всегда выполняется добавлением 1 к младшему разряду 18-битного промежуточного результата.

Окончательный результат получается путем выбора 16 бит непосредственно справа от самого старшего бита каждого 18-битового промежуточного результата и упаковки в операнд-назначение.

1 Packed Multiply High with Round and Scale

Перемешивание упакованных байтов

Есть две инструкции перестановки упакованных байтов, представленные одним мнемоническим символом.

Одна работает со 128-битными операндами, а другая — с 64-битными.

Операции перемешивания выполняются побайтно для операнда-адресата с использованием исходного операнда в качестве маски управления.

PSHUFB переставляет каждый байт на место в соответствии с маской управления перемешиванием.

Младшие три или четыре бита каждого байта маски управления перемешиванием образуют индекс перестановки.

Маска перемешивания не изменяется.

Если установлен самый старший бит (бит 7) байта управления перемешиванием, в байт результата записывается нулевая константа.

Изменение знака упакованных чисел

Имеется шесть инструкций с упакованными знаками, представленные тремя мнемониками. Три работают со 128-битными операндами, а три другие — с 64-битными.

Ширина каждого элемента данных для этих инструкций составляет 8-, 16- или 32-разрядные целые числа со знаком.

PSIGNB/W/D меняет знак каждого целочисленного элемента со знаком целевого операнда, если соответствующий элемент данных в исходном операнде меньше нуля.

Выравнять вправо упакованные числа (Packed Align Right)

Есть две инструкции pack-align-right, представлены одной мнемоникой.

Одна работает со 128-битными операндами, а другая — с 64-битными.

Эти инструкции объединяют целевой и исходный операнды в композицию и извлекают результат из композиции в соответствии с указанной константой.

Исходный операнд инструкции **PALIGNR** приписывается справа за операндом-адресатом, образуя промежуточное значение, вдвое превышающее ширину исходного операнда.

Результат извлекается из промежуточного значения в операнд-адресат путем выбора 128-битного или 64-битного значения, выровненного по правому краю относительно байтового смещения, указанного в непосредственном значении.

Обзор SSE4

SSE4 состоит из двух наборов расширений — SSE4.1 и SSE4.2.

SSE4.1 предназначен для повышения производительности рабочих нагрузок мультимедиа, обработки изображений и 3D. SSE4.1 добавляет инструкции, которые улучшают векторизацию, выполняемую компилятором, и значительно увеличивают поддержку вычислений с упакованными двойными словами. Технология также предоставляет подсказки (hints), которые могут улучшить пропускную способность памяти при чтении из некэшируемого типа памяти WC[2].

47 инструкций SSE4.1 включают:

- Две инструкции выполняют умножение упакованных двойных слов.
- Две инструкции выполняют скалярные произведения с плавающей запятой с выбором ввода/вывода.
- Одна инструкция выполняет загрузку с подсказкой потоковой передачи.
- Шесть инструкций упрощают смешивание упакованных.
-Восемь инструкций расширяют поддержку инструкций MIN/MAX для упакованных целых.
- Четыре инструкции предназначены для округления с плавающей запятой с возможностью выбора режима округления и отмены исключения потери точности.
- Семь инструкций улучшают вставку и извлечение данных из регистров XMM
- Двенадцать инструкций улучшают преобразование упакованных целочисленных форматов (знаковое и нулевое расширения).

---

2 Комбинирование записи (WC) — это метод работы компьютерной шины, позволяющий объединять данные и временно сохранять их в буфере — буфере объединения записи (WCB) — для последующей выдачи (burst) в пакетном режиме вместо записи (немедленно) отдельными битами или небольшими порциями.

- Одна инструкция улучшает генерацию SAD (суммы абсолютных разностей) для блоков малых размеров.
- Одна инструкция помогает горизонтальным поисковым операциям.
- Одна инструкция улучшает маскированные сравнения.
- Одна инструкция добавляет сравнение на равенство упакованных `qword`.
- Одна инструкция добавляет упаковку двойного слова с беззнаковым насыщением.

Инструкции SSE4.2, работающие с регистрами XMM, улучшают производительность в следующих областях:

- Обработка строк и текста, которая может использовать преимущества методов программирования с использованием нескольких данных с одной инструкцией.
- Целочисленная инструкция SIMD, которая расширяет возможности 128-битной целочисленной SIMD в SSE4.1.