Лекция

Режимы работы современных процессоров

Программная модель современных процессоров x86
Современные представители семейства x86 являются 32-битными процессорами; в новых моделях появилось 64-битное расширение. История 32-битных процессоров Intel (архитектуры IA-32) началась с процессора 80386. Он вобрал в себя все черты своих 16-битных предшественников 8086/88 и 80286 для обеспечения совместимости с громадным объемом ПО, существовавшего на момент его появления.

Разрядность адреса определяет, сколько битов (16, 32 или 64) используется в регистрах, формирующих адрес данных или инструкций, расположенных в памяти. *Разрядность данных* определяет, сколько битов используется в инструкциях, оперирующих словами. Каждому режиму работы процессоров соответствуют своя разрядность, применяемая по умолчанию. При необходимости для каждой исполняемой инструкции разрядность адреса или/и операнда может изменяться с помощью специальных *префиксов* (байтов перед кодом инструкции).
32-битные регистры процессоров позволяют непосредственно адресовать до 4 Гбайт памяти. Встроенный блок управления памятью поддерживает механизмы *сегментации* и *страничной трансляции адресов.*

Расширения x86-64 и EM64T в первую очередь предназначены для радикального увеличения объема адресуемой памяти: 64-битные регистры позволяют адресовать до 2^{64} =18,4 x 10^{18} байт. Это число и является пределом объема виртуальной памяти 64-битного процессора, но пока используют только младшие 48 битов адреса.
Процессоры предоставляют четырехуровневую *систему привилегий* для защиты памяти, ввода-вывода и прерываний, а также механизм *переключения задач* для многозадачных ОС.

Процессоры могут работать в различных *режимах,* определяющих возможности адресации памяти и защиты.

Режим работы процессора задается операционной системой с учетом режима работы приложений (задач). У процессоров с 64-битным расширением появляются новые режимы, среди которых есть и режимы, обеспечивающие совместимость с 32-разрядными операционными системами и приложениями. Новые режимы используются только в 64-битных ОС, а полностью их преимущества доступны только 64-битным приложениям.

Режимы работы процессоров

32-битные процессоры могут работать в одном из следующих режимов:

♦ *Режим реальной адресации* (real address mode), или просто *реальный режим* (real mode), полностью совместим с 8086. В этом режиме возможна адресация до 1 Мбайт физической памяти (на самом деле, как и у 80286, почти на 64 Кбайт больше).

♦ *Защищенный режим виртуальной адресации* (protected virtual address mode), или просто *защищенный режим* (protected mode). В этом режиме у процессора включаются механизмы сегментации и страничной трансляции. Механизм сегментации позволяет поддерживать виртуальную память объемом до 64 Тбайт. На практике используется только страничная трансляция, благодаря которой каждой задаче предоставляется до 4 Гбайт виртуального адресного пространства. По умолчанию и адреса, и операнды имеют разрядность 32 бита. В защищенном режиме процессор может выполнять дополнительные инструкции, недоступные в реальном режиме; ряд инструкций, связанных с передачей управления, обработкой прерываний, и некоторые другие выполняются иначе, чем в реальном режиме.

Есть возможность организации 16-разрядного защищенного режима в стиле процессора 80286, но этот режим не представляет интереса.

♦ *Режим виртуального процессора 8086* (Virtual 8086 Mode, V86) является особым состоянием задачи защищенного режима, в котором процессор функционирует как 8086 (16-битные адрес и данные). На одном процессоре в таком режиме могут параллельно исполняться несколько задач с изолированными друг от друга ресурсами. При этом использование физического адресного пространства памяти управляется механизмами сегментации и трансляции страниц. Попытки выполнения недопустимых команд, выхода за рамки отведенного пространства памяти и разрешенной области ввода-вывода контролируются системой защиты. Более эффективен *расширенный режим вир*туального процессора 8086 (Enhanced Virtual 8086 Mode, EV86), в котором оптимизирована виртуализация прерываний.

♦ *«Нереальный» режим* (unreal mode, он же *big real mode)* — это «неофициальный» режим, который поддерживают все 32-битные процессоры. Он позволяет адресоваться к 4-гигабайтному пространству памяти. В этом режиме инструкции исполняются так же, как и в реальном режиме, но с помощью дополнительных сегментных регистров FS и GS программы получают непосредственный доступ к данным во всей физической памяти.

♦ В *режиме системного управления* (System Management Mode, SMM) процессор выходит в иное, изолированное от остальных режимов пространство памяти. Этот режим используется в служебных и отладочных целях. С его помощью, например, скрытно выполняются функции управления энергопотреблением, эмулируются обращения к несуществующим аппаратным средствам (эмуляция клавиатуры и мыши PS/2 для USB).

Для процессоров x86-64 вышеперечисленные режимы объединены понятием *legacy mode;* кроме того, появился новый режим *long mode* с двумя подрежимами:

♦ *64-битный режим* (64-bit mode) — это режим полной поддержки 64-битной виртуальной адресации и 64-битных расширений регистров. В этом режиме используется только плоская модель памяти (общий сегмент для кода, данных и стека). По умолчанию разрядность адреса составляет 64 бита, а операндов (для большинства инструкций) — 32 бита, однако префиксом (REX) можно заказать 64-битные операнды. Имеется новый способ адресации данных — относительно указателя инструкций. Режим предназначен для использования 64-битными ОС при запуске 64-битных приложений — он включается операционной системой для сегмента кода конкретной задачи;

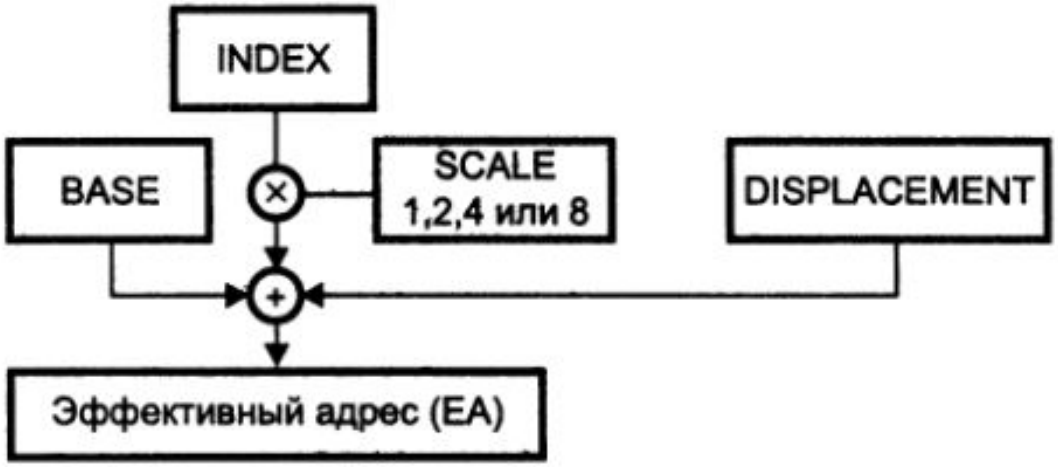
♦ *режим совместимости* (compatibility mode) позволяет 64-битным ОС работать с 32- и 16-битными приложениями. Для приложений процессор выглядит как обычный 32-битный со всеми атрибутами защищенного режима, сегментацией и страничной трансляцией. 64-битные свойства используются только операционной системой, что отражается в процедурах трансляции адресов, обработки исключений и прерываний. Режим включается операционной системой для сегмента кода конкретной задачи.

32-битные ОС используют процессоры x86-64 только в режиме *legacy mode* (как обычный процессор IA-32).

Пространство памяти (memory space) предназначено для хранения кодов инструкций и данных. Память может логически организовываться в виде одного или множества сегментов произвольной длины (в реальном режиме — фиксированной). Помимо сегментации в защищенном режиме возможно (при страничной трансляции адресов) разбиение логической памяти на страницы размером 4 Кбайт, каждая из которых может отображаться на любую область физической памяти. Начиная с 5-го поколения появилась возможность увеличения размера страницы до 4 Мбайт. Сегментация и страничная трансляция адресов могут применяться совместно и по отдельности. Сегментация является средством организации логической памяти на прикладном уровне. Страничная трансляция адресов применяется на системном уровне для управления физической памятью. Сегменты и страницы могут выгружаться из физической оперативной памяти на диск и по мере необходимости подкачиваться с него обратно в физическую память. Таким образом реализуется виртуальная память.

Эффективный адрес

При обращении к памяти (к данным), как и при формировании адреса перехода, процессор строит *эффективный адрес,* который может включать до трех компонентов (рис. 1). Такой сложный способ задуман для облегчения доступа к элементу массива: компонент *BASE* — базовый адрес массива, *INDEX* — номер элемента, *DISPLACEMENT* — смещение внутри элемента. Массив может состоять из байтов, слов, двойных и учетверенных слов — это учитывается масштабным коэффициентом *SCALE* (1, 2, 4 или 8). Компоненты эффективного адреса могут быть константами (в инструкции), находиться в регистрах и даже в памяти. Такая универсальность оборачивается значительными микроархитектурными издержками.



Преобразование адресов

Применительно к памяти различают три адресных пространства: логическое, линейное и физическое. По сочетанию сегментации и страничной трансляции различают две модели памяти:

♦ В сегментной модели памяти приложение использует несколько сегментов памяти (для кода, данных, стека) и может переключать используемые сегменты. В этой модели приложение оперирует логическими адресами.

♦ В плоской модели памяти приложению для всех целей выделяется единственный сегмент. В этой модели приложение оперирует линейными адресами. Плоская модель гораздо проще и удобнее в обращении и используется в современных ОС.

Логический адрес состоит из селектора сегмента Seg и эффективного адреса, называемого также смещением (offset). Логический адрес обозначается в форме Seg:Offset. *Селектор сегмента* хранится в старших 14 битах сегментного регистра (CS, DS, ES, SS, FS или GS), участвующего в адресации конкретного элемента памяти. По значению селектора из специальных *таблиц дескрипторов сегментов,* хранящихся в памяти, извлекается начальный адрес сегмента. Поскольку каждая задача может иметь до 16К селекторов (2^{14}), а смещение, ограниченное размером сегмента, - достигать 4 Гбайт, логическое адресное пространство для каждой задачи может равняться 64 Тбайт. Операционная система может ограничить число доступных сегментов и их конкретные размеры.

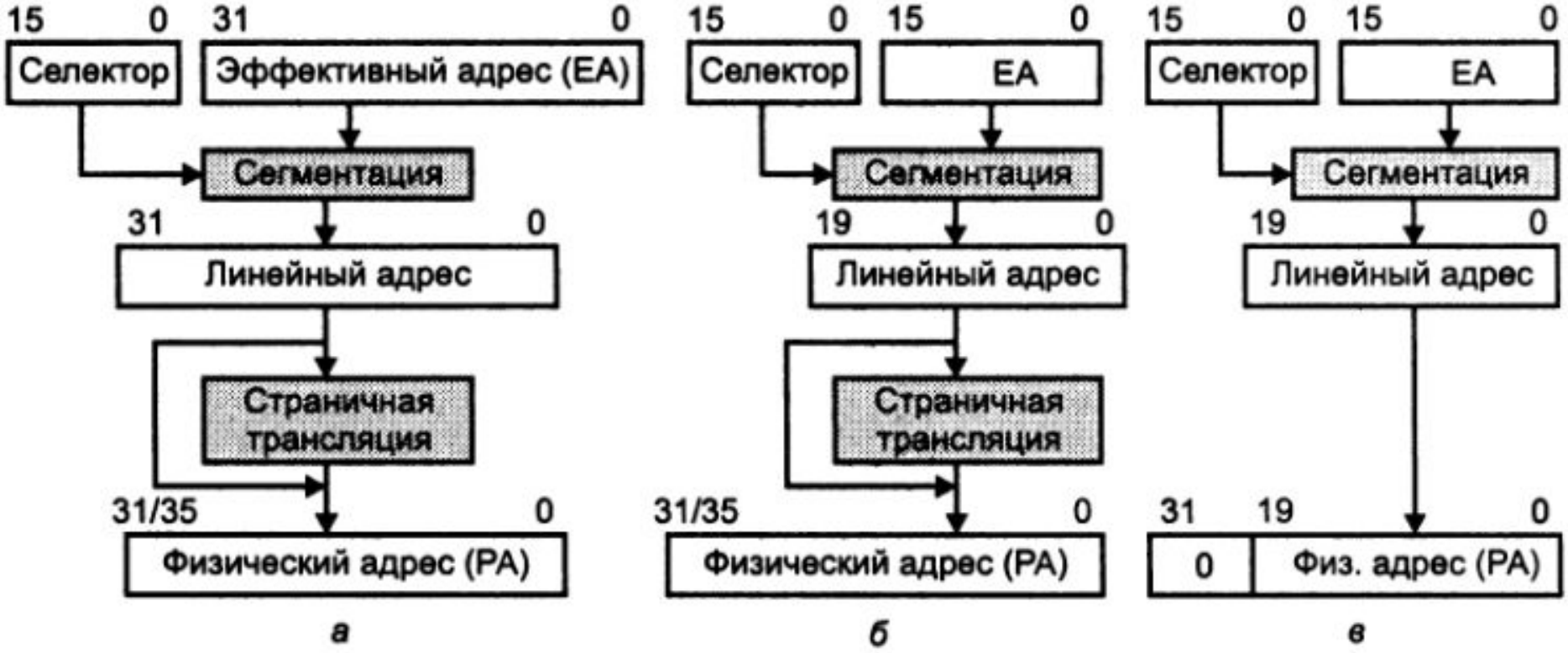
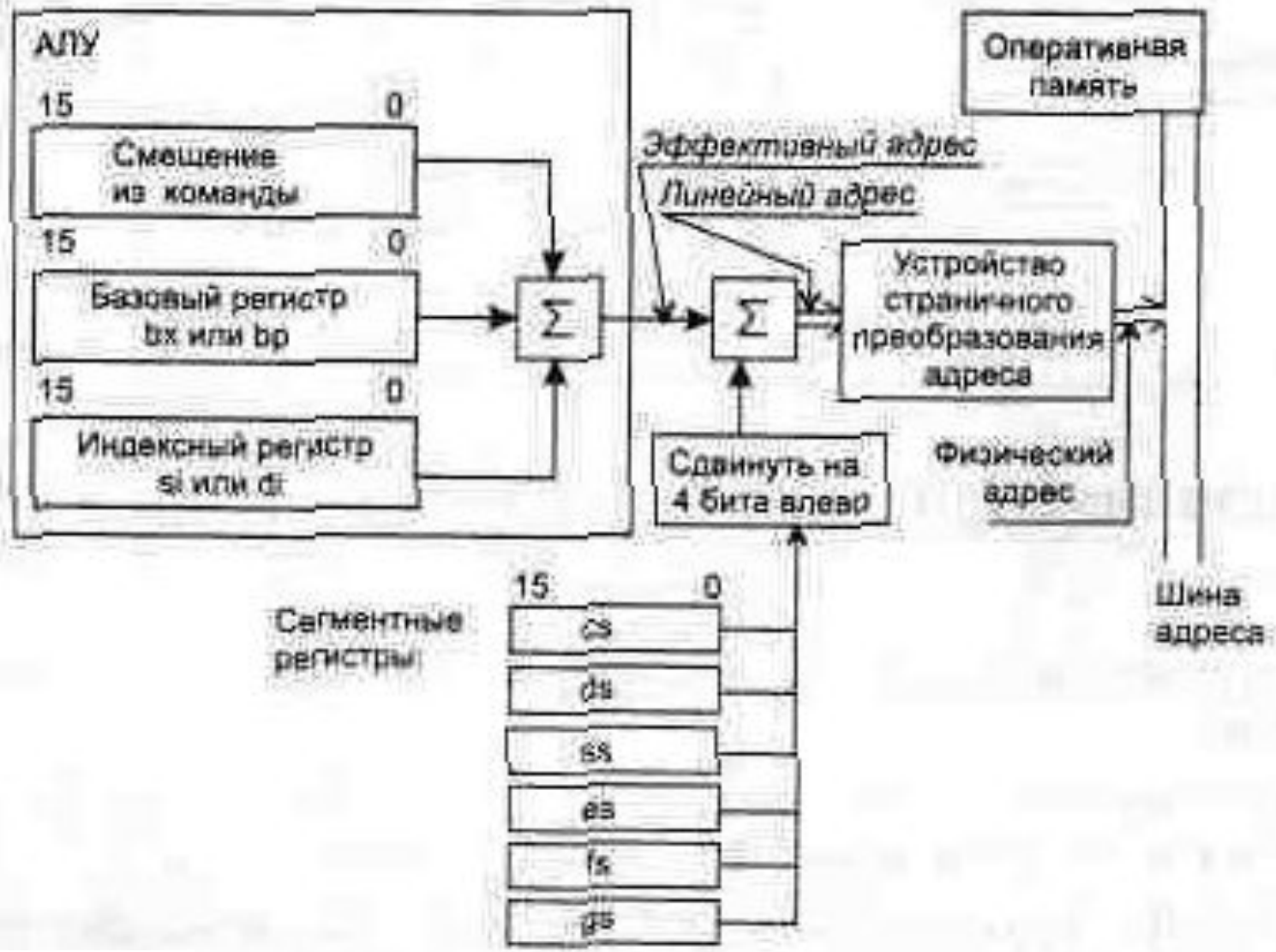


Рис. 2. Формирование адреса памяти в 32-битных процессорах: *а* — в защищенном режиме, *б* — в режиме V86, *в* — в реальном режиме

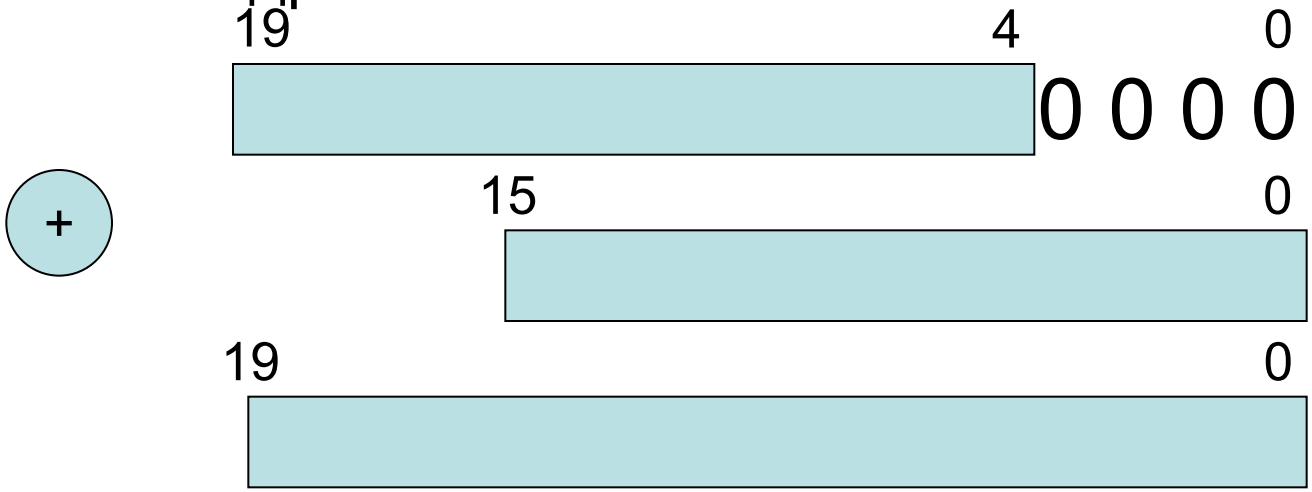
Преобразование логического адреса в физический для 32-битных процессоров иллюстрирует рис. 2. Блок сегментации транслирует логическое адресное пространство в 32-битное пространство линейных адресов. *Линейный адрес* образуется сложением базового адреса сегмента с эффективным адресом.

В реальном режиме селектор любого сегмента равен адресу его начала, деленому на 16. Чтобы получить адрес в памяти, 16-битное смещение складывают с этим селектором, сдвинутым предварительно влево на 4 разряда. Таким образом, оказывается, что максимальный доступный адрес в реальном режиме 2^{20}-1 = 1 048 575.


АЛУ
15
0
Смещение из команды
15
0
Базовый регистр bx или bp
15
0
Индексный регистр si или di
Σ
Σ
Эффективный адрес
Линейный адрес
Оперативная память
Устройство страничного преобразования адреса
Физический адрес
Сдвинуть на 4 бита влево
Шина адреса
Сегментные регистры
15
0
cs
ds
ss
es
fs
gs

Формирование линейного адреса

Содержимое сегментного регистра сдвигается влево на 4 бита и складывается со смещением, в результате чего получается двадцатиразрядный физический адрес.



В реальном режиме микропроцессор работает как 8086 с возможностью использования 32-битных расширений.

В отличие от 8086 микропроцессоры 286+ в определенных ситуациях генерируют исключения, например, при превышении предела сегмента, который для всех сегментов в реальном режиме - 0FFFFh.

Имеется две фиксированные области в памяти, которые резервируются в режиме реальной адресации:

область инициализации системы
область таблицы прерываний

Ячейки от 00000h до 003FFH резервируются для векторов прерываний. Каждое из 256 возможных прерываний имеет зарезервированный 4-байтовый адрес перехода.

Ячейки от FFFFFFF0H до FFFFFFFFH резервируются для инициализации системы.

Данный механизм образования физического адреса позволяет сделать программное обеспечение перемещаемым, то есть не зависящим от конкретных адресов загрузки его в оперативной памяти.

Недостатки такой организации памяти:

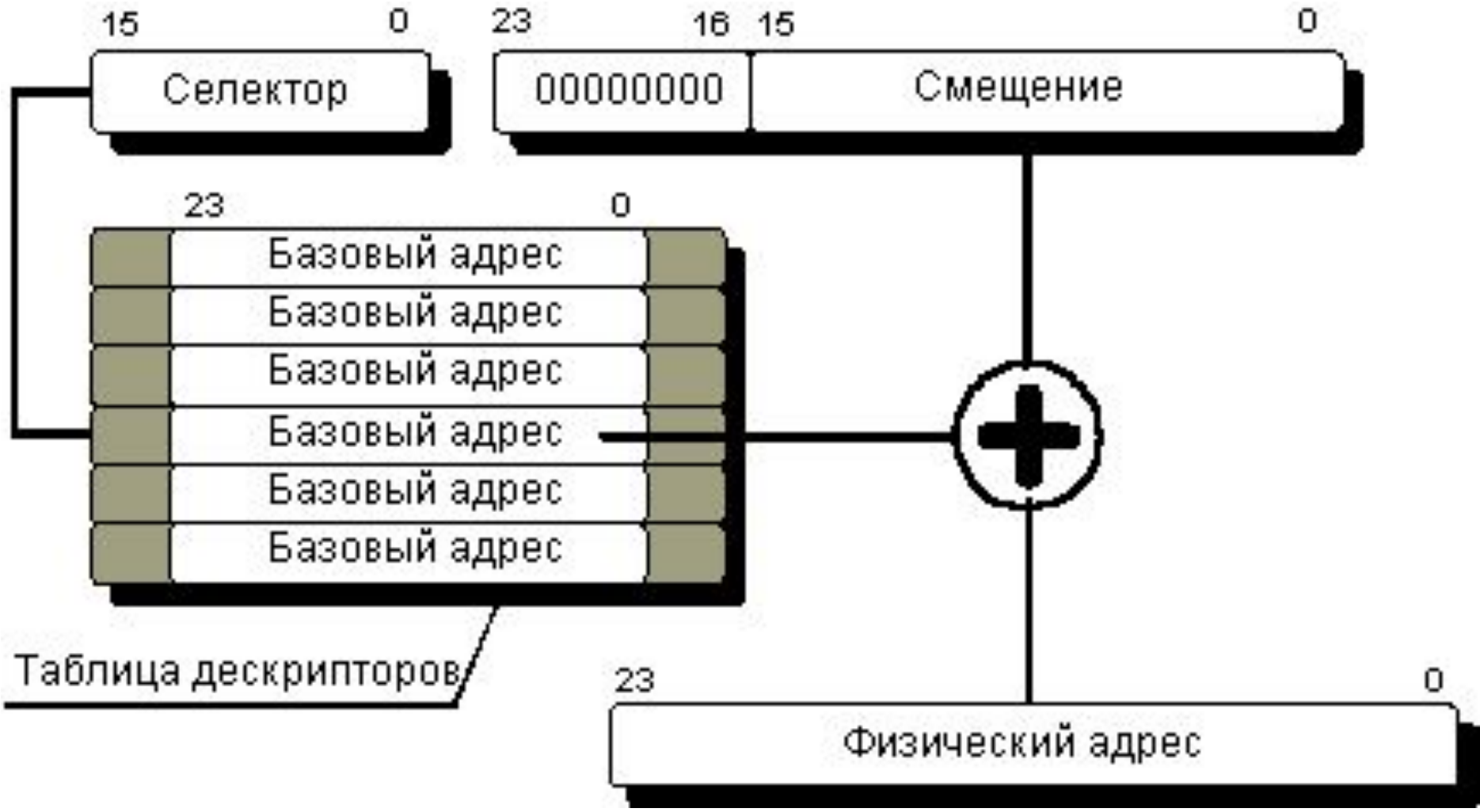
сегменты бесконтрольно размещаются с любого адреса, кратного 16 (так как содержимое сегментного регистра аппаратно смещается на 4 разряда). Как следствие, программа может обращаться по любым адресам, в том числе и реально не существующим;
сегменты имеют максимальный размер 64 Кбайт;
сегменты могут перекрываться с другими сегментами.

Защищенный режим лишен недостатков реального режима, в нем можно адресоваться к участку памяти размером 4 Гб как к одному непрерывному массиву и вообще забыть о сегментах и смещениях. Этот режим намного сложнее реального, поэтому, чтобы переключить в него процессор и поддерживать работу в этом режиме, надо написать небольшую операционную систему. Кроме того, если процессор уже находится под управлением какой-то операционной системы, которая перевела его в защищенный режим, например Windows , она, скорее всего, не разрешит программе устранить себя от управления компьютером. С этой целью были разработаны специальные интерфейсы, позволяющие программам, запущенным в режиме V86 в DOS, переключаться в защищенный режим простым вызовом соответствующего прерывания — VCPI и DPMI.

В защищенном режиме базовый адрес загружается из дескриптора, хранящегося в таблице, по селектору, загруженному в используемый сегментный регистр.

РАБОТА С АДРЕСАМИ

В защищенном режиме, как и в реальном, логический адрес состоит из двух компонент. Однако эти компоненты называются не сегмент и смещение, а селектор и смещение . Для вычисления физического адреса в процессоре 80286 используются также две таблицы дескрипторов - глобальная таблица дескрипторов GDT (Global Descriptor Table) и локальная таблица дескрипторов LDT (Local Descroptor Table). Селектор используется для адресации ячейки одной из таблиц дескрипторов, содержащей помимо прочей информации базовый 24-разрядный адрес сегментов. Для получения физического адреса базовый адрес складывается со смещением, расширенным до 24 разрядов.



Получение физического адреса в процессоре 80286

Согласно этой схеме адресации памяти, селектор содержит номер ячейки таблицы дескрипторов, но не компоненту физического адреса. Программа может задавать не любые значения селекторов, а только те, которые соответствуют существующим ячейкам таблицы дескрипторов. Разумеется, программа может загрузить в сегментный регистр любое значение, однако при попытке обратиться к сегменту памяти с использованием неправильного селектора работа программы будет прервана.
Таким образом, несмотря на то, что компоненты адреса остались, как и в реальном режиме, 16-разрядными, новая схема адресации защищенного режима процессора 80286 позволяет адресовать до 16 Мбайт памяти, так как в результате преобразования получается 24-разрядный физический адрес.
Кроме индекса, используемого для выбора ячейки дескрипторной таблицы при формировании физического адреса, селектор содержит еще два поля
Поле TI (Table Indicator)
бит 2: индикатор таблицы 0/1 — использовать GDT/LDT

Поле RPL
биты 1 – 0: уровень привилегий запроса (RPL)— это число от 0 до 3, указывающее уровень защиты сегмента, для доступа к которому используется данный селектор.

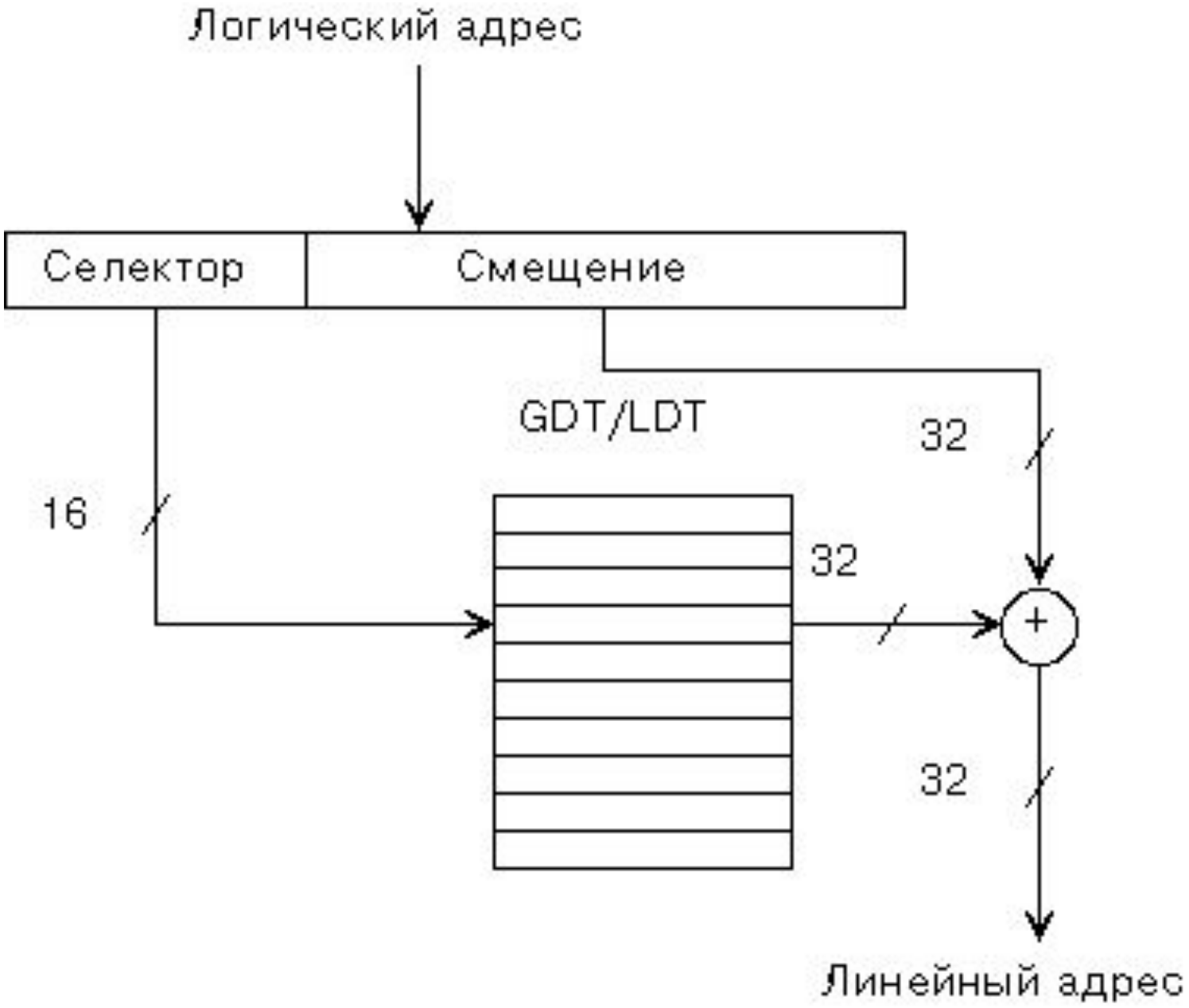
Преобразование адресов в защищённом режиме

Процессор i80386 и выше использует трёхступенчатую схему преобразования адреса. Программы используют логический адрес, состоящий из селектора и смещения (аналогично процессору i80286). Селектор полностью аналогичен используемому в процессоре i80286. Компонента смещения является 32-разрядной, т.к. допустимый размер сегмента значительно превышает 64 килобайта.

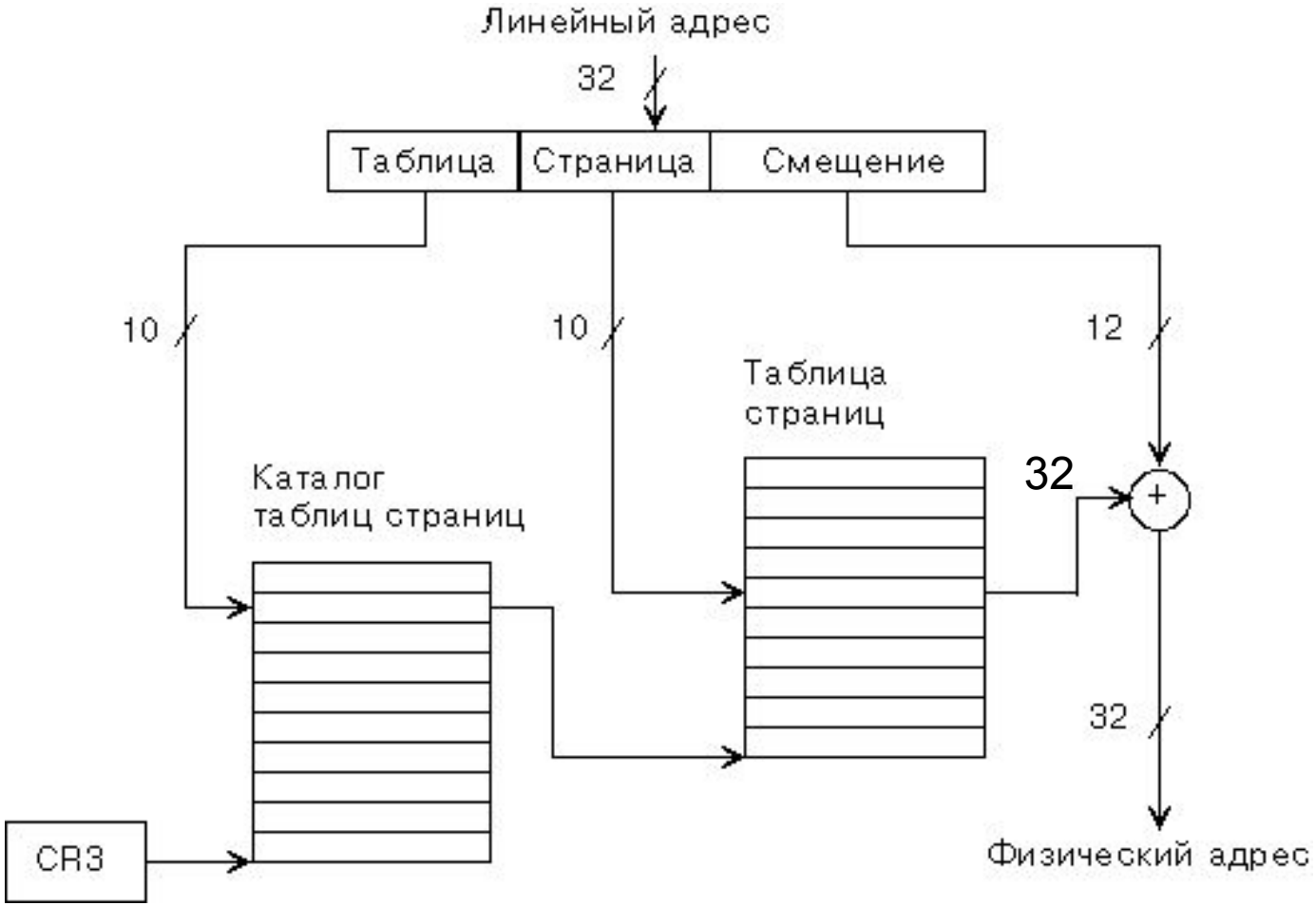
Уровень логического адреса - это первая ступень в схеме преобразования адресов.

Вторая ступень - получение из логического адреса 32-разрядного линейного адреса. Линейный адрес берётся из глобальной или локальной таблицы дескрипторов (GDT или LDT) в зависимости от соответствующего бита селектора (бит 2). Механизм получения линейного адреса напоминает механизм получения 24-разрядного физического адреса в процессоре i80286. Однако линейный адрес не отображается непосредственно на адресную шину памяти, то есть он не является физическим адресом.

Для получения из линейного адреса физического адреса используется третья ступень - механизм страничной адресации. С помощью этого механизма 20 старших бит линейного адреса используются для выбора блока памяти размером 4 килобайта. Такой блок называется страницей физической памяти. Оставшиеся 12 бит линейного адреса представляют собой смещение внутри страницы. Процесс преобразования логического адреса в линейный иллюстрируется рис. 4.

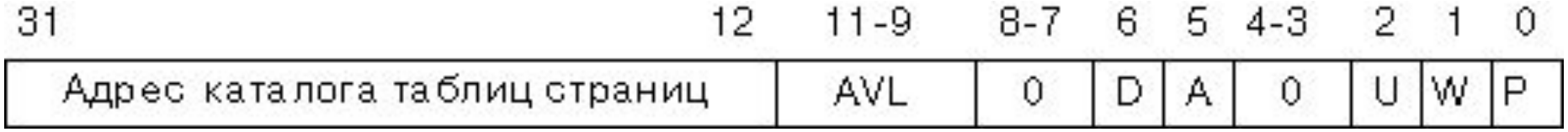

Логический адрес
Селектор
Смещение
GDT/LDT
32
16
32
+
32
Линейный адрес

Значение из поля индекса селектора используется в качестве индекса в таблице LDT или GDT для выборки 32-разрядного базового адреса. Этот базовый адрес складывается со второй компонентой логического адреса - смещением. В результате получается 32-разрядный линейный адрес. Преобразование линейного адреса в физический иллюстрируется рис. 5.



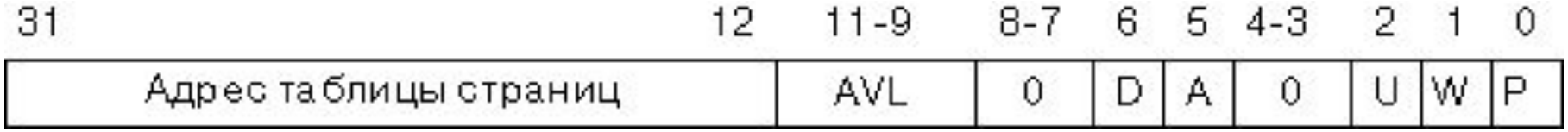
Процесс вычисления адреса страницы часто называют трансляцией страниц. Старшие 10 бит линейного адреса используются как индекс в таблице, называемой каталогом таблиц страниц. Расположение каталога таблиц страниц в физической памяти определяется содержимым системного регистра процессора CR3.
Каталог таблиц страниц содержит дескрипторы таблиц страниц, определяющие физический адрес таблиц страниц. В каталоге таблиц страниц всего может быть 1024 дескриптора. Самих же каталогов может быть сколько угодно, но в каждый момент времени используется только один - тот, на который указывает регистр CR3.
Следующие 10 бит линейного адреса предназначены для индексации таблицы страниц, выбранной с помощью старших 10 бит адреса. Таблица страниц содержит 1024 дескриптора, определяющих физические адреса страниц памяти. Размер одной страницы составляет 4 килобайта, т.е. 4096 байт.

Младшие 12 бит линейного адреса указывают смещение к адресуемому байту внутри страницы.
На рис. 6 представлен формат дескриптора таблицы страниц.



Дескриптор таблицы страниц.

Для представления старших 20 битов физического адреса таблицы страниц в дескрипторе используются биты 12-31. Младшие 12 битов адреса таблицы всегда равны нулю, таким образом, таблица страниц должна быть выровнена в памяти на границу 4096 байт (на границу страницы). Формат дескриптора страницы представлен на рис.7.



Дескриптор страницы.

Назначение бит 0-11 одинаково и для дескриптора таблицы страниц, и для дескриптора страницы. В таблице 1 приведено описание этих бит.

Номер бита	Назначение
0 (P)	Бит присутствия в памяти. Установлен в 1, если определяемая данным дескриптором таблица страниц находится в оперативной памяти. Этот бит используется для организации виртуальной памяти.
1 (W)	Разрешение записи. Если бит установлен в 1, то запись в страницы разрешена. Бит используется для организации защиты от записи на уровне страниц.
2 (U)	Пользователь/супервизор. Используется для разграничения доступа к страницам операционной системы (страницы супервизора) и страницам программ пользователя. Значение бита, равное 0, соответствует страницам супервизора, 1 - страницам программы пользователя.
3-4	Эти биты зарезервированы и должны быть установлены в 0 для совместимости со следующими моделями процессора.

5 (A)	Бит доступа. Он устанавливается процессором перед выполнением операций чтения страницы или записи в страницу.
6 (D)	Бит мусора. Устанавливается, если была выполнена запись в каталог или страницу.
7-8	Эти биты зарезервированы и должны быть установлены в 0 для совместимости со следующими моделями процессора.
9-12 (AVL)	Эти биты доступны для использования операционной системой (AVL - Available for use).

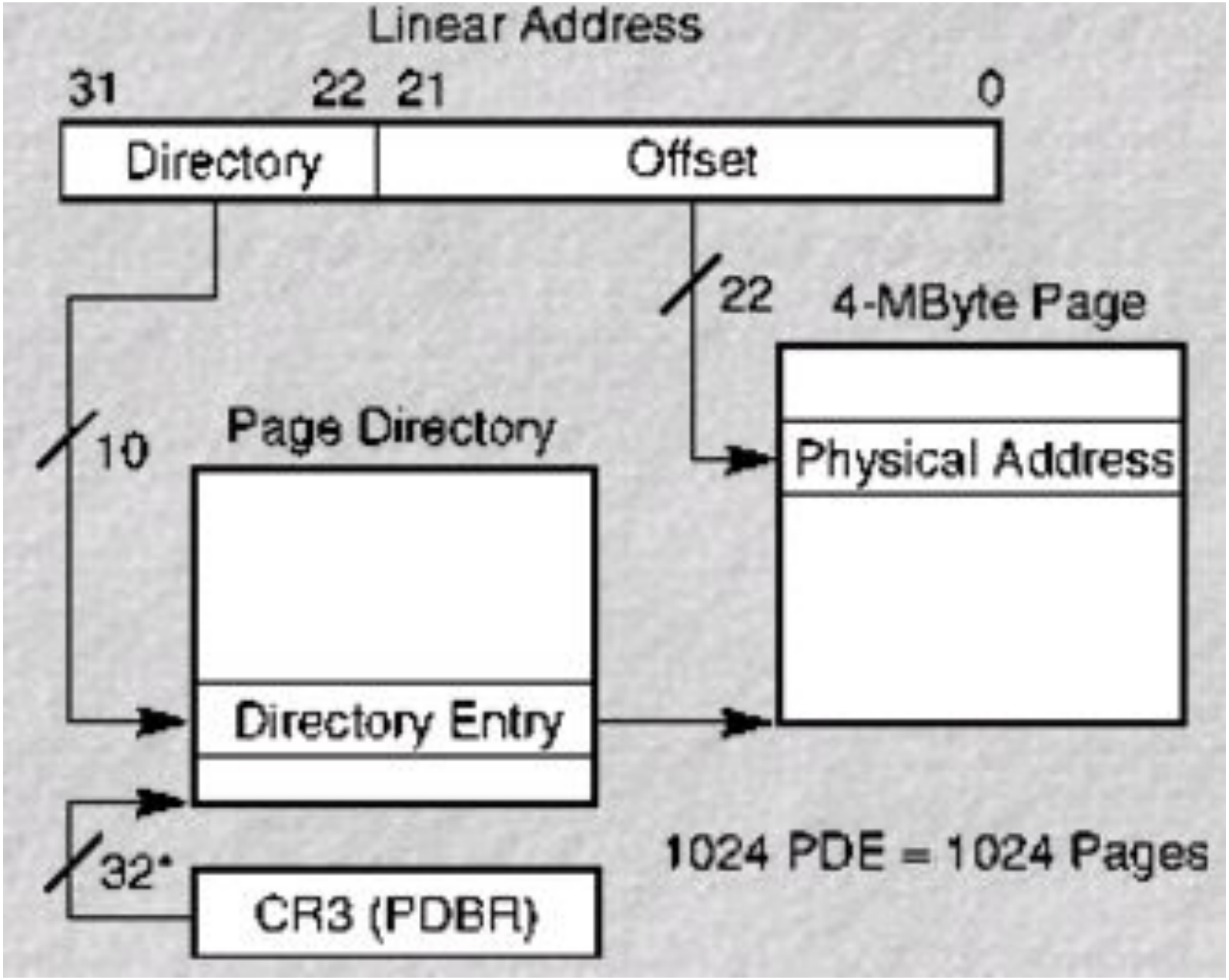
Для использования механизма трансляции страниц операционная система должна установить в 1 старший бит системного регистра CR0. Если этот бит не установлен в 1, физический адрес будет равен линейному, содержимое регистра адреса каталога таблиц страниц CR3 при этом для преобразования адреса использоваться не будет.
Включенный блок страничной трансляции адресов осуществляет трансляцию линейного адреса в физический страницами размером 4 Кбайт (для последних поколений процессоров также возможны страницы размером 2 или 4 Мбайт). Блок трансляции может включаться только в защищенном режиме.

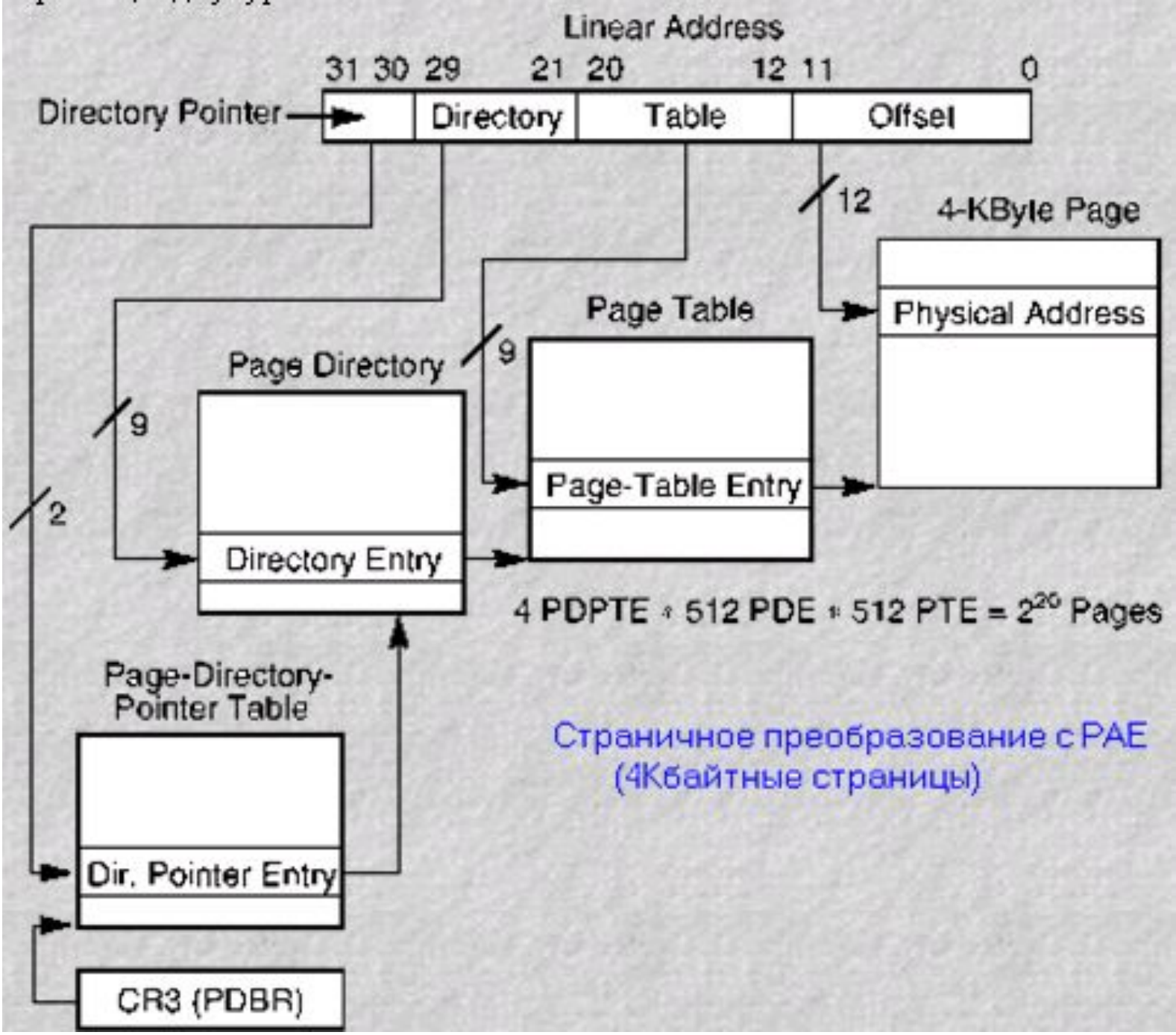
Каков объем виртуального адресного пространства? Для индекса дескриптора отведено 13 бит. Отсюда следует, что в дескрипторной таблице может быть до 8К дескрипторов. Однако в действительности их в два раза больше, так как программа может работать не с одной, а с двумя дескрипторными таблицами. Таким образом, всего программе могут быть доступны 2^{14} = 16 К дескрипторов, т.е. 16 К сегментов. Поскольку размер каждого сегмента, определяемый максимальной величиной смещения, может достигать 2^{32} = 4 Гбайт, объем виртуального адресного пространства оказывается равным 16К * 4 Г = 64 Тбайт (10^{12}).

Реально, однако, оперативная память компьютера с 32-разрядной адресной шиной не может быть больше 4 Гбайт, т.е. при сделанных выше предположениях (16 К сегментов размером 4 Гбайт каждый) в памяти может поместиться максимум один сегмент из более чем 16 тысяч. Где же будут находиться все остальные?

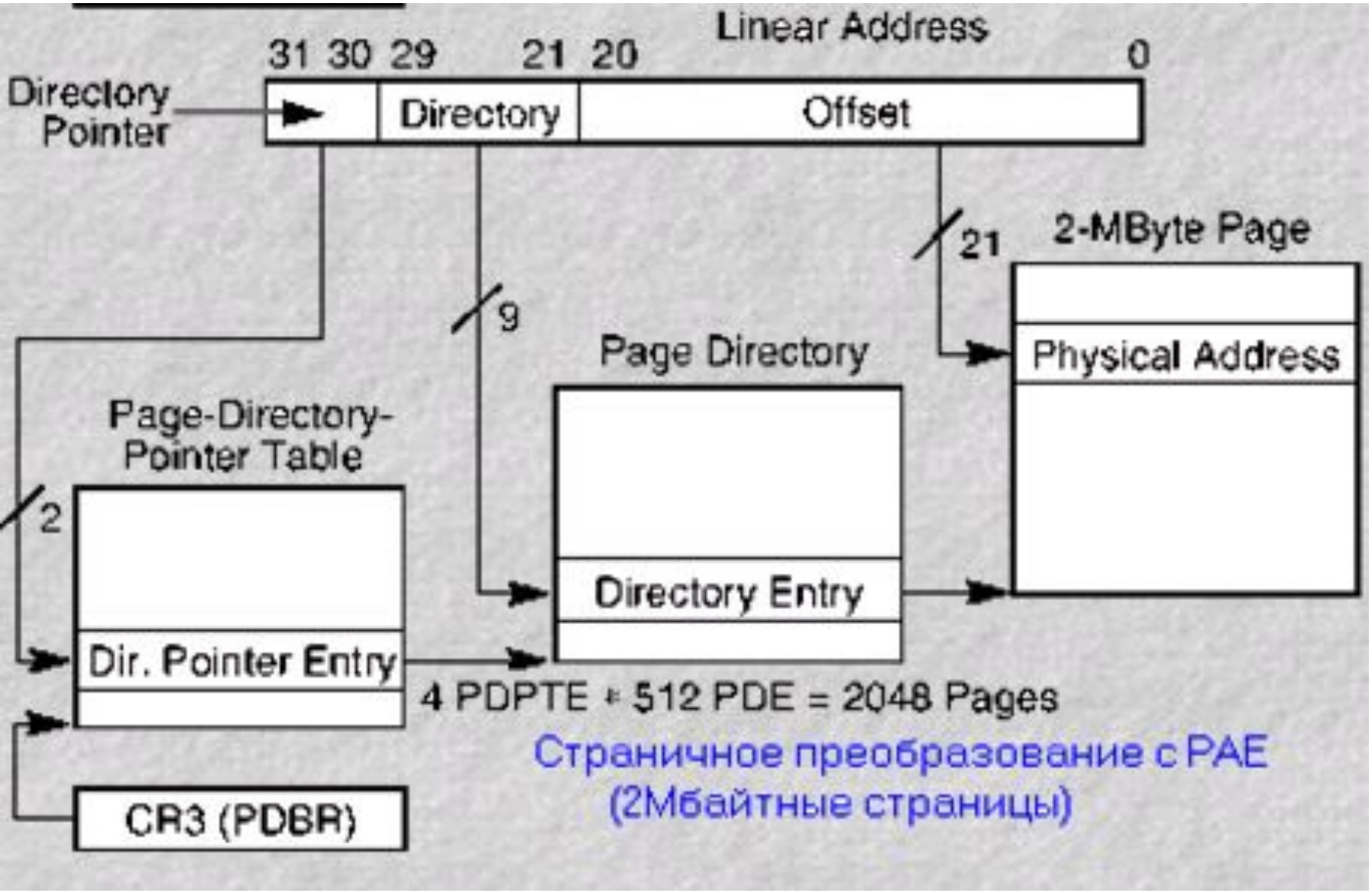
Полный объем виртуального пространства может быть реализован только с помощью многозадачной операционной системы, которая хранит все неиспользуемые в настоящий момент сегменты на диске, загружая их в память по мере необходимости. Разумеется, если мы хотим полностью реализовать возможности, заложенные в современные процессоры, нам потребуется диск довольно большого объема - 64 Тбайт.

В процессорах, начиная с Pentium, страницы могут иметь размер 4Кбайт или 4Мбайт (эта возможность называется *расширением размера страниц*), а в процессорах с архитектурой P6 при включенном *расширении физического адреса* - 4Кбайт или 2Мбайт. (Расширение физического адреса заключается в использовании 36-битного физического адреса вместо 32-битного.) Расширение размера страниц включается установкой бита 4 (Page Size Extension) в регистре CR4, а расширение физического адреса - установкой бита 5 (Physical Address Extension) в регистре CR4. Обе возможности работают только в защищенном режиме при включенной страничной трансляции адресов. Для страниц размером 4Мбайт действует упрощенная (одноуровневая) схема формирования физического адреса. В этом случае физический адрес (старшие 10 бит) страницы хранится непосредственно в каталоге таблиц. Младшие 22 бита линейного адреса задают смещение от начала страницы. Конечно, страницы большого размера неудобны для подкачки при работе с маленькими приложениями, но тот факт, что при включенном PSE (или PAE) в системе можно использовать страницы обоих размеров позволяет повысить эффективность работы: на страницах большого размера можно разместить код операционной системы, к которому часто обращаются все приложения и который не следует выгружать из памяти, при этом экономится место - не нужны промежуточные таблицы страниц.


Linear Address
31
22
21
0
Directory
Offset
22
4-MByte Page
Physical Address
10
Page Directory
Directory Entry
32*
CR3 (PDBR)
1024 PDE = 1024 Pages



Страничное преобразование с PAE
(4Кбайтные страницы)


Linear Address
31 30 29 21 20 0
Directory Pointer
Directory
Offset
2-MByte Page
21
9
Page Directory
Physical Address
Page-Directory-Pointer Table
2
Directory Entry
Dir. Pointer Entry
4 PDPTE * 512 PDE = 2048 Pages
CR3 (PDBR)
Страничное преобразование с PAE
(2Мбайтные страницы)