**МИКРОАРХИТЕКТУРА PENTIUM 4**

**NetBurst (ПОТОКОВО-СЕТЕВАЯ) МИКРОАРХИТЕКТУРА**

ОРИЕНТИРОВАНА НА ЭФФЕКТИВНУЮ ОБРАБОТКУ ВЫСОКОСКОРОСТНЫХ ПОТОКОВ ДАННЫХ:

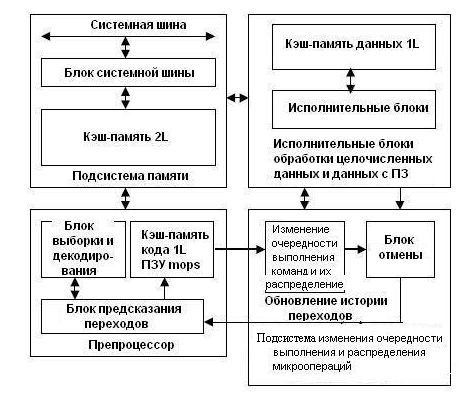
1. ИНТЕРНЕТ-ПРИЛОЖЕНИЯ
2. МУЛЬТИМЕДИЙНЫЕ ПРИЛОЖЕНИЯ

3. РАЗВИТИЕ SIMD ТЕХНОЛОГИЙ, РЕАЛИЗОВАННЫХ В БОЛЕЕ РАННИХ ПРОЦЕССОРАХ IA-32

**ОСОБЕННОСТИ МИКРОАРХИТЕКТУРЫ NetBurst PENTIUM 4**

1. ГАРВАРДСКАЯ АРХИТЕКТУРА – РАЗДЕЛЬНЫЕ КЭШи КОМАНД (ДЕКОДИРОВАННЫХ) И ДАННЫХ 1-го УРОВНЯ
2. ГИПЕРКОНВЕЙЕРНАЯ ОБРАБОТКА – БОЛЬШОЕ КОЛИЧЕСТВО СТУПЕНЕЙ КОНВЕЙЕРА (20 СТУПЕНЕЙ) – ПОВЫШЕНИЕ ТАКТОВОЙ ЧАСТОТЫ РАБОТЫ ПРОЦЕССОРА
3. СУПЕРСКАЛЯРНАЯ АРХИТЕКТУРА – ОДНОВРЕМЕННОЕ ИСПОЛНЕНИЕ НЕСКОЛЬКИХ КОМАНД В ПАРАЛЛЕЛЬНО РАБОТАЮЩИХ ИСПОЛНИТЕЛЬНЫХ БЛОКАХ
4. ПРЕДСКАЗАНИЕ ПЕРЕХОДОВ (СТАТИЧЕСКОЕ И ДИНАМИЧЕСКОЕ)
5. ДИНАМИЧЕСКОЕ ИЗМЕНЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ВЫПОЛНЕНИЯ КОМАНД (СПЕКУЛЯТИВНОЕ ВЫПОЛНЕНИЕ – ВЫПОЛНЕНИЕ КОМАНД С ОПЕРЕЖЕНИЕМ)
6. ВСТРОЕННАЯ КЭШ-ПАМЯТЬ 2-ГО УРОВНЯ (L2)
7. УСОВЕРШЕНСТВОВАННЫЙ БЛОК ОБРАБОТКИ ДАННЫХ SSE-2 (SIMD)

**УКРУПНЕННАЯ СТРУКТУРА PENTIUM 4**

****

ОСНОВНЫЕ БЛОКИ МИКРОАРХИТЕКТУРЫ:

1.ПОДСИСТЕМА ПАМЯТИ

2. ПРЕПРОЦЕССОР ВЫБОРКИ/ДЕКОДИРОВАНИЯ КОМАНД

3. ПОДСИСТЕМА ИЗМЕНЕНИЯ ОЧЕРЕДНОСТИ ВЫПОЛНЕНИЯ И РАСПРЕДЕЛЕНИЯ МИКРООПЕРАЦИЙ ПО ИСПОЛНИТЕЛЬНЫМ БЛОКАМ

4. ПОДСИСТЕМА ПАРАЛЛЕЛЬНО РАБОТАЮЩИХ ИСПОЛНИТЕЛЬНЫХ БЛОКОВ

**ПОДСИСТЕМА ПАМЯТИ**

* КЭШ ПАМЯТЬ 2-го УРОВНЯ ЯВЛЯЕТСЯ ОБЩЕЙ ДЛЯ КОМАНД И ДАННЫХ
* 8-ми ВХОДОВАЯ МНОЖЕСТВЕННО-АССОЦИАТИВНАЯ КЭШ ПАМЯТЬ С ДЛИНОЙ СТРОКИ 128 БАЙТ
* ПРИ ОТСУТСТВИИ ИНФОРМАЦИИ В КЭШе ВЫПОЛНЯЮТСЯ ДВА 64-ти РАЗРЯДНЫХ ЦИКЛА ОБРАЩЕНИЯ К ОЗУ (ОБМЕН ДАННЫМИ С ПАМЯТЬЮ ОСУЩЕСТВЛЯЕТСЯ СТРОКАМИ!!!)
* ИСПОЛЬЗУЕТСЯ ОТЛОЖЕННАЯ ЗАПИСЬ
* БЛОК СИСТЕМНОЙ ШИНЫ (FSB) ДЛЯ СВЯЗИ С ОСНОВНОЙ ПАМЯТЬЮ И ВНЕШНИМИ УСТРОЙСТВАМИ

**ХАРАКТЕРИСТИКИ СИСТЕМНОЙ ШИНЫ**

* ЭКВИВАЛЕНТНАЯ ЧАСТОТА РАБОТЫ – 400 Мгц
* 64-х РАЗРЯДНАЯ ШИНА ДАННЫХ СО СКОРОСТЬЮ ОБМЕНА 3.2 Гбайт/Сек
* 41-ой РАЗРЯДНАЯ ШИНА АДРЕСА С ВОСЕМЬЮ ДЕКОДИРОВАННЫМИ МЛАДШИМИ РАЗРЯДАМИ (ДЛЯ ВЫБОРА БАЙТОВ)
* АДРЕСНОЕ ПРОСТРАНСТВО ВНЕШНЕЙ ПАМЯТИ - 64 Гбайт

**ПРЕПРОЦЕССОР ВЫБОРКИ/ДЕКОДИРОВАНИЯ**

* ИЗВЛЕКАЕТ СЛЕДУЮЩУЮ КОМАНДУ И ПОДГОТАВЛИВАЕТ ЕЕ ДЛЯ КОНВЕЙЕРНОЙ ОБРАБОТКИ - ОБЕСПЕЧИВАЕТ ФОРМИРОВАНИЕ ВЫСОКОСКОРОСТНОГО ПОТОКА ДЕКОДИРОВАННЫХ КОМАНД
* ИМЕЕТ В СВОЕМ СОСТАВЕ БЛОК ВЫСОТОЧНОГО ДИНАМИЧЕСКОГО ПРЕДСКАЗАНИЯ ПЕРЕХОДОВ, ОСНОВАННОГО НА ХРАНЕНЕНИИ И ОБНОВЛЕНИИ ПРЕДЫСТОРИИ ДЛЯ СПЕКУЛЯТИВНОГО («ПО ПРЕДПОЛОЖЕНИЮ») ВЫПОЛНЕНИЯ КОМАНД ПЕРЕХОДОВ
* ПО ПРЕДСКАЗАННОМУ АДРЕСУ КОМАНДА ВЫБИРАЕТСЯ ИЗ КЭШ 2L И ДЕКОДИРУЕТСЯ В БАЗОВЫЕ ОПЕРАЦИИ (ПРИМИТИВЫ - МИКРООПЕРАЦИИ), КОТОРЫЕ МОГУТ БЫТЬ ВЫПОЛНЕНЫ В ИСПОЛНИТЕЛЬНЫХ БЛОКАХ
* ПРЕДВАРИТЕЛЬНО МИКРООПЕРАЦИИ ПОМЕЩАЮТСЯ В КЭШ КОМАНД 1L (TRACE CACHE) - ИЗ КЭШа КОМАНДЫ ПОВТОРНО ИЗВЛЕКАЮТСЯ В ДЕКОДИРОВАННОМ ВИДЕ, ЧТО УСКОРЯЕТ ИХ ОБРАБОТКУ (ЭТО ОДНА ИЗ ОСНОВНЫХ ОСОБЕННОСТЕЙ NetBurst МИКРОАРХИТЕКТУРЫ)

**БЛОК ИСПОЛНИТЕЛЬНЫХ УСТРОЙСТВ**

* КЭШ-ПАМЯТЬ ДАННЫХ 1L – ИМЕЕТ ОТДЕЛЬНЫЕ ПОРТЫ ДЛЯ ЧТЕНИЯ И ЗАПИСИ
* ЗА ОДИН ТАКТ МОГУТ ВЫБИРАТЬСЯ ОПЕРАНДЫ ДЛЯ 2-Х КОМАНД (DDR – СИНХРОНИЗАЦИЯ ОБМЕНА ПО ФРОНТУ И СПАДУ ТАКТОВОГО ИМПУЛЬСА)
* ИСПОЛНИТЕЛЬНЫЕ БЛОКИ РАЗЛИЧНЫХ ТИПОВ ДЛЯ ОБРАБОТКИ ЦЕЛОЧИСЛЕННЫХ ДАННЫХ И ДАННЫХ С ПЗ
* ДЛЯ ХРАНЕНИЯ ОПЕРАНДОВ И РЕЗУЛЬТАТОВ ИСПОЛЬЗУЮТСЯ РЕГИСТРОВЫЕ ФАЙЛЫ (БЛОКИ РЕГИСТРОВ ЗАМЕЩЕНИЯ – ОТДЕЛЬНО ДЛЯ ЦЕЛОЧИСЛЕННЫХ АЛУ И FPU)
* РЕГИСТРОВЫЕ ФАЙЛЫ ИСПОЛЬЗУЮТСЯ ДЛЯ ПЕРЕИМЕНОВАНИЯ РЕГИСТРОВ В КОМАНДАХ С ЦЕЛЬЮ УСТРАНЕНИЯ КОНФЛИКТОВ WAW И WAR

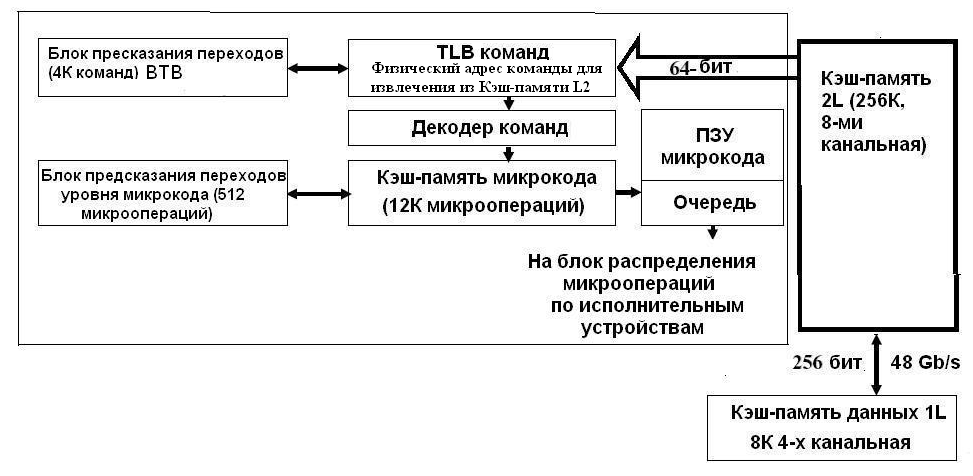
**КЭШ-ПАМЯТЬ ДАННЫХ 1-го УРОВНЯ**

* ИСПОЛЬЗУЕТСЯ ДЛЯ ХРАНЕНИЯ ДАННЫХ ЦЕЛОЧИСЛЕННОЙ, ПЗ (ПЛАВАЮЩЕЙ ЗАПЯТОЙ) И SSE2 ОБРАБОТКИ
* МНОЖЕСТВЕННО-АССОЦИАТИВНЫЙ 4-х КАНАЛЬНЫЙ КЭШ ОБЪЕМОМ 8Кбайт, 64 БАЙТА В СТРОКЕ
* СКВОЗНАЯ ЗАПИСЬ, ПРИ ЗАПИСИ В ЭТОТ КЭШ КОПИЯ ПОМЕЩАЕТСЯ В КЭШ 2-го УРОВНЯ ЗА 1 ТАКТ
* ЗАГРУЗКА В КЭШ ЗА 2 ТАКТА ДЛЯ ЦЕЛОЧИСЛЕННЫХ ДАННЫХ И 6 ТАКТОВ ДАННЫЕ С ПЗ И ДЛЯ SSE2

**БЛОК ИЗМЕНЕНИЯ ОЧЕРЕДНОСТИ ВЫПОЛНЕНИЯ И РАСПРЕДЕЛЕНИЯ МИКРООПЕРАЦИЙ (ПЛАНИРОВЩИК)**

* НАЗНАЧЕНИЕ БЛОКА – ОБЕСПЕЧИТЬ МАКСИМАЛЬНУЮ ЗАГРУЗКУ (КОЭФФИЦИЕНТ ИСПОЛЬЗОВАНИЯ) ИСПОЛНИТЕЛЬНЫХ БЛОКОВ (ДИНАМИЧЕСКОЕ ПЛАНИРОВАНИЕ)
* ОПТИМИЗИРУЕТ ПОТОК ДЕКОДИРОВАННЫХ КОМАНД ПУТЕМ ИЗМЕНЕНИЯ ПОРЯДКА ИХ ВЫПОЛНЕНИЯ ПО ПРИНЦИПУ ГОТОВНОСТИ ОПЕРАНДОВ И НЕЗАВИСИМОСТИ КОМАНД
* ПОЛУЧАЕТ ОТ ИСПОЛНИТЕЛЬНЫХ УСТРОЙСТВ ИНФОРМАЦИЮ О ЗАВЕРШЕНИИ ВЫПОЛНЕНИЯ КОМАНД И СВЯЗЫВАЕТ ИХ ПО ПОЛУЧЕННЫМ РЕЗУЛЬТАТАМ
* ВОССТАНАВЛИВАЕТ ПОРЯДОК ВЫПОЛНЕНИЯ КОМАНД, ПРЕДУСМОТРЕННЫЙ ПРОГРАММОЙ
* ОБЕСПЕЧИВАЕТ ОБНОВЛЕНИЕ ПРЕДЫСТОРИИ ПЕРЕХОДОВ

**ДЕТАЛИЗАЦИЯ СТРУКТУРЫ ПРЕПРОЦЕССОРА**

****

**Кэш-память микрокода**

* КЭШ-ПАМЯТЬ МИКРОКОДА ВЫДАЕТ ДО 3-х МИКРООПЕРАЦИЙ ЗА ТАКТ (ГРУППИРОВАНИЕ МИКРООПЕРАЦИЙ) ДЛЯ ПЕРЕДАЧИ В БЛОК ИЗМЕНЕНИЯ ПОРЯДКА ВЫПОЛНЕНИЯ И РАСПРЕДЕЛЕНИЯ КОМАНД
* НАЛИЧИЕ ДЕКОДИРОВАННОЙ КОМАНДЫ В КЭШЕ НЕ ТРЕБУЕТ ПОВТОРНОГО ДЕКОДИРОВАНИЯ КОМАНДЫ И УПРОщает РЕАЛИЗАЦИЮ ДЕКОДЕРА
* ПРИ ДЕКОДИРОВАНИИ КОМАНД ВЕТВЛЕНИЯ ВКЛЮЧАЕТСЯ МЕХАНИЗМ ПРЕДСКАЗАНИЯ ПЕРЕХОДОВ, КОТОРЫЙ ФОРМИРУЕТ АДРЕС СЛЕДУЮЩЕЙ КОМАНДЫ ДО ВЫЧИСЛЕНИЯ УСЛОВИЯ ПЕРЕХОДА
* КОМАНДА ПО ПРЕДСКАЗАННОМУ АДРЕСУ ПЕРЕХОДА ДЕКОДИРУЕТСЯ И ВКЛЮЧАЕТСЯ В ПОТОК МИКРООПЕРАЦИЙ

**БЛОК ПРЕДСКАЗАНИЯ ПЕРЕХОДОВ УРОВНЯ МИКРОКОДА**

* ВЫПОЛНЯЕТ ПРЕДСКАЗАНИЕ ПЕРЕХОДОВ ДЛЯ МИКРОКОДОВ, НАХОДЯЩИХСЯ В КЭШе В ТЕКУЩИЙ МОМЕНТ ВРЕМЕНИ
* ИМЕЕТ СТЕК ГЛУБИНОЙ 16 ДЛЯ ХРАНЕНИЯ АДРЕСОВ ВОЗВРАТА ИЗ ПРОЦЕДУР

**ПЗУ МИКРОКОДА**

* ПЗУ ИСПОЛЬЗУЕТСЯ ДЛЯ ОБРАБОТКИ СЛОЖНЫХ КОМАНД (СТРОКОВЫЕ КОМАНДЫ ТРЕБУЮТ ОКОЛО 1000 МИКРООПЕРАЦИЙ)
* ДЛЯ СЛОЖНЫХ КОМАНД НЕОБХОДИМЫЕ МИКРООПЕРАЦИИ ИЗВЛЕКАЮТСЯ ЧАСТИЧНО ИЗ ПЗУ И ЧАСТИЧНО ИЗ КЭШ-ПАМЯТИ МИКРОКОДА
* ВСЕ МИКРООПЕРЦИИ (ПЗУ МИКРОКОДА, КЭШ-ПАМЯТИ МИКРОКОДА) ПЕРЕД ПЕРЕДАЧЕЙ В ПОДСИСТЕМУ ИЗМЕНЕНИЯ ОЧЕРЕДНОСТИ ВЫПОЛНЕНИЯ КОМАНД ПОМЕЩАЮТСЯ В ОЧЕРЕДЬ (БУФЕР МИКРООПЕРАЦИЙ)
* БУФЕР ИСПОЛЬЗУЕТСЯ ДЛЯ ВЫРАВНИВАНИЯ ПОТОКА МИКРОКОМАНД ВО ВРЕМЕНИ(СОГЛАСОВАНИЕ СКОРОСТЕЙ ВЫДАЧИ МИКРООПЕРАЦИЙ И ИХ ПЕРЕДАЧИ НА ИСПОЛНЕНИЕ)

**TLB КОМАНД (БУФЕР АССОЦИАТИВНОЙ ТРАНСЛЯЦИИ СТРАНИЧНОГО АДРЕСА)**

* ИСПОЛЬЗУЕТСЯ В СЛУЧАЕ ОТСУТСТВИЯ НУЖНЫХ МИКРООПЕРАЦИЙ В КЭШе МИКРОКОДОВ – ХРАНИТ БАЗОВЫЕ АДРЕСА АКТУАЛЬНЫХ СТРАНИЦ
* ТРАНСЛИРУЕТ ЛИНЕЙНЫЙ АДРЕС КОМАНДЫ В ФИЗИЧЕСКИЙ АДРЕС ДЛЯ ОБРАЩЕНИЯ К КЭШ-ПАМЯТИ 2-го УРОВНЯ
* ВЫПОЛНЯЕТ КОНТРОЛЬ ПРАВ ДОСТУПА (УРОВНЕЙ ПРИВИЛЕГИЙ) ПРИ СТРАНИЧНОЙ АДРЕСАЦИИ

**БЛОК ПРЕДСКАЗАНИЯ ПЕРЕХОДОВ (ВТВ – BRANCH TARGET BUFFER)**

* ПРЕДНАЗНАЧЕН ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ КОНВЕЙЕРНОЙ ОБРАБОТКИ ЗА СЧЕТ СОКРАЩЕНИЯ ПЕРЕЗАГРУЗОК КОНВЕЙЕРА ДЛЯ ВЫБОРКИ КОМАНДЫ ИЗ АЛЬТЕРНАТИВНОЙ ВЕТВИ
* ХРАНИТ ТАБЛИЦУ АДРЕСОВ ПЕРЕХОДОВ РАНЕЕ ВЫПОЛНЕННЫХ КОМАНД УСЛОВНОГО ПЕРЕХОДА (ТОЛЬКО ПРИ ВЫПОЛНЕНИИ ПЕРЕХОДА!)
* ХРАНИТ БИТЫ ПРЕДЫСТОРИИ ВЕТВЛЕНИЙ ДЛЯ ДИНАМИЧЕСКОГО ПРЕДСКАЗАНИЯ ПОСЛЕДУЮЩИХ ПЕРЕХОДОВ (ВЫПОЛНЯЛСЯ ЛИ ПЕРЕХОД ПРИ ПРЕДШЕСТВУЮЩИХ ВЫПОЛНЕНИЯХ КОМАНДЫ)
* БЛОК НА ОСНОВЕ РЕЗУЛЬТАТОВ ПРЕДСКАЗАНИЯ ПОЗВОЛЯЕТ НАЧАТЬ ИЗВЛЕЧЕНИЕ И ВЫПОЛНЕНИЕ КОМАНДЫ ДО ОПРЕДЕЛЕНИЯ УСЛОВИЯ ПЕРЕХОДА
* ЕСЛИ В БЛОКЕ НЕТ ИНФОРМАЦИИ О КОМАНДЕ ПЕРЕХОДА, ТО ВЫПОЛНЯЕТСЯ СТАТИЧЕСКОЕ ПРЕДСКАЗАНИЕ ПЕРЕХОДА (НАПРИМЕР, СТРАТЕГИИ – ПЕРЕХОД ВСЕГДА ЕСТЬ, ИЛИ ПЕРЕХОДА НИКОГДА НЕТ)

**ПРИНЦИП РАБОТЫ ВТВ**

* КЭШ-ПАМЯТЬ НЕБОЛЬШОГО ОБЪЕМА (212 ЦЕЛЕВЫХ АДРЕСОВ РАНЕЕ ВЫПОЛНЕННЫХ ПЕРЕХОДОВ В КОМАНДАХ УСЛОВНОГО ПЕРЕХОДА )
* ТЭГ – АДРЕС КОМАНДЫ
* ПЕРЕД ВЫБОРКОЙ КОМАНДЫ ЕЕ АДРЕС СРАВНИВАЕТСЯ С ТЭГАМИ ВТВ
* ЕСЛИ КОМАНДА НАЙДЕНА В ВТВ, ТО АНАЛИЗИРУЕТСЯ ПРЕДЫСТОРИЯ ВЕТВЛЕНИЙ. ПРИ ПРЕДСКАЗАНИИ ПЕРЕХОДА ЦЕЛЕВОЙ АДРЕС НЕ ВЫЧИСЛЯЕТСЯ, А БЕРЕТСЯ ИЗ ВТВ (СОКРАЩАЕТСЯ ЦИКЛ КОМАНДЫ НА 1 ТАКТ)
* НЕ ДОЖИДАЯСЬ ФОРМИРОВАНИЯ ПРОВЕРЯЕМОГО УСЛОВИЯ В КОМАНДЕ ПЕРЕХОДА ЗАГРУЖАЕТСЯ КОМАНДА ПО ЦЕЛЕВОМУ АДРЕСУ (СПЕКУЛЯТИВНАЯ ВЫБОРКА И ВЫПОЛНЕНИЕ)
* ЕСЛИ В ВТВ КОМАНДЫ НЕ ОКАЗАЛОСЬ, ОНА ВЫПОЛНЯЕТСЯ СТАНДАРТНЫМ ОБРАЗОМ
* ЕСЛИ ЭТО БЫЛА КОМАНДА ПЕРЕХОДА , ТО ПЕРЕХОД НЕ ПРЕДСКАЗЫВАЕТСЯ, А ПРИ ЕГО ВЫПОЛНЕНИИ ПО СФОРМИРОВАННОМУ УСЛОВИЮ АДРЕС ПЕРЕХОДА С ТЭГОМ ЗАНОСИТСЯ В ВТВ (ЗАМЕЩЕНИЕ ПО АЛГОРИТМУ LRU)

**ДЕКОДЕР КОМАНД**

* ПРЕОБРАЗУЕТ КОДЫ КОМАНД, СЧИТЫВАЕМЫЕ ИЗ КЭШ-ПАМЯТИ 2-го УРОВНЯ В МИКРОКОД ДЛЯ ПЕРЕДАЧИ ИСПОЛНИТЕЛЬНЫМ БЛОКАМ
* ЗА ОДИН ТАКТ ДЕКОДИРУЕТСЯ ОДНА КОМАНДА
* НЕСКОЛЬКО КОМАНД МОГУТ БЫТЬ ПРЕОБРАЗОВАНЫ В ОДНУ МИКРООПЕРАЦИЮ (ГРУППИРОВАНИЕ МИКРООПЕРАЦИЙ)
* ОДНА КОМАНДА МОЖЕТ БЫТЬ ПРЕОБРАЗОВАНА В НЕСКОЛЬКО МИКРООПЕРАЦИЙ (ПОСЛЕДОВАТЕЛЬНОСТЬ МИКРООПЕРАЦИЙ)
* ЕСЛИ ДЛЯ ВЫПОЛНЕНИЯ КОМАНДЫ ТРЕБУЕТСЯ БОЛЕЕ 4-х МИКРООПЕРАЦИЙ, ДЛЯ ДЕКОДИРОВАНИЯ ИСПОЛЬЗУЕТСЯ ПЗУ МИКРОКОДА

**Блок распределения регистров**

* ВХОДИТ В СОСТАВ ПОДСИСТЕМЫ ИЗМЕНЕНИЯ ОЧЕРЕДНОСТИ ВЫПОЛНЕНИЯ И РАСПРЕДЕЛЕНИЯ МИКРООПЕРАЦИЙ ПО ИСПОЛНИТЕЛЬНЫМ БЛОКАМ
* выделЯЕТ для каждого АРХИТЕКТУРНОГО (ЛОГИЧЕСКОГО)регистра (рон, fp, mmx, sse) замещающий регистр из СООТВЕТСТВУЮЩЕГО регистрового файла (128 регистров)
* Допускает Одновременное или с измененной последовательностью выполнение команд, использующих одинаковые АРХИТЕКТУРНЫЕ регистры
* Обеспечивает Минимизацию конфликтов в конвейерах

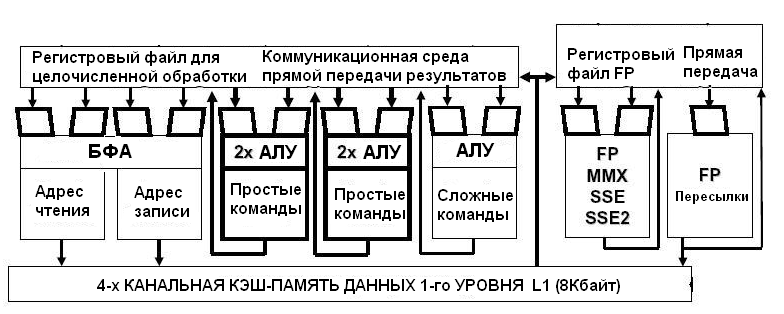
**ОЧЕРЕДЬ МИКРООПЕРАЦИЙ**

* ПОСЛЕ ВЫПОЛНЕНИЯ ЗАМЕЩЕНИЯ АРХИТЕКТУРНЫХ РЕГИСТРОВ СФОРМИРОВАННЫЕ МИКРООПЕРАЦИИ БУФЕРИРУЮТСЯ В ОЧЕРЕДИ (FIFO ЕМКОСТЬЮ 120 МИКРООПЕРАЦИЙ)
* БОЛЬШОЙ РАЗМЕР ОЧЕРЕДИ ПОВЫШАЕТ ВОЗМОЖНОСТИ МАНИПУЛИРОВАНИЯ МИКРООПЕРАЦИЯМИ ДЛЯ ОПТИМИЗАЦИИ ЗАГРУЗКИ ИСПОЛНИТЕЛЬНЫХ УСТРОЙСТВ
* ОПТИМИЗАЦИЯ ЗАГРУЗКИ:

1. ПАРАЛЛЕЛЬНОЕ ВЫПОЛНЕНИЕ МИКРООПЕРАЦИЙ

2. ИЗМЕНЕНИЕ ПОРЯДКА ВЫПОЛНЕНИЯ МИКРООПЕРАЦИЙ (ГОТОВНОСТЬ ИСПОЛНИТЕЛЬНЫХ УСТРОЙСТВ, ПОДГОТОВЛЕННОСТЬ ОПЕРАНДОВ)

**ДЕТАЛИЗАЦИЯ БЛОКА ИСПОЛНИТЕЛЬНЫХ УСТРОЙСТВ**

****

БФА – БЛОК ФОРМИРОВАНИЯ АДРЕСА ПОДКЛЮЧЕН

К КЭШ-ПАМЯТИ ДАННЫХ 1-го УРОВНЯ И

РЕГИСТРОВОМУ ФАЙЛУ (БЛОК РЕГИСТРОВ ЗАМЕЩЕНИЯ - БРЗ)

**АДРЕСАЦИЯ И ВЫБОРКА ОПЕРАНДОВ**

ДЛЯ ОБРАЩЕНИЯ К ПАМЯТИ БФА В СООТВЕТСТВИИ СО СПОСОБАМИ АДРЕСАЦИИ ДЕКОДИРОВАННЫХ КОМАНД ФОРМИРУЕТ ПОСЛЕДОВАТЕЛЬНОСТИ АДРЕСОВ:

1. 48 АДРЕСОВ ДЛЯ ТРАНЗАКЦИЙ ПАМЯТЬ – РЕГИСТР ЗАМЕЩЕНИЯ В РЕГИСТРОВОМ ФАЙЛЕ (ИЗВЛЕЧЕНИЕ ОПЕРАНДА)

2. 24 АДРЕСА ТРАНЗАКЦИЙ РЕГИСТР ЗАМЕЩЕНИЯ В РЕГИСТРОВОМ ФАЙЛЕ – ПАМЯТЬ (ЗАПИСЬ РЕЗУЛЬТАТА ОПЕРАЦИИ)

**СПЕКУЛЯТИВНАЯ ВЫБОРКА ОПЕРАНДОВ**

* БФА ФОРМИРУЕТ АДРЕСА ОПЕРАНДОВ ДЛЯ КОМАНД ЕЩЕ НЕ ПРИНЯТЫХ НА ИСПОЛНЕНИЕ
* ПРИ ОБРАЩЕНИИ К ПАМЯТИ БФА ВЫДАЕТ ДВА АДРЕСА – ОДИН ДЛЯ ЧТЕНИЯ ОПЕРАНДА ИЗ ПАМЯТИ В БРЗ, ДРУГОЙ ДЛЯ ЗАПИСИ РЕЗУЛЬТАТА ИЗ БРЗ В ПАМЯТЬ
* РЕАЛИЗУЕТСЯ ОПЕРЕЖАЮЩЕЕ ЧТЕНИЕ ДАННЫХ ДЛЯ ПОСЛЕДУЮЩЕЙ ОБРАБОТКИ В ИСПОЛНИТЕЛЬНЫХ БЛОКАХ – СПЕКУЛЯТИВНАЯ ВЫБОРКА

**ПОРТЫ ДЛЯ ПЕРЕДАЧИ МИКРООПЕРАЦИЙ НА ИСПОЛНИТЕЛЬНЫЕ БЛОКИ**

* ДЛЯ ПЕРЕДАЧИ МИКРООПЕРАЦИЙ НА ИСПОЛНИТЕЛЬНЫЕ БЛОКИ ИСПОЛЬЗУЮТСЯ 4 ПОРТА
* ЗА ОДИН ТАКТ ИСПОЛНИТЕЛЬНЫМ БЛОКАМ МОГУТ ПЕРЕДАВАТЬСЯ ДО 6 МИКРООПЕРАЦИЙ

**ИСПОЛНИТЕЛЬНЫЕ БЛОКИ (АЛУ)**

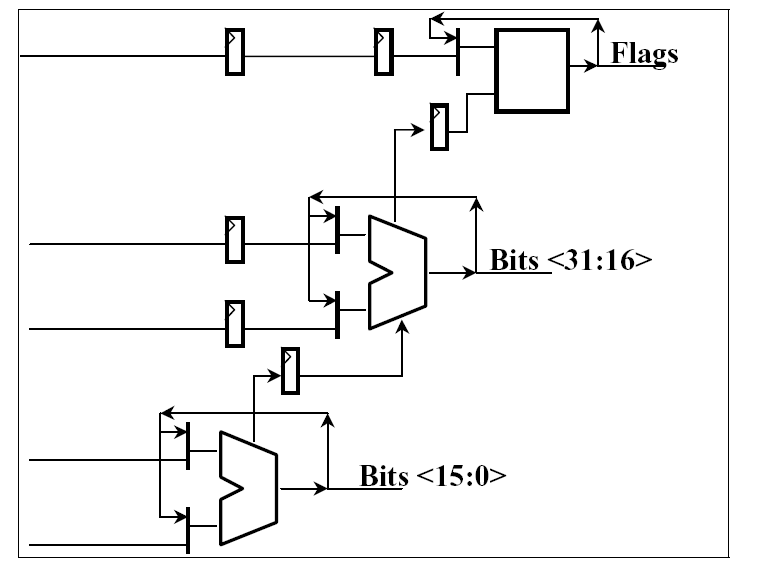
БЫСТРОДЕЙСТВУЮЩИЕ АЛУ (ЧАСТО ВЫПОЛНЯЕМЫЕ КОМАНДЫ):

* ВЫПОЛНЯЮТ ПРОСТЫЕ ОПЕРАЦИИ (МОГУТ ВЫПОЛНЯТЬСЯ ПО ДВЕ ОПЕРАЦИИ ЗА ОДИН ТАКТ – УДВОЕННАЯ ЧАСТОТА)
* ВЫХОДЫ БЫСТРОДЕЙСТВУЮЩИХ АЛУ СОЕДИНЕНЫ ЛИНИЯМИ ПРЯМОЙ ПЕРЕДАЧИ СО ВХОДАМИ – ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ КОНВЕЙЕРНОЙ ОБРАБОТКИ (ИСПОЛЬЗОВАНИЕ ДО ЗАПИСИ В РЕГИСТР РЕЗУЛЬТАТА ВЫПОЛНЕНИЯ КОМАНДЫ В КАЧЕСТВЕ ОПЕРАНДА СЛЕДУЮЩЕЙ КОМАНДЫ)
* ЭТИ ПРЯМЫЕ ПЕРЕДАЧИ РАБОТАЮТ ПРИ ВЫПОЛНЕНИИ 60-70% МИКРООПЕРАЦИЙ

АЛУ ДЛЯ СЛОЖНЫХ КОМАНД:

1. ОПЕРАЦИИ СДВИГА, ЦИКЛИЧЕСКОГО СДВИГА – 4 ТАКТА
2. УМНОЖЕНИЕ, ДЕЛЕНИЕ – ОТ 14 ДО 60 ТАКТОВ

**ОСОБЕННОСТИ ПОСТРОЕНИЯ СУММАТОРА АЛУ**

****

1. 32-х РАЗРЯДНОЕ СУММИРОВАНИЕ ВЫПОЛНЯЕТСЯ

ЗА ТРИ ПОЛУТАКТА ОСНОВНОЙ ЧАСТОТЫ

1. В 1-ом ПОЛУТАКТЕ СУММИРУЮТСЯ МЛАДШИЕ 16 РАЗРЯДОВ
2. ВО 2-ом ПОЛУТАКТЕ СУММИРУЮТСЯ СТАРШИЕ 16 РАЗРЯДОВ
3. В 3-ем ПОЛУТАКТЕ ВЫРАБАТЫВАЮТСЯ ФЛАГИ

5. МЛАДШИЕ 16 РАЗРЯДОВ МОГУТ ИСПОЛЬЗОВАТЬСЯ ПРИ ФОРМИРОВАНИИ

АДРЕСА ДЛЯ ОБРАЩЕНИЯ К КЭШ-ПАМЯТИ ДАННЫХ 1-ГО УРОВНЯ

**ПРЕДПОСЫЛКИ ПОЯВЛЕНИЯ ММХ (MULTIMEDIA EXTENTION) ТЕХНОЛОГИИ ОБРАБОТКИ ДАННЫХ**

АЛГОРИТМЫ ОБРАБОТКИ мультимедийной информации ХАРАКТЕРИЗУЮТСЯ:

* МАЛОРАЗРЯДНЫЕ ЦЕЛОЧИСЛЕННЫЕ данныЕ - 8‑ми разрядные графические пиксели и 16-разрядная оцифровка звука
* короткие циклы с высокими коэффициентами повторяемости
* большое количество операций умножения и суммирования, (алгоритмы цос)
* широкое использование операций с высоким уровнем параллелизма

**Недостатки ТРАДИЦИОННЫХ АРХИТЕКТУР ДЛЯ МУЛЬТИМЕДИА ПРИЛОЖЕНИЙ**

* СТАТИСТИКА: в мультимедийных приложениях 80% времени выполнения программы приходится на 10-20% программного кода
* Малая разрядность МУЛЬТИМЕДИЙНЫХ данных (8 бит) при их обработке на МП IA-32 ПРИВОДИТ К УСЛОЖНЕНИЮ КОДА И СНИЖЕНИЮ ПРОИЗВОДИТЕЛЬНОСТИ ИЗ-ЗА НЕВОЗМОЖНОСТИ СРАЗУ ОБРАБАТЫВАТЬ ПО 4 БАЙТА
* Пример: видеопамятЬ ОБЪЕМОМ 1 Мбайт, каждый пиксель кодируется 1 байтом. Задача: к содержимому каждой ячейки видеопамяти прибавить константу (байт). Необходимо выполнить около 1 млн операций суммирования однобайтовых чисел. Одновременное суммирование 4 пар операндов невозможно в архитектуре IA-32 из-за отсутствия соответствующих команд в системе команд и форматов данных

**ММХ ТЕХНОЛОГИИ**

* В МП Pentium MMX ВПЕРВЫЕ БЫЛ ИНТЕГРИРОВАН БЛОК MMX
* БЛОК ВЫПОЛНЯЕТ однУ КОМАНДУ ММХ СРАЗУ над несколькими ЦЕЛОЧИСЛЕННЫМИ операндами (SIMD ОБРАБОТКА) разрядностью 1, 2, 4 или 8 байт
* существенно повыШАЕТСЯ скорость обработки изображений и звукА
* В МИКРОПРОЦЕССОРЕ Pentium III ИНТЕГРИРОВАН блок SSE (Streaming SIMD Extension — Потоковое SIMD-расширение) для групповой обработки чисел с плавающей точкой (70 КОМАНД)
* для эффективной реализации принципА SIMD необходимы адаптированная система команд и ТИПЫ данных, а также специализированные аппаратные средства

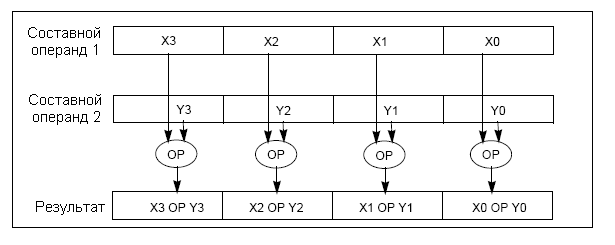
**ОСОБЕННОСТИ СИСТЕМЫ КОМАНД ММХ**

* В систему команд МП Pentium MMX добавлены 57 инструкций, поддерживающих ммх обработку данных. Их использование обеспечивает следующие преимущества:

1. уменьшение времени выполнения мультимедийных приложений

2. Оптимизация многоступенчатой конвейерной обработки команд за счет реализации операций типа сложение с насыщением (пропадает необходимость в командах условного перехода по флагу переполнения)

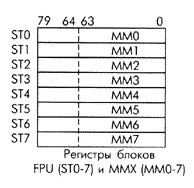
**Общая схема simd работки данных**

****

**Ммх Типы данных (64-Х РАЗРЯДНЫе упакованные, ЦЕЛОЧИСЛЕННЫе)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **64** | | | | | | | |  |  |
|  | | | | | | | | 1 операнд  64 разряда | Пересылки между ММХ регистрами и памятью |
| 32 | | | | 32 | | | |  |  |
|  | | | |  | | | | 2 операнда  по 32 разряда | Арифметические,  логические  операции |
| 16 | | 16 | | 16 | | 16 | |  |
|  | |  | |  | |  | | 4 операнда  по 16 разрядов |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |  |
|  |  |  |  |  |  |  |  | 8 операндов  по 8 разрядов |

**ХРАНЕНИЕ ДАННЫХ ДЛЯ ММХ ОБРАБОТКИ**

****

1. 8 ММХ 64-ти РАЗРЯДНЫХ РЕГИСТРОВ ОТОБРАЖАЮТСЯ НА 80-ти РАЗРЯДНЫЕ РЕГИСТРЫ БЛОКА FPU

2. ОБЕСПЕЧИВАЕТСЯ СОГЛАСОВАННАЯ И С ЧЕРЕДОВАНИЕМ ОБРАБОТКА ИНСТРУКЦИЯМИ ММХ ЦЕЛОЧИСЛЕННЫХ ДАННЫХ И ОБРАБОТКА ЧИСЕЛ С ПЗ В БЛОКЕ FPU

3. ИСПОЛЬЗУЮТСЯ СТАНДАРТНЫЕ МЕХАНИЗМЫ СОХРАНЕНИЯ/ВОССТАНОВЛЕНИЯ РЕГИСТРОВ БЛОКА FPU ПРИ ПЕРЕКЛЮЧЕНИИ ЗАДАЧ

4. ММХ РЕГИСТРЫ НЕ МОГУТ ИСПОЛЬЗОВАТЬСЯ ДЛЯ АДРЕСАЦИИ ПАМЯТИ

**РЕЖИМЫ ДОСТУПА К ММХ РЕГИСТРАМ**

* К ММХ регистрам возможен доступ в режиме 64-х битового или 32-х битового ОБМЕНА:

64-х разрядный обмен:

-обращение к памяти

-пересылки между регистрами ММХ

* 32-х разрядный обмен:

- Обращение к памяти

-Пересылки между РОН и регистрами ММХ

**Примеры КОМАНД ММХ (№1)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **8** | **8** | **8** | **8** | **8** | **8** | **8** | **8** |  |
| -2 | 3 | 4 | -5 | 1 | 3 | 5 | -6 | операнд 1 |
|  |  |  |  |  |  |  |  |  |
| 3 | -1 | 2 | 4 | -2 | 1 | 6 | -7 | операнд 2 |
|  |  |  |  |  |  |  |  |  |
| 3 | 3 | 4 | 4 | 1 | 3 | 6 | -6 | результат |

Команда PMAXSB реализована в блоке SSE микропроцессора Pentium III

Выполняет нахождение максимума одновременно для 8-ми пар однобайтных операндов (максимум для каждой пары операндов)

**Примеры ммх команд (№2)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **8** | **8** | **8** | **8** | **8** | **8** | **8** | **8** |  |
| -2 | 3 | 4 | -5 | 1 | 3 | 5 | -6 | операнд 1 |
|  |  |  |  |  |  |  |  |  |
| 3 | -1 | 2 | 4 | -2 | 1 | 6 | -7 | операнд 2 |
|  |  |  |  |  |  |  |  |  |
| 3 | 3 | 4 | 4 | 1 | 3 | 6 | -6 | результат |

Команда PMAXSB реализована в блоке SSE микропроцессора Pentium III

Выполняет нахождение максимума одновременно для 8-ми пар однобайтных операндов (максимум для каждой пары операндов)

**РАЗВИТИЕ ММХ ТЕХНОЛОГИИ –(SSE РАСШИРЕНИЕ)**

* ТРИ КАТЕГОРИИ КОМАНД SSE:

1. ИСПОЛЬЗОВАНИЕ УПАКОВАННЫХ В 64 ИЛИ 128 БИТНЫЕ ГРУППЫ (СОСТАВНЫЕ ОПЕРАНДЫ) ОПЕРАНДОВ РАЗМЕРОМ 8, 16, 32, 64 БИТА
2. ИСПОЛЬЗОВАНИЕ ОДНОЙ ПАРЫ 32-Х ИЛИ 64-Х БИТОВЫХ ЧИСЕЛ С ПЗ
3. ОБРАБОТКА 4-Х ПАР ВЕЩЕСТВЕННЫХ ЧИСЕЛ ОБЫЧНОЙ ТОЧНОСТИ ИЛИ 2-Х ПАР ЧИСЕЛ С ДВОЙНОЙ ТОЧНОСТЬЮ

* В PENTIUM4 271 КОМАНДА SSE2 – ЛОГИЧЕСКИЕ, АРИФМЕТИЧЕСКИЕ, ПЕРЕСЫЛКИ, СРАВНЕНИЕ, ПЕРЕГРУППИРОВКА.

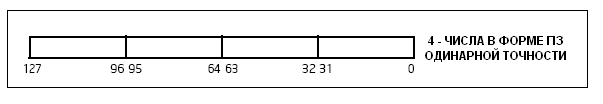
**РЕГИСТРЫ ДЛЯ sse/sse2 ОБРАБОТКИ ДАННЫХ**

ИСПОЛЬЗУЮТСЯ 8 РЕГИСТРОВ:

1. НЕ ОТОБРАЖАЮТСЯ НА АРХИТЕКТУРНЫЕ РЕГИСТРЫ (IA-32)

2. НЕ МОГУТ ИСПОЛЬЗОВАТЬСЯ ДЛЯ АДРЕСАЦИИ ПАМЯТИ

**ОСНОВНОЙ ТИП ДАННЫХ SSE ОБРАБОТКИ**

****

ФОРМАТ ЧИСЛА С ПЗ ОДИНАРНОЙ ТОЧНОСТИ:

1БИТ – ЗНАК

8БИТ – СМЕЩЕННЫЙ ПОРЯДОК

23БИТА – МАНТИССА

КОМАНДЫ ПРЕОБРАЗОВАНИЯ ТИПОВ:

2 ЧИСЛА В ФОРМЕ ПЗ ОДИНАРНОЙ ТОЧНОСТИ – 2 УПАКОВАННЫХ 32-Х РАЗРЯДНЫХ ЦЕЛЫХ

ЧИСЛО В ФОРМЕ ПЗ ОДИНАРНОЙ ТОЧНОСТИ – 32-Х РАЗРЯДНОЕ ЦЕЛОЕ

**Дополнительные Типы ЦЕЛОЧИСЛЕННЫХ данных для обработки в блоке sse/sse2**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **128** | | | | | | | | | | | | | | | |  |
| 64 | | | | | | | | 64 | | | | | | | |  |
|  | | | | | | | |  | | | | | | | | 2 операнда  по 64 разряда |
| 32 | | | | 32 | | | | 32 | | | | 32 | | | |  |
|  | | | |  | | | |  | | | |  | | | | 4 операнда  по 32 разряда |
| 16 | | 16 | | 16 | | 16 | | 16 | | 16 | | 16 | | 16 | |  |
|  | |  | |  | |  | |  | |  | |  | |  | | 8 операндов  по 16 разрядов |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 16 операндов  по 8 разрядов |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Пример команды sse/sse2 ОБРАБОТКИ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **32** | **32** | **32** | **32** |  |
| В3 | В2 | В1 | В0 | операнд 1 |
|  |  |  |  |  |
| D3 | D2 | D1 | D0 | операнд 2 |
|  |  |  |  |  |
| D3 + D2 | D1 + D0 | B3 + B2 | B1 + B0 | результат |

Команда HADDPS – осуществляет горизонтальное сложение 4-х пар операндов с одинарной точностью