

4.4 ЗАСОБИ ПІДТРИМКИ МЕХАНІЗМІВ ВІРТУАЛЬНОЇ ПАМ'ЯТІ

4.4.1 Віртуальний адресний простір

При роботі процесора Pentium у сегментному режимі в розпорядженні програміста є віртуальний адресний простір, що представляється сукупністю сегментів.

Кожен сегмент віртуальної пам'яті процесу має опис, який називається дескриптором сегменту. Дескриптор сегменту має розмір 8 байт і містить усі характеристики сегменту, необхідні для перевірки правильності доступу до нього і знаходження його у фізичному адресному просторі (рис. 4.9).

Структура дескриптора, яка підтримується в процесорі Pentium, склалася історично. Багато що в ній пов'язане із забезпеченням сумісності з попередніми процесорами сімейства x86. Саме цим пояснюється те, що базова адреса сегменту представлена в дескрипторі у вигляді трьох частин, а розмір сегменту займає два поля.

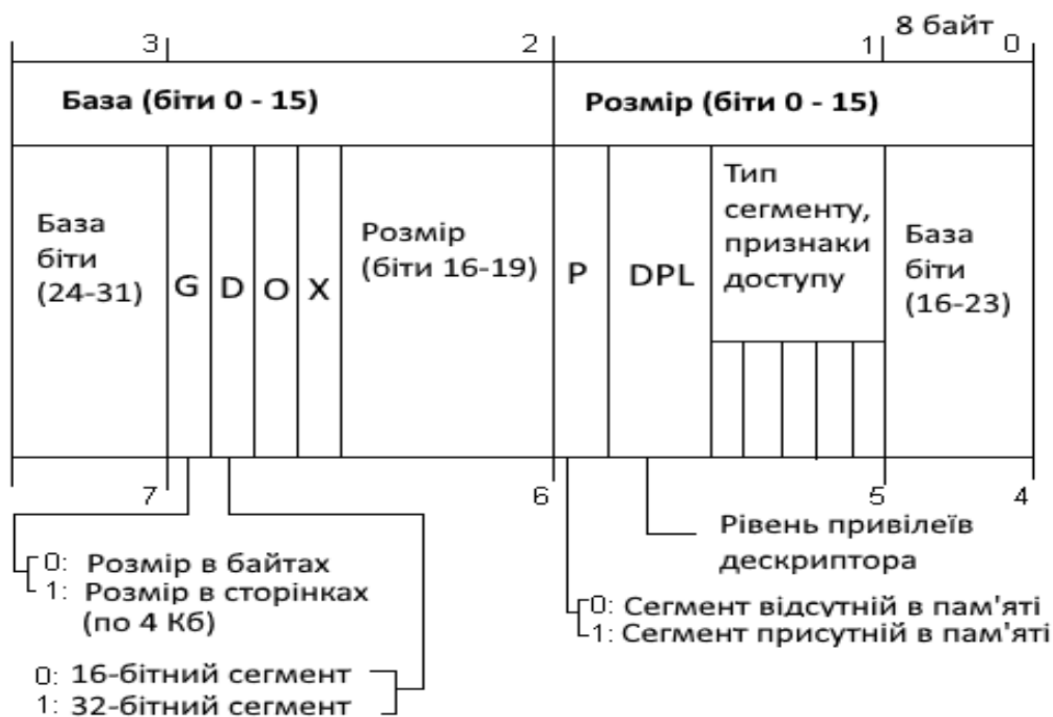


Рисунок 4.9 – Формат дескриптора сегмента даних або коду

Нижче перераховані основні поля дескриптора.

База – базова адреса сегменту (32 біти = 16+8+8), розділена на три частини із-за сумісності з i286, в якому це поле має тільки 24 біта.

Розмір – розмір сегменту (20 біт = 16+4), рознесений на дві частини.

G (Granularity) – одиниця виміру розміру сегменту, один біт. Якщо $G=0$, то розмір заданий в байтах і тоді сегмент не може бути більше 64 Кб, якщо $G=1$, то розмір сегменту вимірюється в сторінках по 4 Кб.

Байт доступу (5-й байт дескриптора) містить інформацію, яка використовується для ухвалення рішення про можливість або неможливість звернення до цього сегменту. Біт P (Present) визначає, чи знаходиться відповідний сегмент в даний момент в пам'яті ($P=1$) або він вивантажений на диск ($P=0$). Поле DPL (Descriptor Privilege Level, 2 біта) містить дані про рівень привілеїв, необхідний для доступу до сегменту (0-3). Інші 5 бітів байта доступу залежать від типу сегменту і визначають спосіб, яким можна використати цей сегмент (тобто читати, писати, виконувати). Відрізняються три основні типи сегментів: сегмент даних; кодовий сегмент; системний сегмент (GDT, TSS тощо).

Дескриптори сегментів об'єднуються в таблиці. Процесор Pentium для управління пам'яттю підтримує два типи таблиць дескрипторів сегментів:

- глобальну таблицю дескрипторів (Global Descriptor Table, GDT), яка призначена для опису сегментів операційної системи і загальних сегментів для усіх прикладних процесів, наприклад сегментів міжпроцесної взаємодії;
- локальну таблицю дескрипторів (Local Descriptor Table, LDT), яка містить дескриптори сегментів окремого призначеного для користувача процесу.

Таблиця GDT одна, а таблиць LDT стільки, скільки в системі виконується задач (процесів). При цьому в кожен момент часу операційною системою і апаратними засобами процесора використовується тільки одна з таблиць LDT, а саме та, яка відповідає виконуваному в даний момент призначеному для користувача процесу. Таблиця GDT описує загальну частину віртуального адресного простору процесів, а LDT – індивідуальну частину для кожного процесу. Таблиці GDT і LDT розміщені в оперативній пам'яті у вигляді окремих сегментів. Сегменти LDT і GDT містять системні дані, тому їх дескриптори зберігаються в таблиці GDT. Таким чином, таблиця GDT разом із записами про інші сегменти містить запис про саму себе, а також про усі таблиці LDT.

У кожен момент часу в спеціальних регістрах GDTR і LDTR зберігається інформація про місце розташування і розміри глобальної таблиці GDT і активної

таблиці LDT відповідно. Регістр GDTR містить 32-розрядну фізичну адресу початку сегменту GDT в пам'яті, а також 16-бітовий розмір цього сегменту (рис. 4.10). Регістр LDTR вказує на розташування сегменту LDT в оперативній пам'яті опосередковано – він містить індекс дескриптора в таблиці GDT, в якому міститься адреса таблиці LDT і її розмір.



Рисунок 4.10 – Формат регістра GDTR

Процес звертається до фізичної пам'яті за віртуальною адресою (селектор, зміщення). Селектор однозначно визначає віртуальний сегмент, до якого належить шукана адреса, тобто він може інтерпретуватися як номер сегменту, а зміщення, як це і виходить з його назви, фіксує положення шуканої адреси відносно початку сегменту. Зміщення задається в машинній інструкції, а селектор поміщається в один із сегментних регістрів процесора. Під зміщення відводиться 32 біти, що забезпечує максимальний розмір сегменту 4 Гб (2³²).

Селектор витягується з одного з шести 16-розрядних сегментних регістрів процесора (CS, SS, DS, ES, FS або GS) залежно від типу команди і стадії її виконання – вибірки коду команди або даних. Наприклад, при зверненні до пам'яті під час вибірки наступної команди використовується селектор з сегментного регістра коду CS, а при записі результатів в сегмент даних процесу селектори витягуються з сегментних регістрів даних DS або ES. Якщо ж дані записуються в стек по команді PUSH, то механізм віртуальної пам'яті витягає селектор з сегментного регістра стека SS.

Селектор складається з трьох полів (рис. 4.11):

- індексу, який задає послідовний номер дескриптора в таблиці GDT або LDT (13 біт);
- покажчика типу використовуваної таблиці дескрипторів: GDT або LDT (1 біт);
- необхідного рівня привілеїв – RPL (2 біти), це поле використовується механізмом захисту даних (0 – ядро, 1 – системні вузли, 2 – бібліотеки спільного

доступу, 3 – програми користувача). Рівні привілейованості забороняють виконуваному коду звернутися до нижчого рівня. Віртуальний адресний простір процесу складається з усіх сегментів, описаних в загальній для усіх процесів таблиці GDT, і сегментів, описаних в його власній таблиці LDT. Розрядність (13) поля індексу визначає максимальне число глобальних і локальних сегментів процесу – по 8 Кб (2¹³) сегментів кожного типу (2*2¹³), всього 16 Кб сегментів.

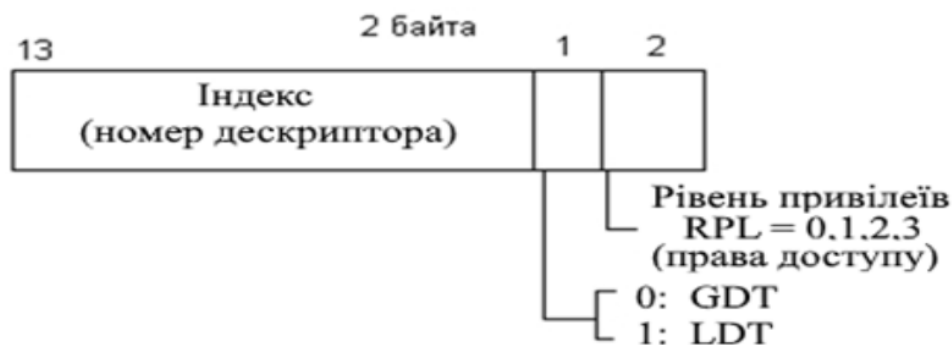


Рисунок 4.11 – Формат селектора в Pentium

З урахуванням максимального розміру сегменту – 4 Гб – кожен процес при чисто сегментній організації віртуальної пам'яті (без включення сторінкового механізму) може працювати у віртуальному адресному просторі в 64 Тбайт (16Кб*232 = 214*232 = 246 = 26*240).

Кожен дескриптор в таблицях GDT і LDT має розмір 8 байт, тому максимальний розмір кожної з цих таблиць – 64 Кб (8 байт * 8 Кб дескрипторів). З приведенного опису видно, що процесор Pentium забезпечує підтримку роботи ОС у двох відношеннях:

- підтримує роботу віртуальної пам'яті за рахунок швидкого апаратного способу перетворення віртуальної адреси у фізичну;
- забезпечує захист даних і кодів різних додатків.

4.4.2 Перетворення адрес

Тепер простежимо, яким чином віртуальний простір в 64 Тбайт відображається на фізичний простір розміром в 4 Гб. Механізм відображення перетворює віртуальну адресу, яка представлена селектором, що знаходиться в одному з сегментних регістрів, і зміщенням, витягнутим з відповідного поля машинної інструкції, в лінійну фізичну адресу.

Розглянемо спочатку випадок, коли віртуальна адреса належить до одного з сегментів, дескриптори яких містяться в таблиці GDT (рис. 4.12).

1. Значення селектора вказує механізму перетворення адрес, що віртуальна адреса належить до сегменту, що описується в таблиці GDT. Місцезнаходження таблиці GDT система визначає з реєстра GDTR, в якому зберігається повна 32-бітова базова фізична адреса таблиці. Процесор складає базову адресу таблиці, взяту з реєстра GDTR, із зрушенням на 3 розряди вліво (множення на 8 відповідно до числа байтів в одному дескрипторі сегменту) значенням поля індексу з селектора. Результатом є фізична адреса дескриптора сегменту, до якого належить задана віртуальна адреса.

2. За обчисленою адресою процесор витягає з пам'яті дескриптор потрібного сегменту.

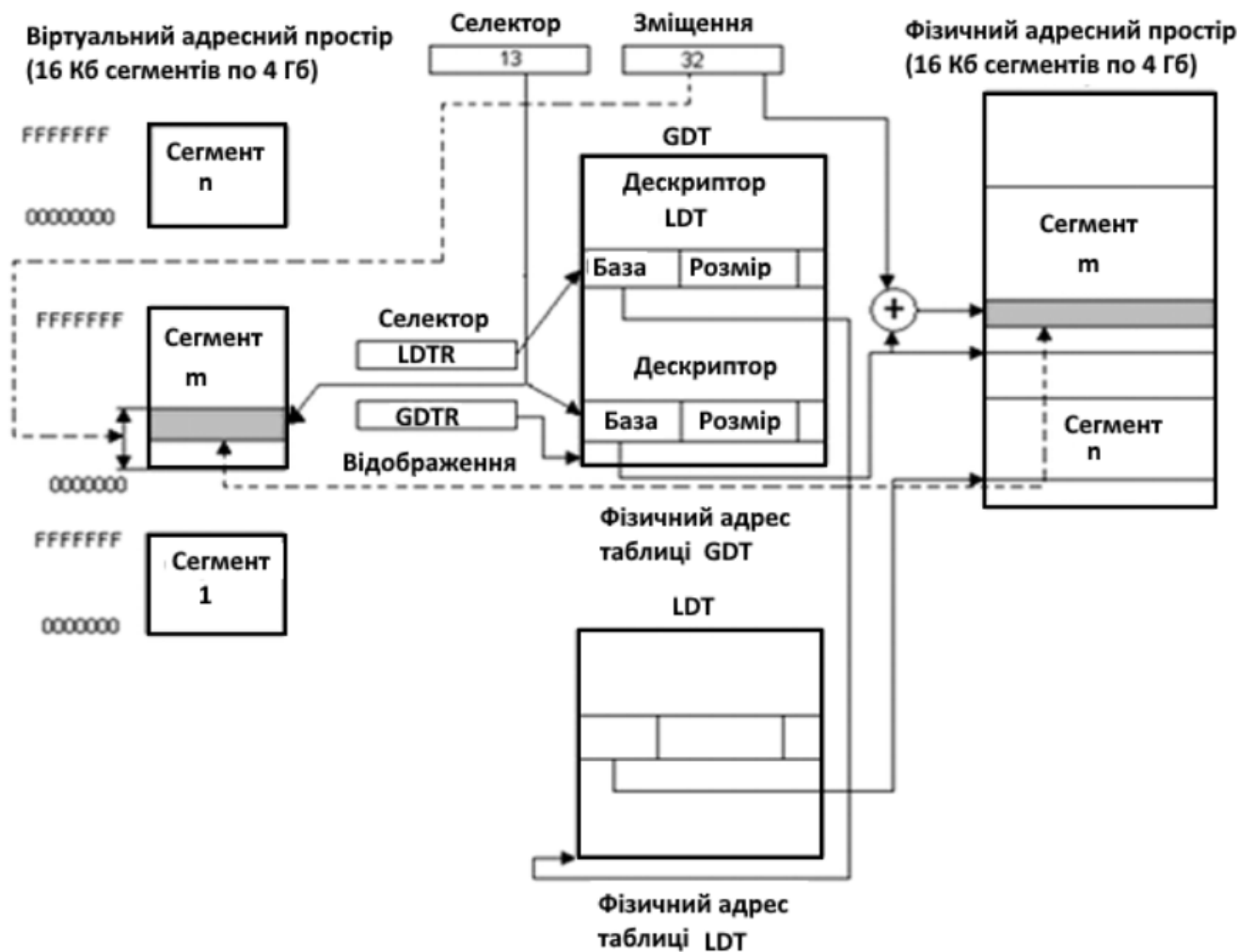


Рисунок 4.12 – Механізм перетворення віртуальної адреси у фізичну при роботі процесора в сегментному режимі

3. Здійснюється перевірка можливості виконання заданої операції доступу за заданою віртуальною адресою:

- спочатку процесор визначає правильність адреси, порівнюючи зміщення, задане у віртуальній адресі, з розміром сегменту, витягнутим з регістра LDTR (у разі виходу адреси за межі сегменту відбувається переривання);
- потім процесор перевіряє права доступу процесу до сегменту пам'яті;
- далі перевіряється наявність сегменту у фізичній пам'яті (якщо біт $P = 0$, тобто сегмент відсутній у фізичній пам'яті, то відбувається переривання).

4. Якщо всі три умови виконані, то доступ за заданою віртуальною адресою дозволений. Виконується перетворення віртуальної адреси у фізичну шляхом складання базової адреси сегменту, витягнутої з дескриптора, і зміщення, заданого в інструкції. Виконується задана операція над елементом фізичної пам'яті за цією адресою.

У разі, коли селектор у віртуальній адресі вказує не на таблицю GDT, а на таблицю LDT, процедура обчислення фізичної адреси дещо ускладнюється. Це пов'язано з тим, що регістр LDTR на відміну від GDTR вказує на розміщення таблиці сегментів не прямо, а побічно. У LDTR міститься індекс дескриптора сегмента LDT. Тому в процедурі перетворення адрес з'являється додатковий етап

- визначення базової адреси таблиці LDT. На підставі базової адреси таблиці GDT, узятій з регістра GDTR, і селектора, взятого з регістра LDTR, обчислюється зміщення в таблиці GDT, яке і є адресою дескриптора сегменту LDT.

З дескриптора витягається базова адреса таблиці LDT, і з цієї миті робота механізму відображення повністю аналогічна описаному вище перетворенню віртуальної адреси за допомогою таблиці GDT. Тобто, на підставі базової адреси таблиці LDT і селектора процесу, заданого в одному з сегментних регістрів, обчислюється зміщення в таблиці LDT і визначається базова адреса дескриптора шуканого сегменту. З цього дескриптора витягається базова адреса сегменту, який складається зі зміщенням з віртуальної адреси, що і дає в результаті шукану фізичну адресу.

Таким чином, для використання сегментного механізму процесора Pentium операційній системі необхідно сформувати таблиці GDT і LDT, завантажити їх в

пам'ять (спершу досить завантажити тільки таблицю GDT), завантажити покажчики на ці таблиці в реєстри GDTR і LDTR і вимкнути сторінкову підтримку.

Операційна система може відмовитися від використання засобів сегментації процесора Pentium. У такому разі їй достатньо призначити кожному процесу тільки по одному сегменту і занести у відповідні таблиці LDT по одному дескриптору. Віртуальний адресний простір процесу складається з одного сегменту завдовжки максимум в 4 Гб. Оскільки вивантаження процесів на диск здійснюватиметься повністю, віртуальна пам'ять вироджується в окремому випадку у свопінг.

4.4.3 Сегментно-сторінковий механізм

Включення сторінкового механізму відбувається, якщо в реєстрі управління CR0 найстарший біт PG встановлений в одиницю. При включеній системі управління сторінками паралельно продовжує працювати і описаний вище сегментний механізм, проте, як буде показано нижче, сенс його роботи міняється.

Віртуальний адресний простір процесу при сегментно-сторінковому режимі роботи процесора обмежується розміром 4 Гб, і при 4К-байтній сторінці, сегмент може містити 1 млн сторінок (4Гб/4Кб). У цьому просторі визначені віртуальні сегменти процесу (рис. 4.13). Оскільки тепер усі віртуальні сегменти розділяють один віртуальний адресний простір, то можливе їх накладення, оскільки процесор не контролює такі ситуації, залишаючи цю проблему операційній системі.

Для реалізації механізму управління сторінками як фізичний, так і віртуальний адресні простори розбиті на сторінки розміром 4 Кб. Починаючи з моделі Pentium в процесорах Intel існує можливість використання сторінок і по 4 Мб, але подальший виклад орієнтується на традиційний розмір сторінки в 4 Кб. Всього у віртуальному адресному просторі в сегментно-сторінковому режимі налічується 1 Мб ($2^{20} = 4Гб/4Кб$) сторінок. Незважаючи на наявність декількох віртуальних сегментів, увесь віртуальний адресний простір процесу має загальне розбиття на сторінки, так що нумерація віртуальних сторінок наскрізна.

Віртуальна адреса як і раніше представляє собою пару: селектор, який визначає номер віртуального сегменту, і зміщення усередині цього сегменту. Перетворення віртуальної адреси виконується в два етапи: спочатку працює сегментний механізм, а

потім результат його роботи поступає на вхід сторінкового механізму, який і обчислює шукану фізичну адресу.

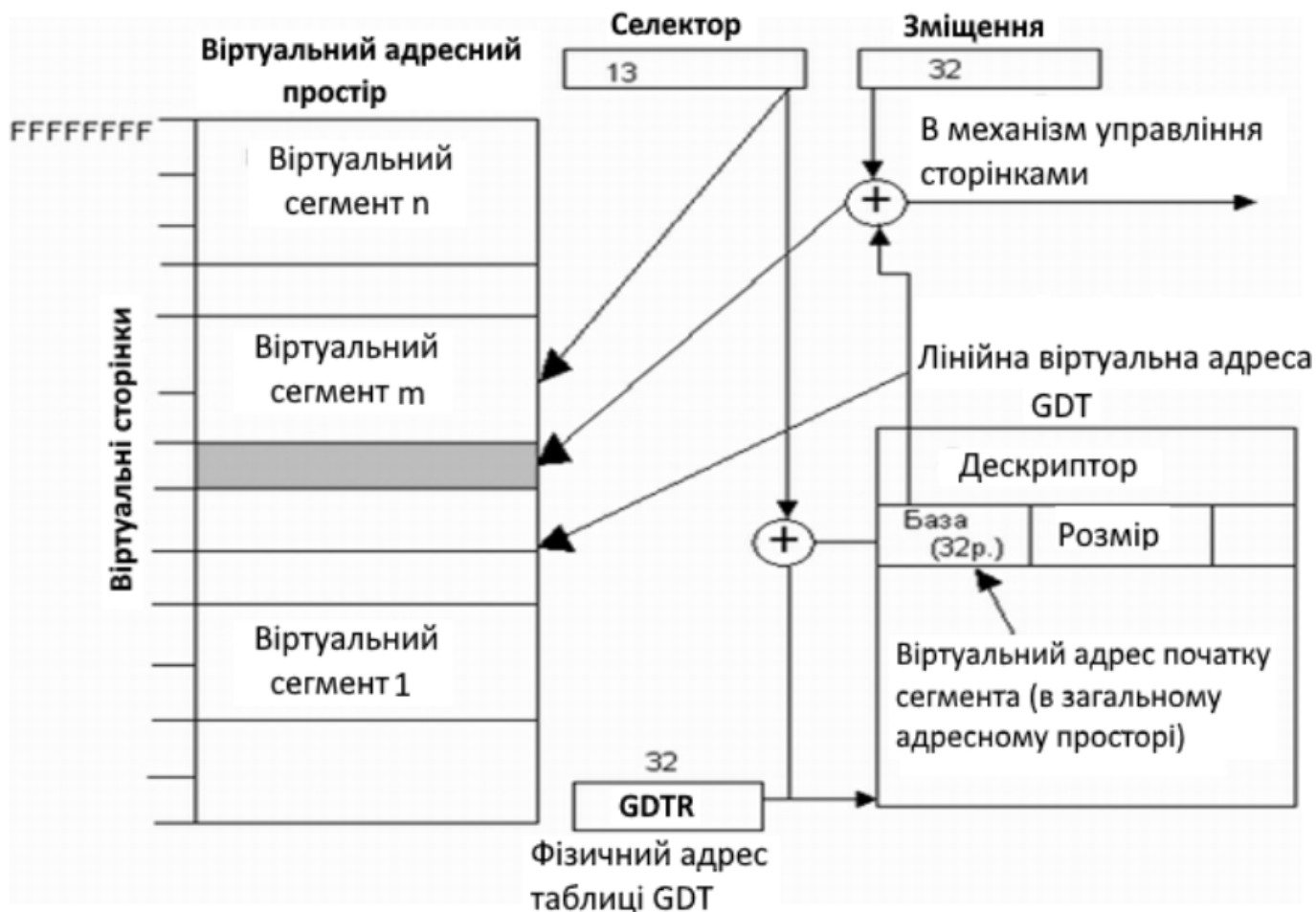


Рисунок 4.13 – Робота механізму в сегментно-сторінковому режимі

Робота сегментного механізму в даному випадку багато в чому повторює його роботу при відключеному сторінковому механізмі. На підставі значення індексу в селекторі вибирається потрібний дескриптор з таблиці GDT або LDT. З дескриптора витягається базова адреса сегменту і складається зі зміщенням.

Дескриптори і таблиці мають ту ж структуру. Проте є і принципова відмінність, полягає в інтерпретації утримуваного поля базової адреси в дескрипторах сегментів. Якщо раніше дескриптор сегменту містив базову адресу сегменту у фізичній пам'яті і при складанні цієї адреси зі зміщенням з віртуальної адреси виходила фізична адреса, то тепер дескриптор містить базову адресу сегменту у віртуальному адресному

просторі, і в результаті його складання зі зміщенням виходить лінійна віртуальна адреса.

Результуюча лінійна 32-розрядна віртуальна адреса передається сторінковому механізму для подальшого перетворення. Так як розмір сторінки дорівнює 4 Кб (2¹²), то в адресі можна легко виділити номер віртуальної сторінки (старші 20 розрядів) і зміщення в сторінці (молодші 12 розрядів).

Для відображення віртуальної сторінки у фізичну досить побудувати таблицю сторінок, кожен елемент якої – дескриптор віртуальної сторінки – містив би номер фізичної сторінки, яка відповідала б їй, і її атрибутам (рис. 4.14). Двадцять розрядів, в яких знаходиться номер сторінки, можуть інтерпретуватися і як базова адреса сторінки в пам'яті, яку необхідно доповнити 12 нулями, оскільки молодші 12 розрядів базової адреси сторінки завжди дорівнюють нулю. Окрім номера сторінки, дескриптор сторінки містить також поля, близькі за змістом відповідним полям дескриптора сегменту:

- P – біт присутності сторінки у фізичній пам'яті;
- W – біт дозволу запису в сторінку;
- U – біт користувач/супервізор;
- A – ознака доступу, що мав місце до сторінки;
- D – ознака модифікації вмісту сторінки;
- PWT і PCD – управляють механізмом кешування сторінок (введені починаючи з процесора i486);
- AVL – резерв для потреб операційної системи (AVaiLable for use).

20	3	2	1	1	1	1	1	1	1
№ сторінки	AVL	0	D	A	PCD	PWT	U	W	P

Рисунок 4.14 – Формат дескриптора сторінки

При невеликому розмірі сторінки процесора Pentium порівняно з розмірами адресних просторів таблиця сторінок повинна займати в пам'яті значне місце – 4 байт * 1 Мб = 4 Мб. Це надто багато для нинішніх моделей персональних комп'ютерів, тому в процесорі Pentium використовується ділення усієї таблиці сторінок на частини

– розділи по 1024 дескриптори. Розмір розділу вибраний так, щоб один розділ займав одну фізичну сторінку(1024 * 4 байт – 4 Кб). Таким чином таблиця сторінок ділиться на 1024 розділи.

Щоб постійно не зберігати в пам'яті всі розділи, створюється таблиця розділів (каталог сторінок), що складається з дескрипторів розділів, які мають таку ж структуру, що і дескриптори сторінок. Максимальний розмір таблиці розділів складає 4 Кб, тобто одна сторінка. Віртуальні сторінки, що містять

розділи, як і усі інші сторінки, можуть вивантажуватися на диск. Віртуальна сторінка, що зберігає таблицю розділів, завжди знаходиться у фізичній пам'яті, і номер її фізичної сторінки вказується в спеціальному управляючому реєстрі CR3 процесора. Перетворення лінійної віртуальної адреси у фізичну відбувається таким чином (рис. 4.15).

Поле номеру віртуальної сторінки (старші 20 розрядів) ділиться на дві рівні частини по 10 розрядів – поле номера розділу і поле номера сторінки в розділі. На підставі заданого в реєстрі CR3 номера фізичної сторінки, що зберігає таблицю розділів, і зміщення в цій сторінці, що задається полем номеру розділу, процесор знаходить дескриптор віртуальної сторінки розділу.

Відповідно до атрибутів цього дескриптора визначаються права доступу до сторінки, а також наявність її у фізичній пам'яті. Якщо сторінки немає в оперативній пам'яті, то відбувається переривання, в результаті якого операційна система повинна виконати завантаження необхідної сторінки в пам'ять.

Після того як сторінка, що містить потрібний розділ, завантажена, з неї витягається дескриптор сторінки даних, номер якої вказаний в лінійній віртуальній адресі. І нарешті, на підставі базової адреси сторінки, отриманої з дескриптора, і зміщення, заданого в лінійній віртуальній адресі, обчислюється шукана фізична адреса. Таким чином, при доступі до сторінки в процесорі використовується дворівнева схема адресації сторінок, яка хоч і уповільнює перетворення, але дозволяє використати сторінковий механізм для таблиці сторінок, що істотно зменшує об'єм фізичної пам'яті, потрібної для її зберігання.

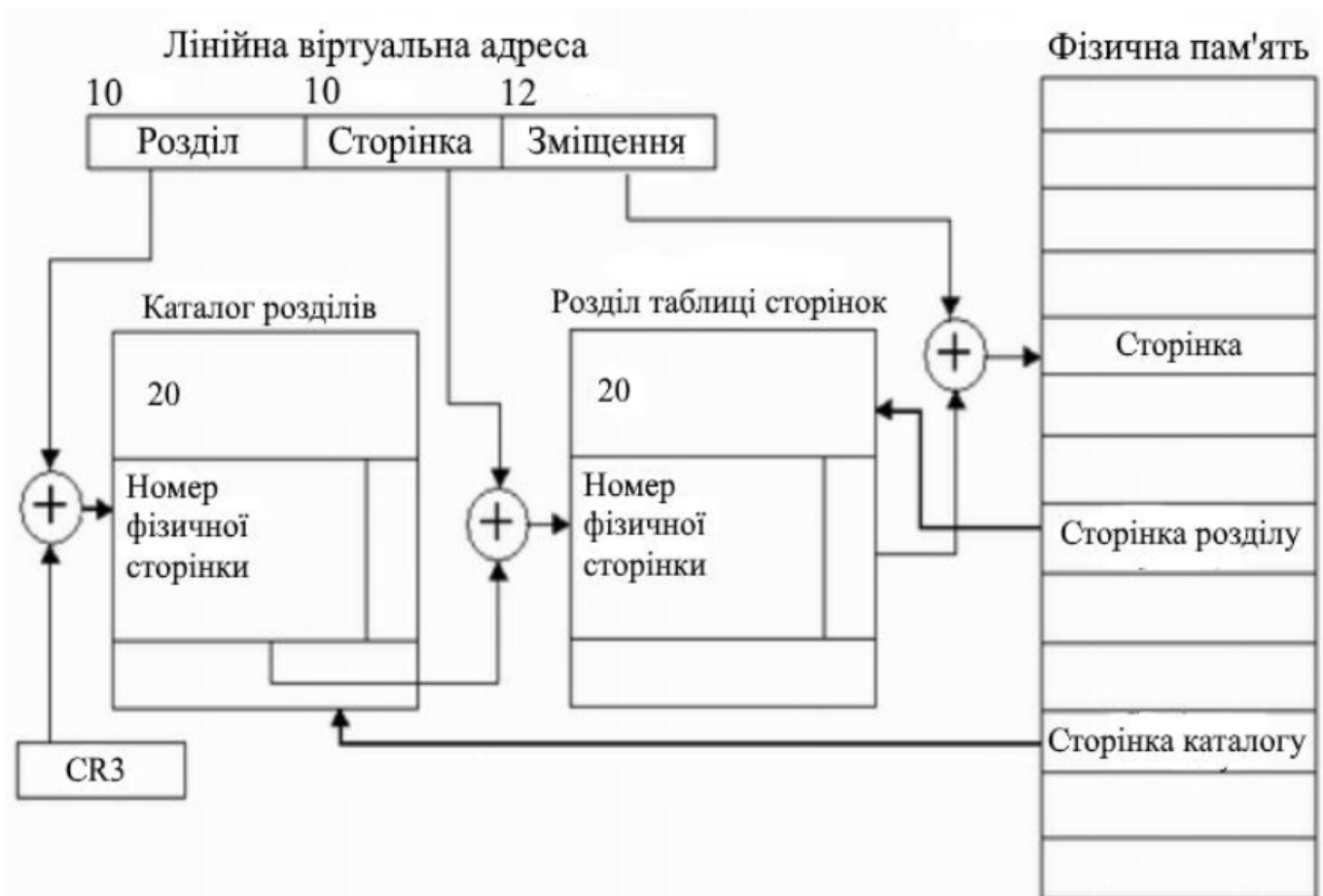


Рисунок 4.15 – Перетворення лінійної віртуальної адреси у фізичну адресу

Для прискорення перетворення адрес у блоці управління сторінками використовується асоціативна пам'ять, в якій знаходиться 32 дескриптори активно використовуваних сторінок, що дозволяє за номером віртуальної сторінки швидко знайти номер фізичної сторінки без звернення до таблиць розділів і сторінок.