Страничное преобразование в INTEL x86

Семинар.

Содержание

Приложение

x86 (англ. *Intel 80x86*) — архитектура процессора и одноимённый набор команд, впервые реализованные в процессорах компании Intel. За время своего существования набор команд постоянно расширялся, сохраняя совместимость с предыдущими поколениями. Помимо Intel, набор команд x86 также реализован в процессорах других производителей: AMD, VIA, Transmeta, IDT, Zhaoxin[2], МЦСТ (в процессорах Эльбрус) и др. В настоящее время для 32-разрядной версии архитектуры существует ещё одно название — IA-32 (Intel Architecture — 32).

1. *Уровни привилегий*

Важнейшим вопросом, решаемым любой ОС является обеспечение защиты самой ОС. В Intel для этого определены четыре кольца защиты (название взято из ОС Multics)

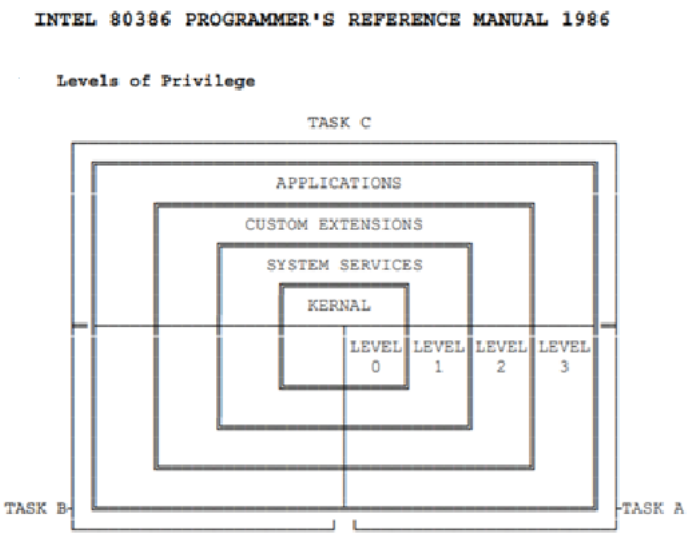


Рис.1

На рис.2 показана система контроля уровня привилегий на примере ображения к данным. Процессор автоматически оценивает доступ к сегменту данных, сравнивая уровни привилегий. Оценка выполняется в то время, когда селектор для дескриптора целевого сегмента загружается в регистр сегмента данных, как показано на рисунке 2,

Для проверки используются три типа уровня привилегий:

1. CPL - текущий уровень привилегий.

2. RPL – запрошенный уровень привилегий в селекторе, используемого для указания целевого сегмента.

3. DPL – уровень привилегий сегмента в дескрипторе целевого сегмента.

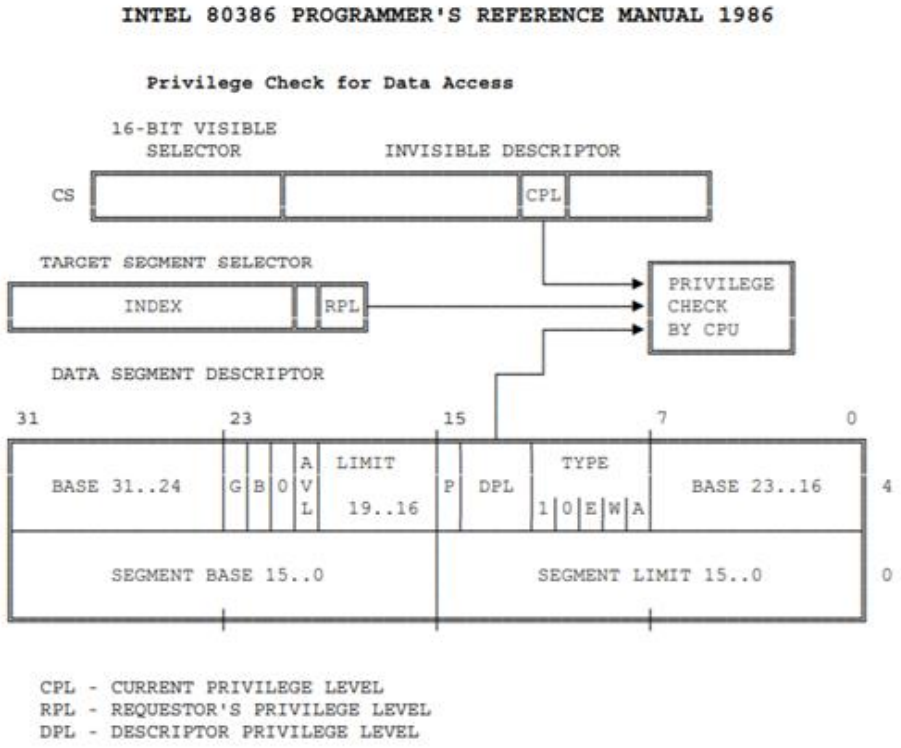


Рис.2

На рисунке показаны все три значения, участвующие в контроле уровня привилегий.

2. Страничное преобразование

Как известно процессоры Intel (386, 486, Pentium) поддерживают две независимые схемы управления виртуальной памятью: управление памятью сегментами по запросу и управление памятью страницами по запросу. Для определения способа управления памятью в процессоре имеется специальный флаг, который находится в управляющем регистре CR0.

В управляющем регистре CR0 бит 0 определяет режим работы процессора – pe (protaction enable), если он установлен, то процессор работает в защищенном режиме; бит 31 определяет включено ли страничное преобразование – pm (paging enable), если он установлен, а если сброшен, то выполняется управление памятью сегментами по запросу.

Причем из анализа флагов можно сделать вывод, что страничное преобразование «включать» и «выключать» можно, а сегментное нельзя. Оно присуще архитектуре.

В соответствии с документацией INTEL сегментация и страничное преобразование связаны, как это показано на следующих рисунках.

2.1 Сегменты и страницы

Документация Intel предлагает обощение для сегментов и страниц при преобразовании виртуального адреса в физический.

Путем соответствующего выбора опций и параметров для обоих

фаз, программное обеспечение для управления памятью может реализовывать несколько различных стилей управление памятью. Мы рассмотрим только плоскую модель памяти.

2.1.1 «Плоская» архитектура

Когда 80386 используется для выполнения программного обеспечения, разработанного для архитектур, которые не имеют сегментов, может оказаться целесообразным эффективно «выключить» особенности сегментации 80386.

80386 не имеет режима, который отключает сегментацию, но тот же эффект может быть достигнут использованием дескрипторов, которые включают все 32-битное линейное адресное пространство.

Для процесса создается локальная таблица сегментов (рис.3), которая содержит только 2 элемента: нулевой дескриптор и дескриптор, описывающий один сегмент размером 4ГБ.

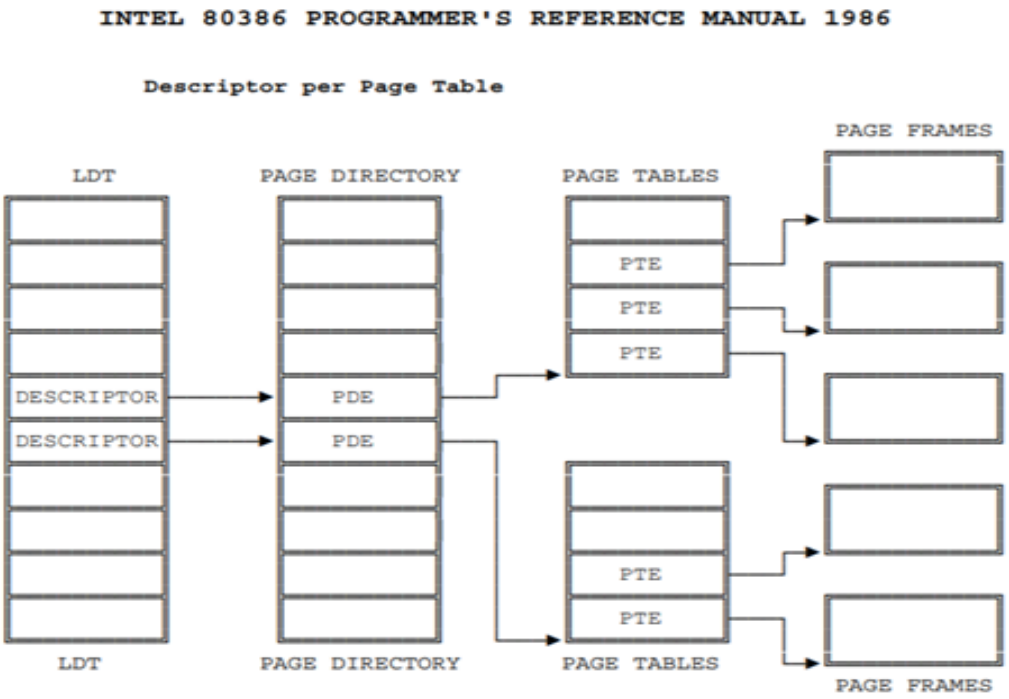


Рис.3

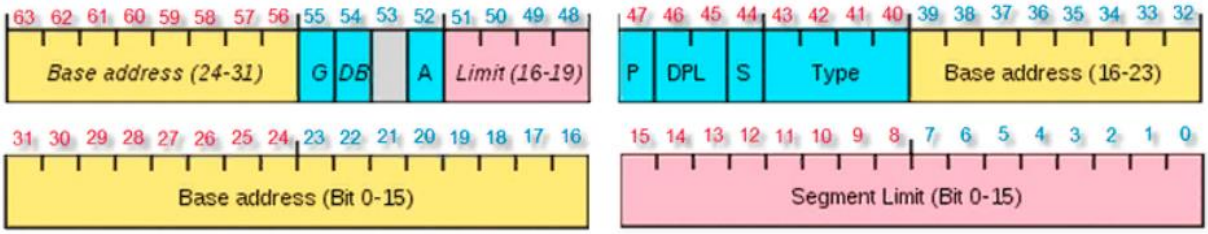


Рис.4

Для сегмента устанавливается:

- начальный адрес – 00000000;
- лимит, равный FFFFF;
- бит гранулярности G = 1;
- бит D, определяющий разрядность дескрипторов, равен 1. (*Значение адреса «вершина» зависит от 54го D бита, если он равен 0, тогда вершина равна 0xFFF(64кб-1), если D бит равен 1, тогда вершина равна 0xFFFFFFFF (4Гб-1)*)

  *С **41-43** бит кодируется тип сегмента.*
  000 — сегмент данных, только считывание
  001 — сегмент данных, считывание и запись
  010 — сегмент стека, только считывание
  011 — сегмент стека, считывание и запись
  100 — сегмент кода, только выполнение
  101- сегмент кода, считывание и выполнение
  110 — подчиненный сегмент кода, только выполнение
  111 — подчиненный сегмент кода, только выполнение и считывание
  *Бит **44** S бит если равен 1 тогда дескриптор описывает реальный сегмент оперативной памяти, иначе значение S бита равно 0.*

32-битные смещения, используемые инструкциями 80386: достаточны для адресации всего линейного адресного пространства.

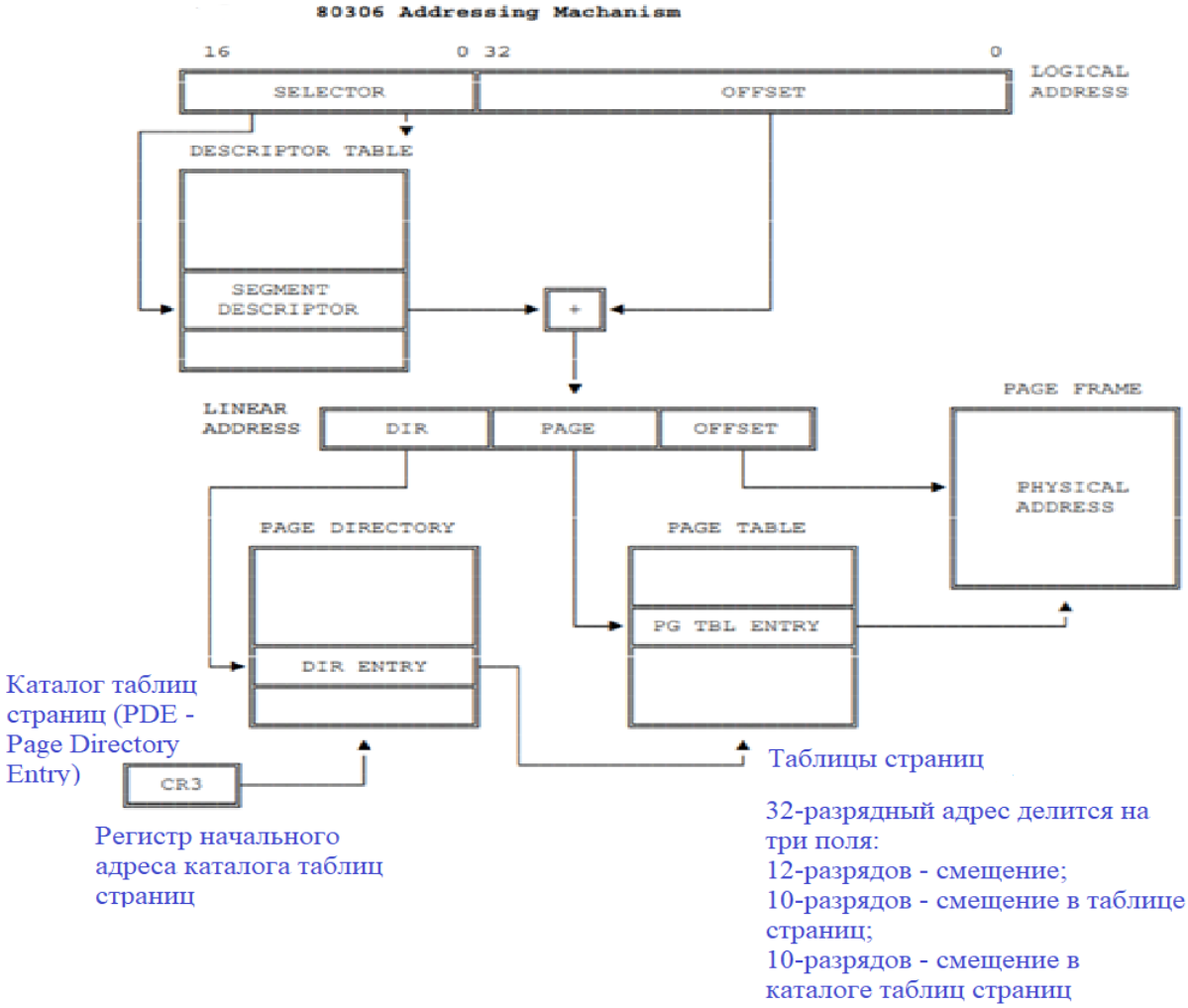
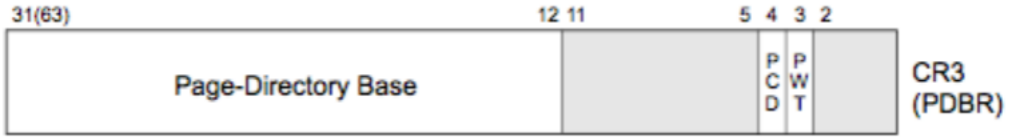


Рис.5

x86 процессоры используют двух уровневую схему страничного преобразования. Такая схема впервые была применена в IBM360/67 и IBM370.

Регистр CR3 всегда содержит начальный адрес каталога таблиц страниц текущего процесса.

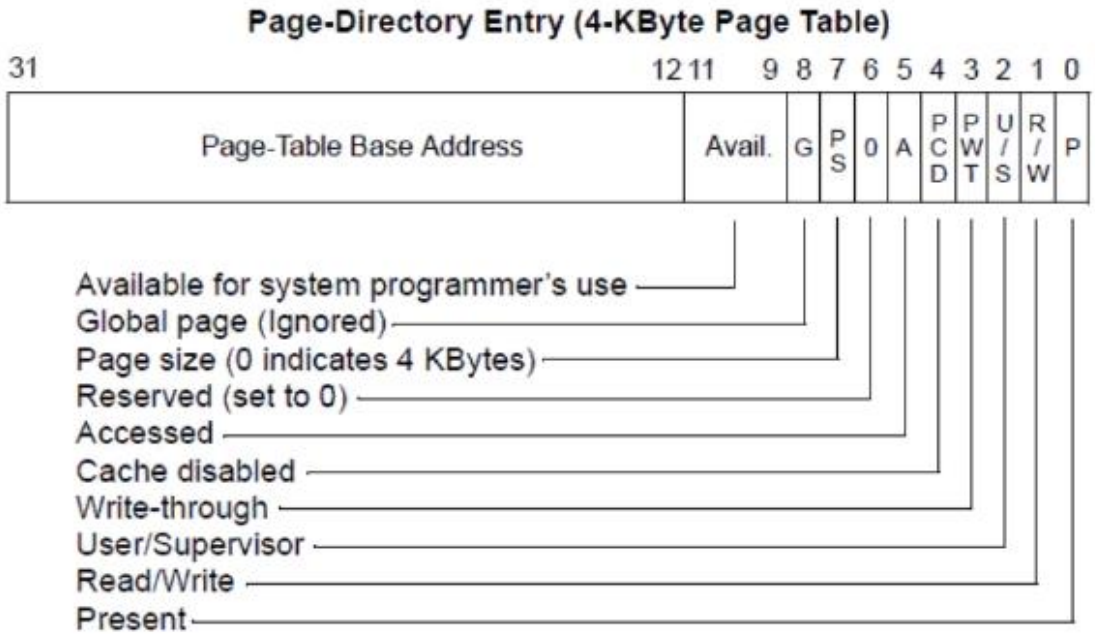


Обратите внимание, что адрес выровнен по границе 4 КБ. Это связано с тем, что адрес фактически указывается только в верхних 20 битах CR3, а затем предполагается, что нижние 12 бит (2 ^ 12 = 4KB) адреса равны 0. (Это оставляет нижние 12 бит CR3 доступными для разного использования, и действительно есть пара флагов для кеширования, которые мы можем игнорировать).

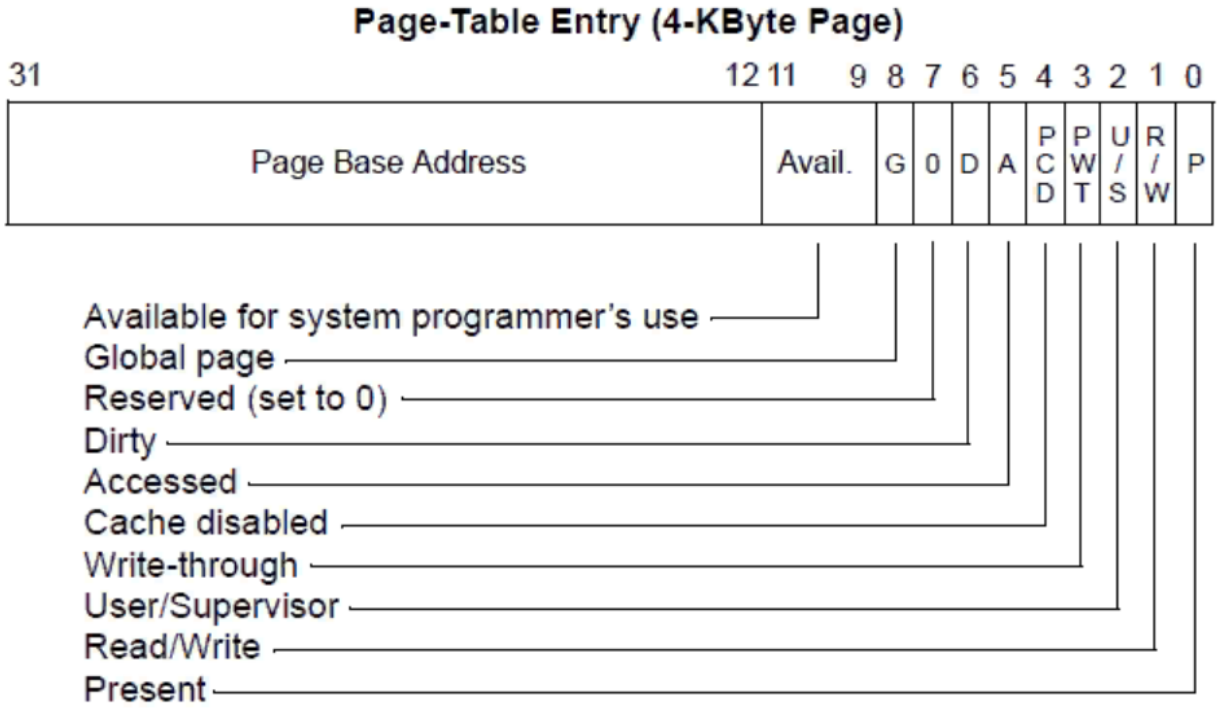
В соответствии с разрядностью адреса для 4 ГБ адресного пространства создается 1024 таблицы страниц. Каталог таблиц страниц содержит 1024 дескриптора таблиц страниц по 4 байта. Дескрипторы страниц так же имеют размер 4 байта. В результате каждая таблица занимает одну четырех

килобайтную страницу (рис.5). Таблицы страниц процесса создаются по мере необходимости, так что каталог таблиц страниц большинства процессов ссылается лишь на небольшой набор таблиц страниц.

2.1.2 Форматы дескрипторов



а)



б)

Рис.6 а, б

PTE:

биты	назначение
11- U	Зарезервирован. В многопроцессорных системах указывает, можно ли записывать на эту страницу
10 - P	Зарезервирован.
9 - Cw	Зарезервирован. В многопроцессорных системах Copy-on-write – копирование при записи
8 - GI	Global - глобальная. Трансляция относится ко всем процессам (например, сброс буфера трансляции не повлияет на этот PTE)
7 - L	Зарезервирован. Большая страница в случае PAE.
6 - D	Грязная (Dirty) - модифицированная страница
5 - A	Accessed – бит обращения. Устанавливается при обращении к странице
4 - Cd	Cache disabled – кэширование страницы отключено
3 - Wt	Write through – отключает кэширование записи на данную страницу, в результате чего все измененные данные сразу же сбрасываются на диск
2 - 0	User/Superuser или Owner – указывает доступна ли страница из кода пользовательского режима
1 - W	Write – в однопроцессорных системах указывает тип доступа (w/r, only r), а в многопроцессорных системах указывает возможность записи в страницу
0 – P или V	Present или Valid – присутствует или действительный – указывает, соответствует ли PTE странице в физической памяти

3. Структуры пользовательского адресного пространства на платформе x86

По умолчанию каждый пользовательский процесс в 32-разрядной версии Windows располагает собственным адресным пространством, размер которого может варьироваться от 2х до 3х – гигабайт, как показано на рис.7.

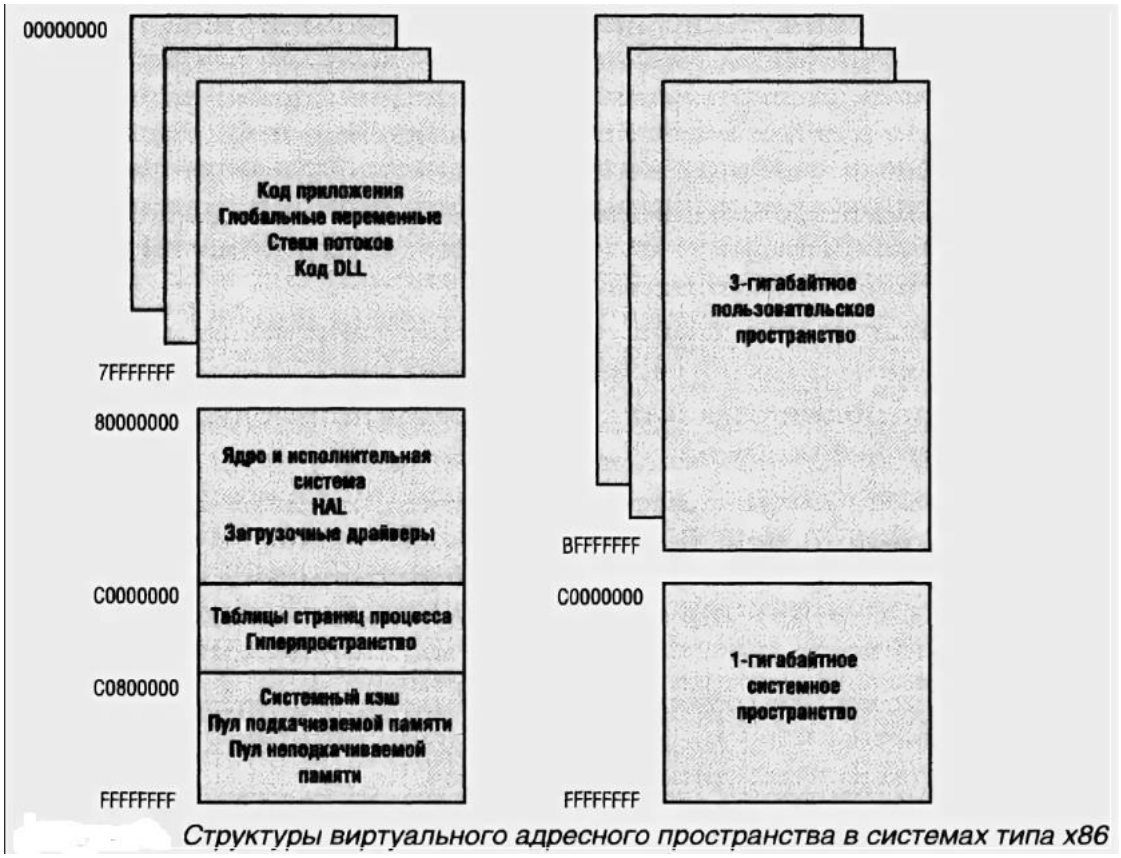


Рис.7

Это возможность поддерживается с помощью загрузочного ключа в Boot.ini. Windows XP Windows и Server 2003 поддерживают дополнительный ключ /UVERSA. Адресное пространство пользователя процесса является защищенным. Системное адресное пространство процесса это – отображение (mapping).

4. Кэширование

Большинство людей путают термины буферизация и кеширование. Хотя оба хранят данные временно, но между ними имеются отличия. Буферизация в основном используется для согласования скорости передачи между отправителем и получателем и хранит исходные данные. Cache ускоряет скорость доступа к многократно используемым данным. Другими словами, кэш всегда хранит копию каких-то данных.

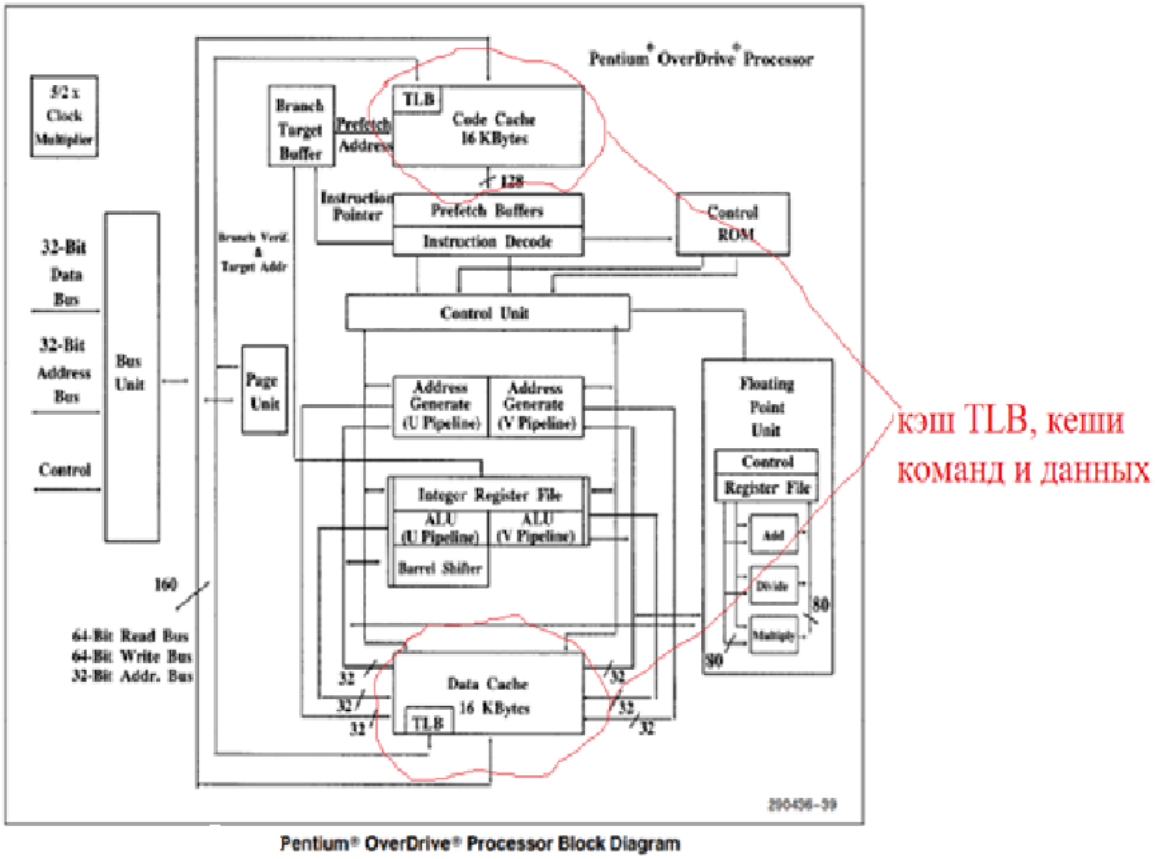


Рис.7

На рис. 7 показаны два конвейера процессора Pentium OverDrive и модуль с плавающей запятой, которые способен к автономной работе. Каждый конвейер выдает часто используемые инструкции за один такт. Вместе двойные каналы могут выдавать две целочисленные инструкции за один такт или одну инструкцию с плавающей запятой (при определенных обстоятельствах 2 команды с плавающей запятой инструкция) в одни часы.

Как видно из рисунка в процессоре каждый канал имеет кэш TLB - буфер ассоциативной трансляции (англ. **Translation lookaside buffer**, **TLB**).

Таблицы страниц хранятся в оперативной памяти. Если при каждом обращении по виртуальному адресу выполнять полностью трансляцию адресов, это будет работать очень медленно. Поэтому в процессоре реализуется специальный кеш под названием «буфер ассоциативной трансляции» (Translation lookaside buffer, TLB).

На практике вероятность промаха TLB невысока и составляет в среднем от 0,01% до 1%.

На рис.8 показана логическая структура кэша TLB для 486 процессора. Он определяется как четырех направленный ассоциативный по множеству буфер.

В кэше три блока:

- Блок достоверности + LRU (Least Reasently Used)
- Блок тегов
- Блок данных.

Выбор одного из 8 множеств осуществляется 3-мя разрядами виртуального адреса – 12, 13, 14. Первые 12 разрядов – смещение – в адресации не участвуют. Выбор подмножества в множестве осуществляется с помощью остальных старших 17 разрядов. В результате выбирается хранящийся в кэше физический адрес.

Замещение в кэше осуществляется на основе бита достоверности и трех битов, с помощью которых реализуется алгоритм псевдо LRU.

Рис. 8

Блок достоверности-LRU работает следующим образом:

Каждая строка имеет бит достоверности. При очистке кэша или сбросе процессора все биты достоверности сбрасываются в 0. Когда производится заполнение строки кэша, просто ищется любая недостоверная строка. Если недостоверных строк нет, то замещаемая строка ищется по алгоритму псевдо-LRU. Обозначим строки в множестве через L0, L1, L2, L3. Каждому

множеству в блоке LRU соответствует 3 бита: b0, b1, b2, которые модифицируются при каждом попадании и заполнении следующим образом:

- если последнее обращение в множестве было к L0 или L1, то бит b0 устанавливается в 1, а если обращение было к L2 или L3, то b0 сбрасывается в 0;

- если последнее обращение в паре L0 – L1 было к L0, то бит b1устанавливается в 1, а если обращение было L1, то b1 – в 0;

- если последнее обращение в паре L2 – L3 было к L2, то бит b2 устанавливается в 1, а если обращение было L3, то b2 – в 0.

Другими словами, выбор заменяемой строки определяется значением трех битов:

b0	b1	b2	заменяется
0	0	X	L0
0	1	X	L1
1	X	0	L2
1	X	1	L3

В многоядерных процессорах количество и размеры кэшей больше. На рис.9 показано одно ядро.

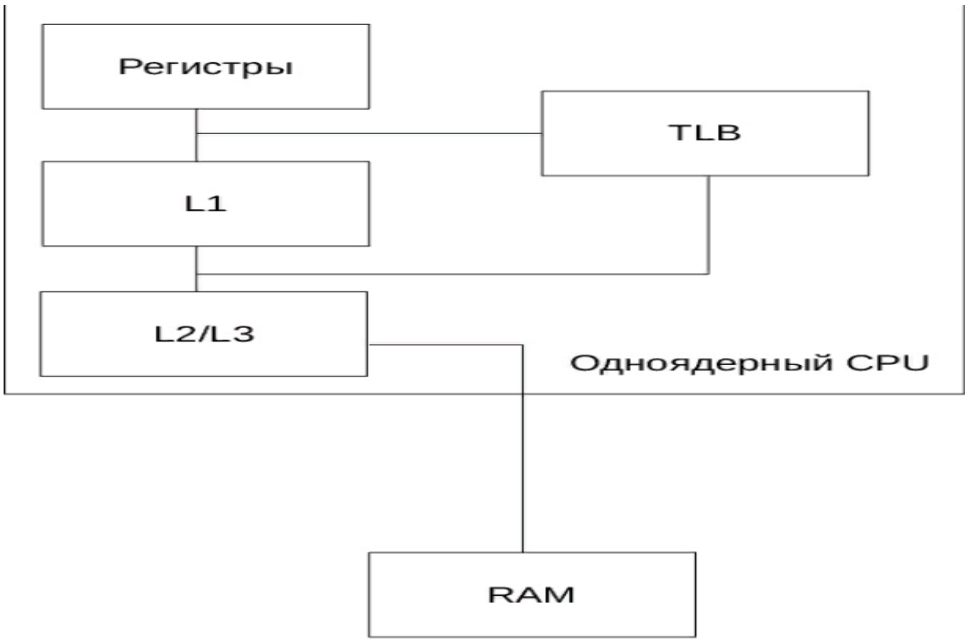


Рис.9

Если представить память компьютера в виде иерархии по её скорости, кэш будет на вершине этой иерархии. К тому же он ближе всего к вычислительным ядрам, так как является частью процессора. Кэш память процессора представляет из себя статическую память (SRAM) и предназначен для ускорения работы с ОЗУ. В отличие от динамической оперативной памяти (DRAM), здесь можно хранить данные без постоянного обновления.

Веся кэш память процессора разделена на три уровни: L1, L2 и L3. Эта иерархия тоже основана на скорости работы кэша, а также на его объеме (рис.10).

- **L1 Cache (кэш первого уровня)** — это максимально быстрый тип кэша в процессоре. С точки зрения приоритета доступа, этот кэш содержит те данные, которые могут понадобиться программе для выполнения определенной инструкции;
- **L2 Cache (кэш второго уровня процессора)** — медленнее, по сравнению L1, но больше по размеру. Его объем может быть от 256 килобайт до восьми мегабайт. Кэш L2 содержит данные, которые, возможно, понадобятся процессору в будущем. В большинстве современных процессоров кэш L1 и L2 присутствуют на самих ядрах процессора, причём каждое ядро получает свой собственный кэш;
- **L3 Cache (кэш третьего уровня)** — это самый большой и самый медленный кэш. Его размер может быть в районе от 4 до 50 мегабайт. В современных CPU на кристалле выделяется отдельное место под кэш L3.

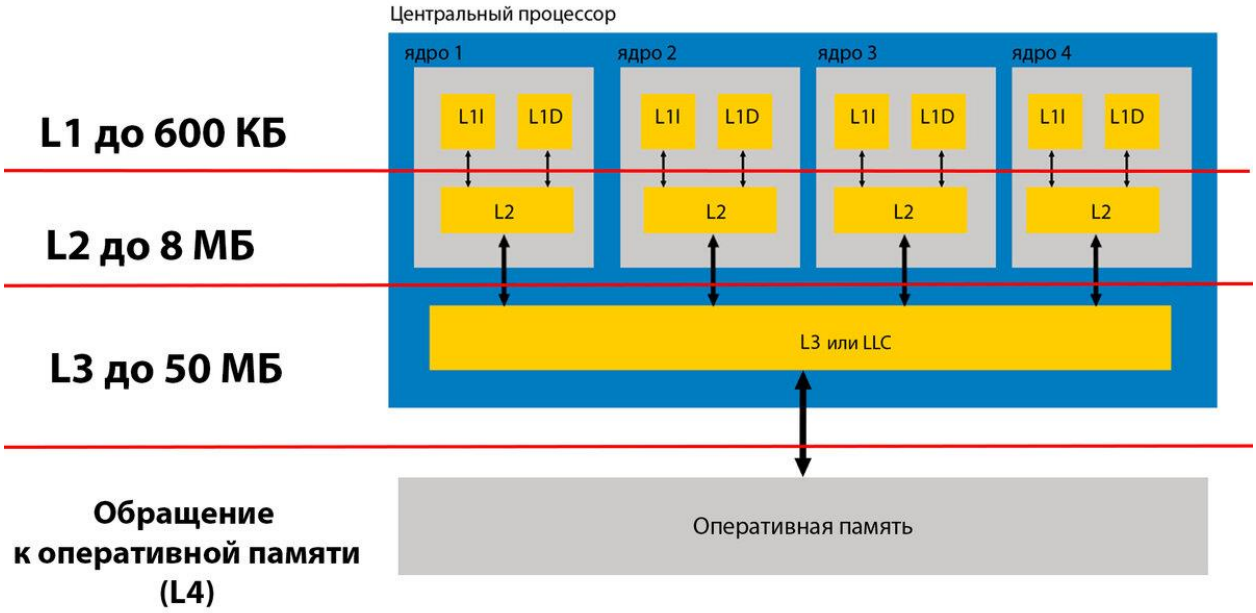


Рис.10

5. *Расширения архитектуры x86*

PAE (Physical Address Extension)

В более поздних 32-разрядных процессорах (начиная с Pentium Pro) появилось PAE (Physical Address Extension) — расширение адресов физической памяти до 36 бит (возможность адресации 64 Гбайт ОЗУ). Это изменение не затронуло разрядности задач — они остались 32-битными.

Впервые расширение появилось в процессоре Pentium Pro. Для использования 36-разрядной адресации памяти необходима поддержка расширения физических адресов на программном уровне (включение режима PAE в ОС) и аппаратном: необходима поддержка как со стороны процессора, так и материнской платы (можно определить по команде CPUID). Материнские платы с поддержкой PAE предназначались для серверов.

В архитектуре x86-64 реализовано PAE.

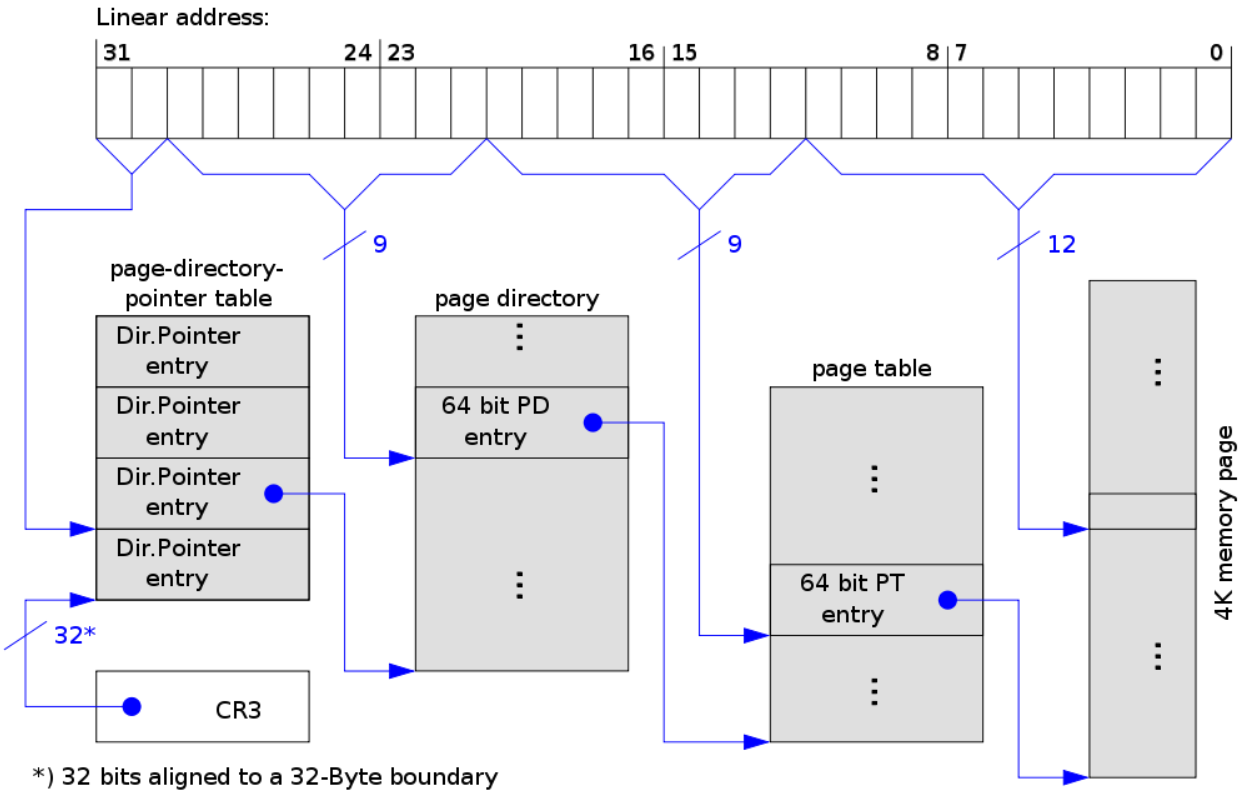


Рис.11

Виртуальный адрес делится на четыре части: четвертое поле в 32-разрядных процессорах занимает только 2 бита и может адресовать таблицу только из четырех элементов. Смещение осталось равным 12 бит, но для выделения 2х бит размер второго и третьего полей стало равным 9 бит.

Таблица Page-Directory-Point содержит 4 дескриптора таблиц Page-Directory, т.е. на один процесс может быть создано до 4х таблиц Page-Directory. Каждая таблица может содержать 512 элементов. Размер дескрипторов увеличивается до 8 байт, что является ключевым моментом.

Четыре таблицы Page-Directory могут адресовать 2048 таблиц страниц с дескрипторами по 8 байт. При делении адресного пространства процесса на две равные части: 1024 таблицы – защищенное адресное пространство и 1024 таблицы – отображение системы.

CR3	Page-Directory Base																			-							PCD	PWT	-		
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CR3 (Control Register 3): 32-битный регистр, содержащий физический адрес корневого каталога страниц памяти (биты 31-11, если флаг PAE в регистре CR4 сброшен; биты 31-5, если флаг PAE в регистре CR4 установлен), а также биты по управлению кэшированием страниц памяти:

бит 4: Флаг запрещения кэширования страниц PCD (486+). Если бит установлен, то текущая страница не будет загружена в кэш.

бит 3: Флаг сквозного кэширования страниц PWT (486+). Этот флаг управляет методом записи страниц во внешний кэш.

CR4	-																					FPE	FSR	PCE	PGE	MCE	PAE	PSE	DE	TSD	PVI	VME
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CR4 (Control Register 4): 32-битный регистр, управляющий использованием архитектурных расширений процессоров Pentium и более новых. Наличие этих расширений необходимо проверять при помощи команды CPUID:

Рис.12

Регистр CR4 впервые был введен в Pentium и он обеспечивает переключение различных режимов и дополнительных возможностей.

Пятый бит регистра CR4 – PAE – расширение физического адреса:

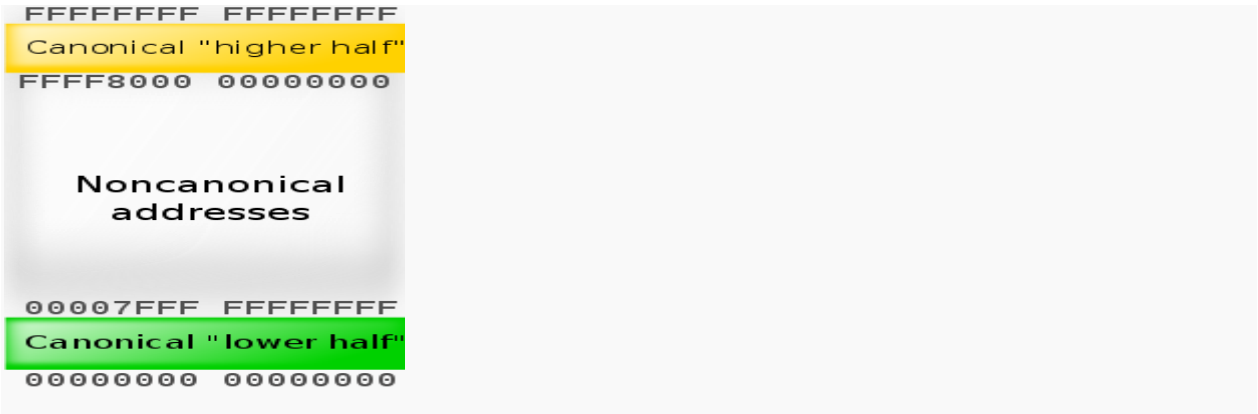
PAE = 1 разрешает использование расширенной 36-разрядной (вместо стандартной 32-разрядной) физической адресации.

Четвертый бит регистра CR4 – PSE – расширение размера страницы:

PSE = 1 разрешает использование страниц расширенного размера – 4Мб или 2Мб.

В дескрипторе странице 12 младших разрядов это – флаги аналогичные рассмотренным. Старшие 24 разряда – физический адрес страницы.

6. *Виртуальное адресное пространство в x86-64*



Хотя виртуальные адреса имеют разрядность в 64 бита, текущие реализации (и все чипы, которые находятся на стадии проектирования) не позволяют использовать

всё виртуальное адресное пространство из 2^{64} байт (16 экзабайт). Это будет примерно в четыре миллиарда раз больше виртуального адресного пространства на 32-битных машинах. В обозримом будущем большинству операционных систем и приложений не потребуется такое большое адресное пространство, поэтому внедрение таких широких виртуальных адресов просто увеличит сложность и расходы на трансляцию адреса без реальной выгоды. Поэтому AMD решила, что в первых реализациях архитектуры фактически при трансляции адресов будут использоваться только младшие 48 бит виртуального адреса.

Кроме того, спецификация AMD требует, что старшие 16 бит любого виртуального адреса, биты с 48-го по 63-й, должны быть копиями бита 47 (по принципу sign extension). Если это требование не выполняется, процессор будет вызывать исключение. Адреса, соответствующие этому правилу, называются «канонической формой». Канонические адреса в общей сложности составляют 256 терабайт полезного виртуального адресного пространства. Это по-прежнему в 65536 раз больше, чем 4 ГБ виртуального адресного пространства 32-битных машин.

Это соглашение допускает при необходимости масштабируемость до истинной 64-разрядной адресации. Многие операционные системы (включая семейство Windows NT и GNU/Linux) берут себе старшую половину адресного пространства (пространство ядра) и оставляют младшую половину (пользовательское пространство) для кода приложения, стека пользовательского режима, кучи и других областей данных. Конструкция «канонического адреса» гарантирует, что каждая совместимая с AMD64 реализация имеет, по сути, две половины памяти: нижняя половина «растет вверх» по мере того, как становится доступнее больше виртуальных битов адреса, а верхняя половина — наоборот, вверху адресного пространства и растет вниз.

Первые версии Windows для x64 даже не использовали все 256 ТБ; они были ограничены только 8 ТБ пользовательского пространства и 8 ТБ пространства ядра. Всё 48-битное адресное пространство стало поддерживаться в Windows 8.1, которая была выпущена в октябре 2013 года.

6.1 Структура таблицы страниц

Ставится задача транслировать 48-битный виртуальный адрес в физический. Она решается аппаратным обеспечением — блоком управления памятью (memory management unit, MMU). Этот блок является частью процессора. Чтобы транслировать адреса, он использует структуры данных в оперативной памяти, называемые таблицами страниц.

Вместо двухуровневой системы таблиц страниц, используемой системами с 32-битной архитектурой x86, системы, работающие в длинном режиме, используют четыре уровня таблицы страниц.

Возможные размеры страниц:

- 4 КБ (2^{12} байт) — наиболее часто используется (как и в x86)
- 2 МБ (2^{21} байт)

- 1 ГБ (2^{30} байт)

Пусть для определённости размер страницы равен 4 КБ. Значит, младшие 12 битов адреса кодируют смещение внутри страницы и не изменяются, а старшие биты используются для определения адреса начала страницы.

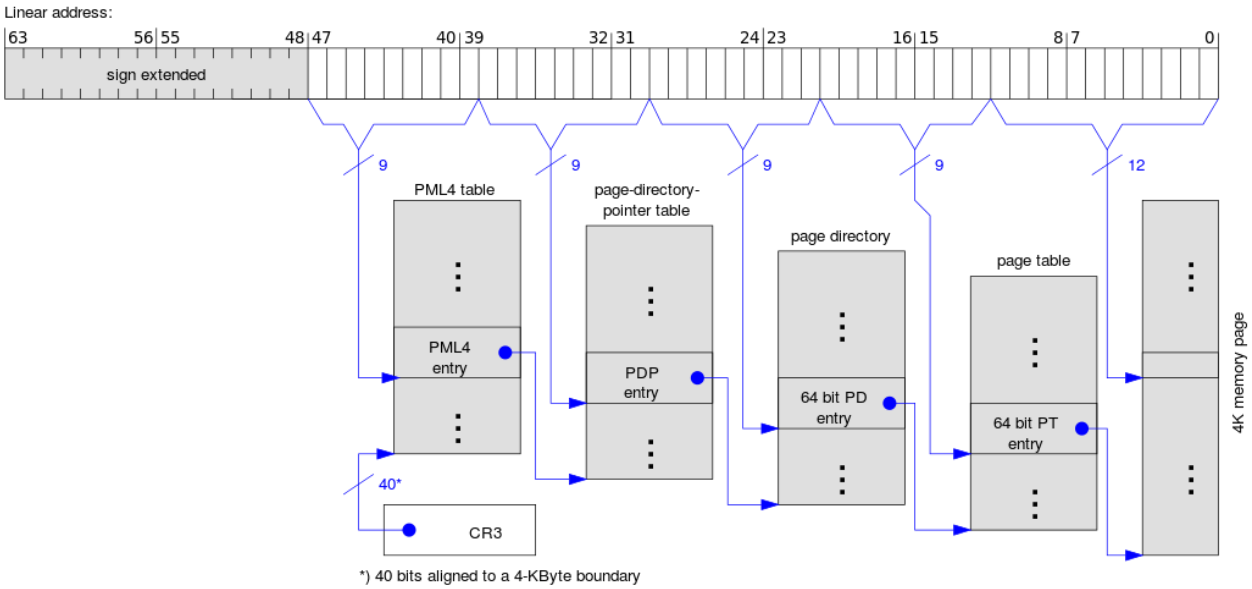


Рис. 12

Формат дескриптора страницы:

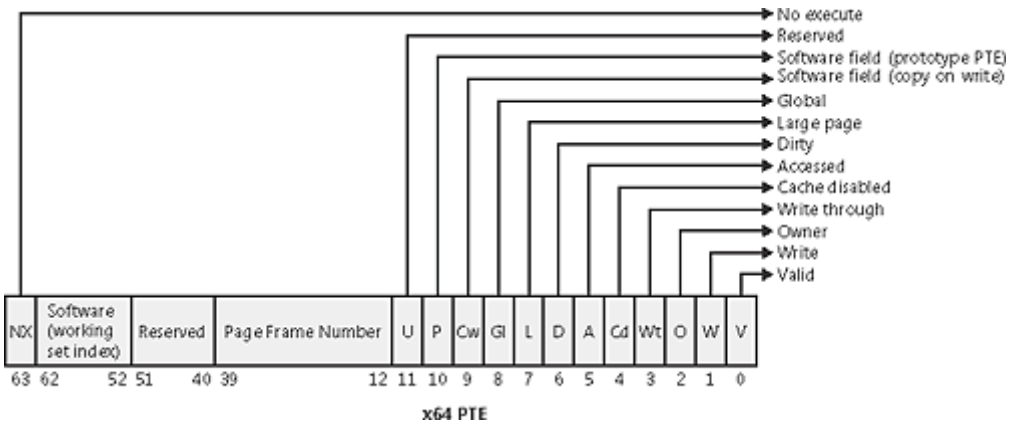
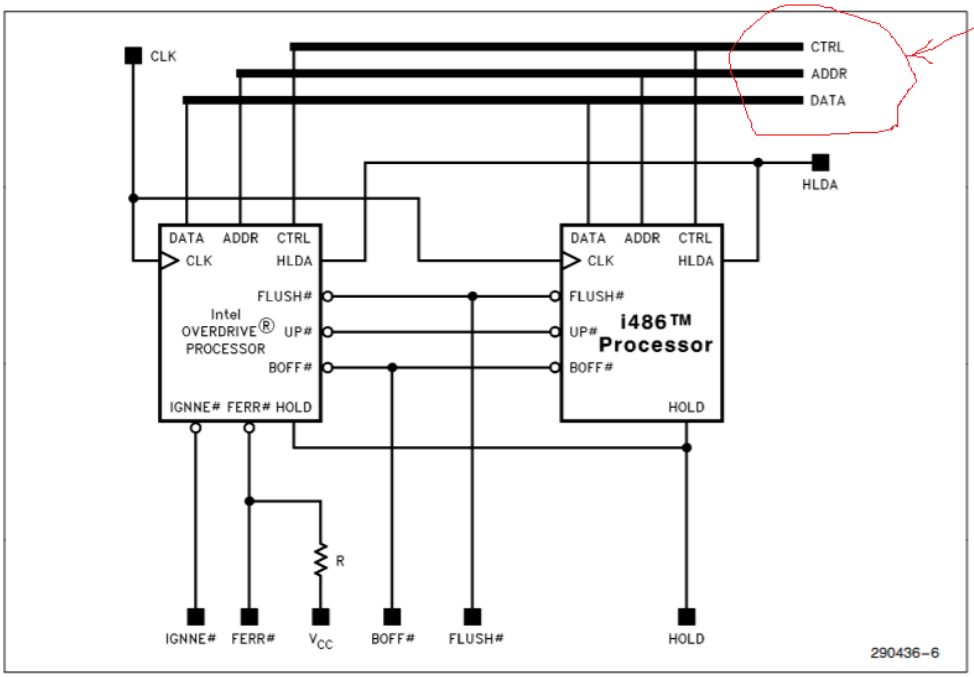


Рис. 13

Полная иерархия сопоставления страниц размером 4 КБ для всего 48-битного пространства займет немногим больше 512 ГБ ОЗУ (около 0.195% от виртуального пространства 256 ТБ).

Приложение

486 процессор



три шины

Figure 9-1. Intel OverDrive® Socket Circuit Diagram for Systems Based on Intel486™ Processors That Have the UP# Input Pin