**МЕХАНИЗМЫ АДРЕСАЦИИ ПАМЯТИ В IA-32**

**ПРЕДСТАВЛЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА ПАМЯТИ МИКРОПРОЦЕССОРОВ**

* ФИЗИЧЕСКОЕ АДРЕСНОЕ ПРОСТРАНСТВО – ОДНОМЕРНЫЙ, МАССИВ БАЙТОВ (НЕСТРУКТУРИРОВАННАЯ, ПЛОСКАЯ (FLAT) МОДЕЛЬ ПАМЯТИ). КАЖДОМУ АДРЕСУ В ЭТОМ ПРОСТРАНСТВЕ СООТВЕТСТВУЕТ УНИКАЛЬНОЕ ЗНАЧЕНИЕ СИГНАЛОВ НА АДРЕСНЫХ ВХОДАХ ПАМЯТИ
* ЛОГИЧЕСКОЕ АДРЕСНОЕ ПРОСТРАНСТВО (ЛАП) – ОРГАНИЗУЕТСЯ ПРИ РАЗРАБОТКЕ ПРОГРАММЫ C ИСПОЛЬЗОВАНИЕМ РАЗЛИЧНЫХ СПОСОБОВ АДРЕСАЦИИ И СТРУКТУРИРУЕТСЯ В ВИДЕ НАБОРОВ БАЙТОВ, СЕГМЕНТОВ ИЛИ СТРАНИЦ

* ИНОГДА ЛАП АССОЦИИРУЕТСЯ С ВИРТУАЛЬНЫМ АДРЕСНЫМ ПРОСТРАНСТВОМ – ОБЩЕЕ АДРЕСНОЕ ПРОСТРАНСТВО ВСЕХ ВИДОВ ПАМЯТИ (ОП+ДИСК)

**ОРГАНИЗАЦИЯ ЛОГИЧЕСКОГО АДРЕСНОГО ПРОСТРАНСТВА**

* ПЛОСКОЕ, ЛИНЕЙНОЕ **ЛАП** – МАССИВ БАЙТОВ. ЛОГИЧЕСКИЙ АДРЕС (ИНДЕКС В МАССИВЕ) СОВПАДАЕТ С ФИЗИЧЕСКИМ АДРЕСОМ
* СЕГМЕНТИРОВАННОЕ **ЛАП** - СОВОКУПНОСТЬ ОЛАСТЕЙ СМЕЖНЫХ ЯЧЕЕК ПАМЯТИ (МОГУТ ПЕРЕКРЫВАТЬСЯ, ИМЕТЬ ОДИНАКОВЫЙ ИЛИ РАЗЛИЧНЫЙ РАЗМЕР). ЯВЛЯЮТСЯ НЕЗАВИСИМЫМИ И ОТДЕЛЬНО АДРЕСУЕМЫМИ ЕДИНИЦАМИ ПАМЯТИ. ЛОГИЧЕСКИЙ АДРЕС СОСТОИТ ИЗ СЕЛЕКТОРА СЕГМЕНТА И СМЕЩЕНИЯ. ФИЗИЧЕСКИЙ (ЛИНЕЙНЫЙ)АДРЕС – НА ВЫХОДЕ БЛОКА СЕГМЕНТАЦИИ MMU
* СЕГМЕНТИРОВАНИЕ ОБЕСПЕЧИВАЕТ ПРОСТОТУ ЛОГИЧЕСКОГО СТРУКТУРИРОВАНИЯ ПРОГРАММ НА ОБЛАСТИ КОДА, ДАННЫХ И СТЕКА
* СТРАНИЧНОЕ **ЛАП** – СОВОКУПНОСТЬ НЕПРЕРЫВНЫХ ОБЛАСТЕЙ ПАМЯТИ ОДИНАКОВОГО РАЗМЕРА. ЛОГИЧЕСКИЙ АДРЕС СОСТОИТ ИЗ НОМЕРА СТРАНИЦЫ И СМЕЩЕНИЯ. ФИЗИЧЕСКИЙ АДРЕС – НА ВЫХОДЕ БЛОКА СТРАНИЧНОГО ПРЕОБРАЗОВАНИЯ MMU
* СЕГМЕНТНО-СТРАНИЧНОЕ ЛАП – СОВОКУПНОСТЬ СЕГМЕНТОВ, СОСТОЯЩИХ ИЗ СТРАНИЦ. ЛОГИЧЕСКИЙ АДРЕС СОСТОИТ ИЗ СЕЛЕКТОРА СЕГМЕНТА И СМЕЩЕНИЯ. ЛИНЕЙНЫЙ АДРЕС С ВЫХОДА БЛОКА СЕГМЕНТАЦИИ ПЕРЕДАЕТСЯ В БЛОК СТРАНИЧНОГО ПРЕОБРАЗОВАНИЯ, КОТОРЫЙ ОПРЕДЕЛЯЕТ ФИЗИЧЕСКИЙ АДРЕС СТРАНИЦЫ И СМЕЩЕНИЕ В НЕЙ

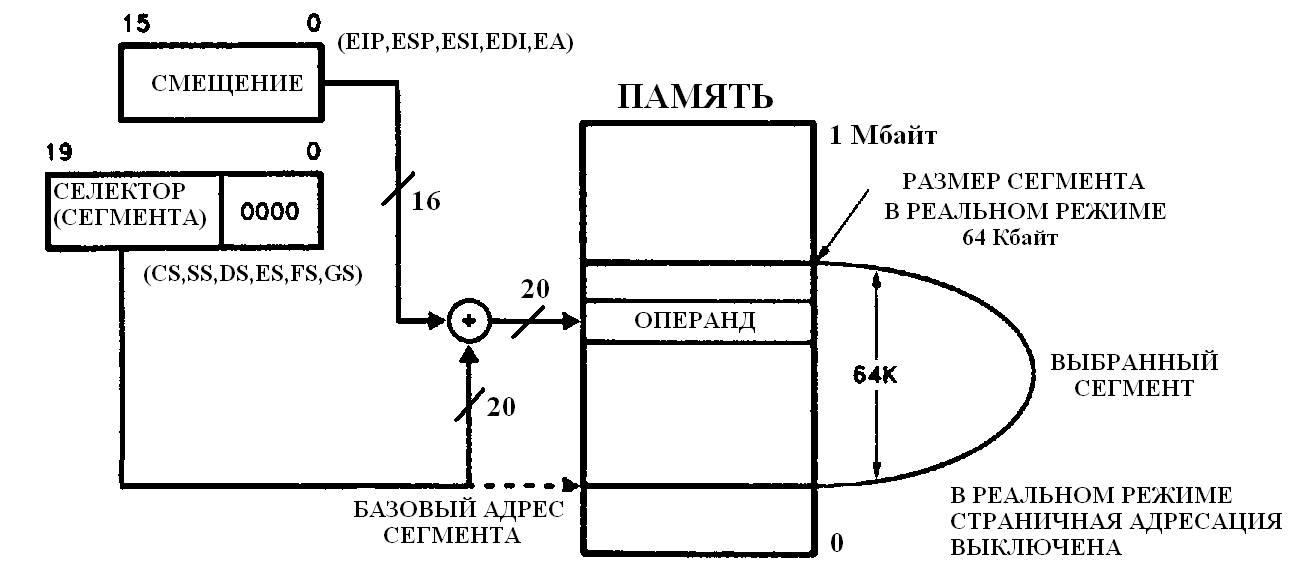
**АДРЕСАЦИЯ В IA-32**

* ЛОГИЧЕСКИЙ (ВИРТУАЛЬНЫЙ) АДРЕС – СОСТОИТ ИЗ СЕЛЕКТОРА (СЕГМЕНТНЫЙ РЕГИСТР)И СМЕЩЕНИЯ. ДЛЯ КАЖДОЙ ЗАДАЧИ В IA-32 МОЖНО ОПРЕДЕЛИТЬ 214 ЗНАЧЕНИЙ СЕЛЕКТОРА

* ПРИ 32-Х РАЗРЯДНОМ СМЕЩЕНИИ ОБЩИЙ ОБЪЕМ АДРЕСУЕМОЙ ПАМЯТИ СОСТАВИТ 246 Байт = 64 ТБайт (ПОТЕНЦИАЛЬНЫЙ РАЗМЕР ВИРТУАЛЬНОЙ ПАМЯТИ)

* ЭФФЕКТИВНЫЙ АДРЕС (ЕА) – ПРЕДСТАВЛЕНИЕ СМЕЩЕНИЯ В ВИДЕ РЕЗУЛЬТАТА СУММИРОВАНИЯ СОДЕРЖИМОГО АДРЕСНЫХ ПОЛЕЙ В ФОРМАТЕ КОМАНДЫ (БАЗА, ИНДЕКС, СМЕЩЕНИЕ)
* ЛИНЕЙНЫЙ АДРЕС – ФОРМИРУЕТСЯ БЛОКОМ СЕГМЕНТАЦИИ MMU И СОВПАДАЕТ С ФИЗИЧЕСКИМ АДРЕСОМ ЕСЛИ ВЫКЛЮЧЕН РЕЖИМ СТРАНИЧНОЙ АДРЕСАЦИИ
* ФИЗИЧЕСКИЙ АДРЕС – ОПРЕДЕЛЯЕТ НЕПОСРЕДСТВЕННЫЕ ЗНАЧЕНИЯ НА АДРЕСНЫХ ВХОДАХ ПАМЯТИ

**ФИЗИЧЕСКИЙ АДРЕС В РЕАЛЬНОМ РЕЖИМЕ**

****

**ОСОБЕННОCТИ СЕГМЕНТАЦИИ ПАМЯТИ В РЕАЛЬНОМ РЕЖИМЕ**

* В ПАМЯТИ ОРГАНИЗОВАНЫ НЕСКОЛЬКО НЕЗАВИСИМЫХ АДРЕСНЫХ ПРОСТРАНСТВ С АППАРАТНОЙ ПОДДЕРЖКОЙ ОБРАЩЕНИЯ К НИМ
* ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СТАНОВИТСЯ ПЕРЕНОСИМЫМ
* НЕДОСТАТКИ МЕХАНИЗМА СЕГМЕНТАЦИИ ПАМЯТИ В РЕАЛЬНОМ РЕЖИМЕ:

1. ОГРАНИЧЕННЫЙ ОБЪЕМ ПАМЯТИ 1 МБ
2. ОГРАНИЧЕННЫЙ РАЗМЕР СЕГМЕНТА 64К – НЕТ ВОЗМОЖНОСТИ РАБОТЫ С БОЛЬШИМИ МАССИВАМИ
3. ВЫПОЛНЯЕТСЯ ТОЛЬКО ОДНА ЗАДАЧА
4. БЕСКОНТРОЛЬНОЕ РАЗМЕЩЕНИЕ СЕГМЕНТОВ В ПАМЯТИ
5. ОТСУТСТВУЕТ МЕХАНИЗМ ЗАЩИТЫ ПАМЯТИ

**СЕГМЕНТАЦИЯ В ЗАЩИЩЕННОМ РЕЖИМЕ**

* ДЛЯ СЕГМЕНТОВ ОПРЕДЕЛЯЮТСЯ БАЗОВЫЙ АДРЕС, РАЗМЕР И АТРИБУТЫ
* В СЕГМЕНТНЫХ РЕГИСТРАХ КОДА, ДАННЫХ И СТЕКА СОДЕРЖАТСЯ СЕЛЕКТОРЫ ДЛЯ ИЗВЛЕЧЕНИЯ ИЗ ПАМЯТИ ДЕСКРИПТОРОВ СЕГМЕНТОВ
* ЛЮБОЙ СЕГМЕНТ ПРОГРАММНО НЕДОСТУПЕН, ПОКА ЕГО ДЕСКРИПТОР НЕ ЗАПИСАН В ТАБЛИЦУ ДЕСКРИПТОРОВ

**ОСОБЕННОСТИ ФОРМИРОВАНИЯ ФИЗИЧЕСКОГО АДРЕСА В ЗАЩИЩЕННОМ РЕЖИМЕ**

* АДРЕСУЕМАЯ ПАМЯТЬ – 4 ГБайта
* СЕГМЕНТЫ ПЕРЕМЕННОЙ ДЛИНЫ ОТ 1 байта ДО 4 Гбайт
* ДОСТУПНА СТРАНИЧНАЯ АДРЕСАЦИЯ
* СЕЛЕКТОР УКАЗЫВАЕТ НА ДЕСКРИПТОР СЕГМЕНТА В ТАБЛИЦАХ (GDT, LDT)
* ТАБЛИЦЫ (GDT, LDT) СОЗДАЮТСЯ ОПЕРАЦИОННОЙ СИСТЕМОЙ
* АНАЛОГИЯ С КОСВЕННОЙ АДРЕСАЦИЕЙ (БАЗОВЫЙ АДРЕС В ПАМЯТИ)

**ТАБЛИЦЫ ДЕСКРИПТОРОВ И СЕЛЕКТОРЫ**

* ДЕСКРИПТОРЫ СЕГМЕНТОВ МОГУТ НАХОДИТЬСЯ В ГЛОБАЛЬНОЙ ТАБЛИЦЕ ДЕСКРИПТОРОВ (GDT) ИЛИ ЛОКАЛЬНОЙ ТАБЛИЦЕ (LDT)

* В ЕДИНСТВЕННОЙ GDT СОДЕРЖАТСЯ ДЕСКРИПТОРЫ СЕГМЕНТОВ, ДОСТУПНЫХ ВСЕМ ЗАДАЧАМ (СЕГМЕНТЫ КОДА И ДАННЫХ ОС, СЕГМЕНТЫ СОСТОЯНИЯ ЗАДАЧ, ДЕСКРИПТОРЫ LDT)
* LDT СОЗДАЮТСЯ ДЛЯ КАЖДОЙ ВЫПОЛНЯЕМОЙ СИСТЕМОЙ ЗАДАЧИ И СОДЕРЖАТ ДЕСКРИПТОРЫ СЕГМЕНТОВ, ДОСТУПНЫХ ТОЛЬКО ОТДЕЛЬНЫМ ЗАДАЧАМ
* ДЛЯ НАХОЖДЕНИЯ ДЕСКРИПТОРА В ТАБЛИЦАХ GDT И LDT ИСПОЛЬЗУЕТСЯ 16-ТИ РАЗРЯДНЫЙ СЕЛЕКТОР, КОТОРЫЙ ОПРЕДЕЛЯЕТСЯ СОДЕРЖИМЫМ СЕГМЕНТНОГО РЕГИСТРА ИЛИ LDTR

**РАСПОЛОЖЕНИЕ ТАБЛИЦ ДЕСКРИПТОРОВ В ПАМЯТИ**

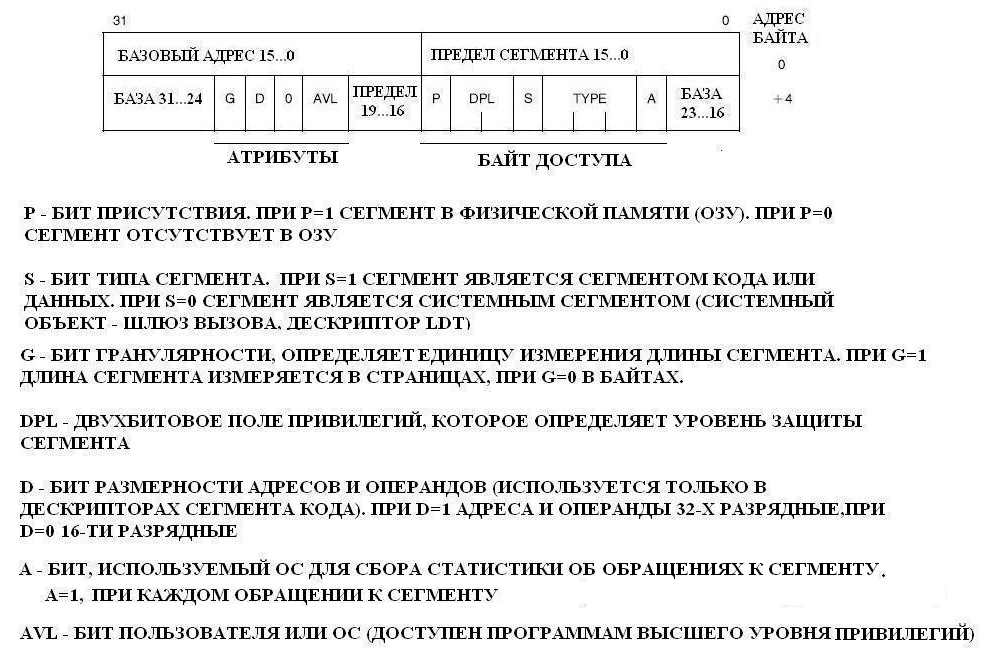
* АДРЕСА И РАЗМЕРЫ GDT И LDT ЗАДАЮТСЯ ПРИ ПОМОЩИ РЕГИСТРОВ GDTR (48 РАЗРЯДОВ) И LDTR (16 РАЗРЯДОВ)

* GDTR: 32 РАЗРЯДА - БАЗОВЫЙ АДРЕСGDT, 16 РАЗРЯДОВ - РАЗМЕР GDT В БАЙТАХ
* LDTR: 16-ТИ РАЗРЯДНЫЙ СЕЛЕКТОР - УКАЗЫВАЕТ НА РАСПОЛОЖЕННЫЙ В GDT ДЕСКРИПТОР LDT (РАЗМЕР LDT УКАЗАН В ДЕСКРИПТОРЕ)

**РАЗМЕЩЕНИЕ ДЕСКРИПТОРОВ РАЗЛИЧНЫХ ТИПОВ**

* ДЕСКРИПТОРЫ СЕГМЕНТОВ КОДОВ ПРОГРАММ - GDT/LDT
* ДЕСКРИПТОРЫ СЕГМЕНТОВ ДАННЫХ - GDT/LDT
* ДЕСКРИПТОРЫ СЕГМЕНТОВ СТЕКА - GDT/LDT
* ДЕСКРИПТОРЫ СЕГМЕНТОВ TSS - GDT
* ДЕСКРИПТОРЫ LDT - GDT
* ШЛЮЗЫ ВЫЗОВА - GDT/LDT
* ШЛЮЗЫ ЗАДАЧ - GDT/LDT

**ДЕСКРИПТОР СЕГМЕНТА**

****

**ИСПОЛЬЗОВАНИЕ ФЛАГОВ ДЕСКРИПТОРА**

* ПРИ ПОМОЩИ ФЛАГА «Р» ОБНАРУЖИВАЮТСЯ ОТСУТСТВУЮЩИЕ В ПАМЯТИ СЕГМЕНТЫ, ИСПОЛЬЗУЕМЫЕ ПРИ ВЫПОЛНЕНИИ КОМАНДЫ

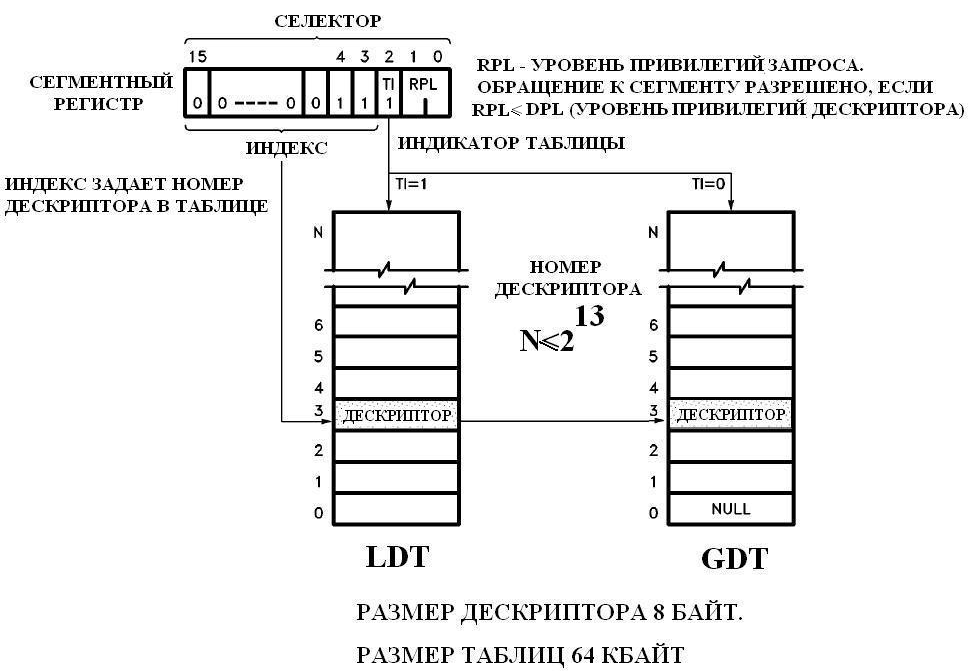
* ПРИ Р=0 – СЕГМЕНТА В ПАМЯТИ НЕТ, ВОЗНИКАЕТ ОСОБЫЙ СЛУЧАЙ (ПРОГРАММНОЕ ПРЕРЫВАНИЕ)

* НУЖНЫЙ СЕГМЕНТ С ДИСКА ЗАПИСЫВАЕТСЯ В ПАМЯТЬ (ОС ОПРЕДЕЛЯЕТ БАЗОВЫЙ АДРЕС СЕГМЕНТА, ПОМЕЩАЕТ ЕГО В ДЕСКРИПТОР СЕГМЕНТА И УСТАНАВЛИВАЕТ Р=1)
* ПОСЛЕ ЭТОГО РЕСТАРТ КОМАНДЫ, ЗАПРОСИВШЕЙ ОТСУТСТВУЮЩИЙ В ПАМЯТИ СЕГМЕНТ
* БИТ «А» (ОБРАЩЕНИЯ) ПОЗВОЛЯЕТ ВЫЯВИТЬ СЕГМЕНТЫ, КОТОРЫЕ РЕДКО ИСПОЛЬЗУЮТСЯ ПРИ ВЫПОЛНЕНИИ ПРОГРАММЫ
* ЭТО НЕОБХОДИМО ДЛЯ ИХ ЗАМЕНЫ НА ВОСТРЕБОВАННЫЕ В ТЕКУЩИЙ МОМЕНТ СЕГМЕНТЫ
* СПОСОБ – ПЕРИОДИЧЕСКИЙ СБРОС БИТА «А» В ДЕСКРИПТОРАХ ВСЕХ СЕГМЕНТОВ

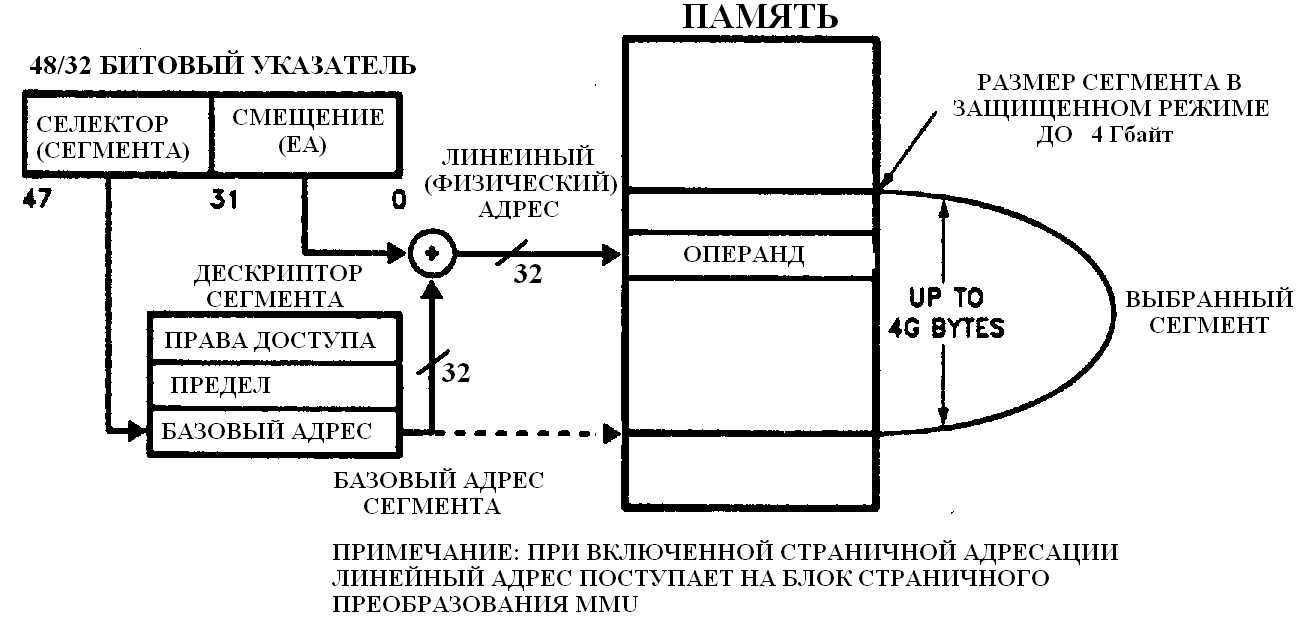
**ТИПЫ ДЕСКРИПТОРОВ**

|  |  |
| --- | --- |
| **TYPE** | **НАЗНАЧЕНИЕ СЕГМЕНТА** |
| 000 | СЕГМЕНТ ДАННЫХ, ТОЛЬКО ЧТЕНИЕ |
| 001 | СЕГМЕНТ ДАННЫХ, ЧТЕНИЕ И ЗАПИСЬ |
| 010 | ЗАРЕЗЕРВИРОВАНО |
| 011 | СЕГМЕНТ СТЕКА, ЧТЕНИЕ И ЗАПИСЬ |
| 100 | СЕГМЕНТ КОДА, ТОЛЬКО ВЫПОЛНЕНИЕ |
| 101 | СЕГМЕНТ КОДА, ВЫПОЛНЕНИЕ И ЧТЕНИЕ |
| 110 | ПОДЧИНЕННЫЙ СЕГМЕНТ КОДА, ТОЛЬКО ВЫПОЛНЕНИЕ |
| 111 | ПОДЧИНЕННЫЙ СЕГМЕНТ КОДА, ВЫПОЛНЕНИЕ И ЧТЕНИЕ |

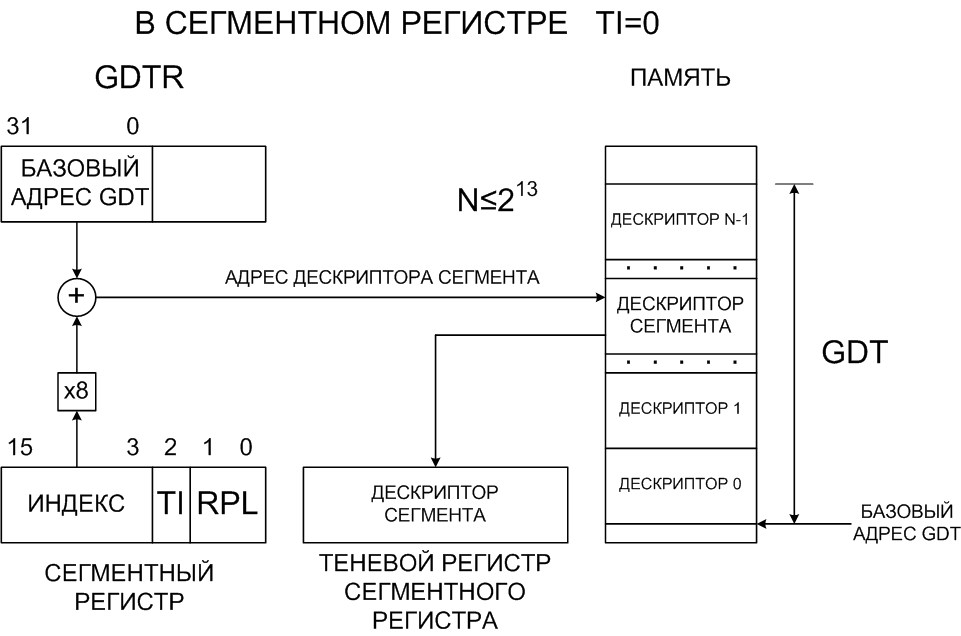
**ФОРМАТ СЕЛЕКТОРА**

****

**ФИЗИЧЕСКИЙ АДРЕС В ЗАЩИЩЕННОМ РЕЖИМЕ (БЕЗ СТРАНИЧНОЙ АДРЕСАЦИИ)**

****

**АВТОМАТИЧЕСКАЯ ЗАГРУЗКА ТЕНЕВОГО РЕГИСТРА СЕГМЕНТА ИЗ GDT**

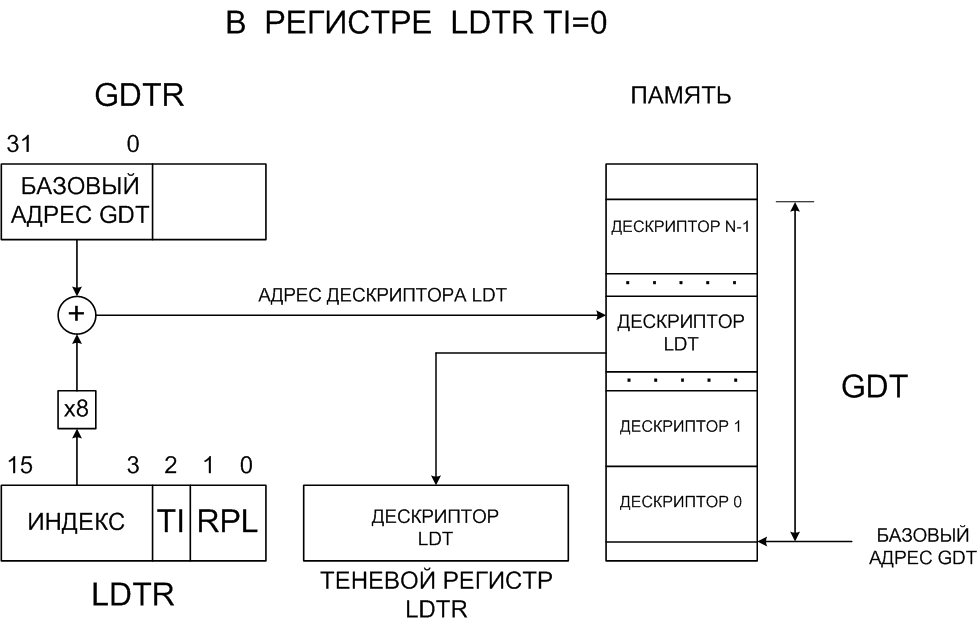
****

**ИЗВЛЕЧЕНИЕ ДЕСКРИПТОРА СЕГМЕНТА ИЗ GDT**

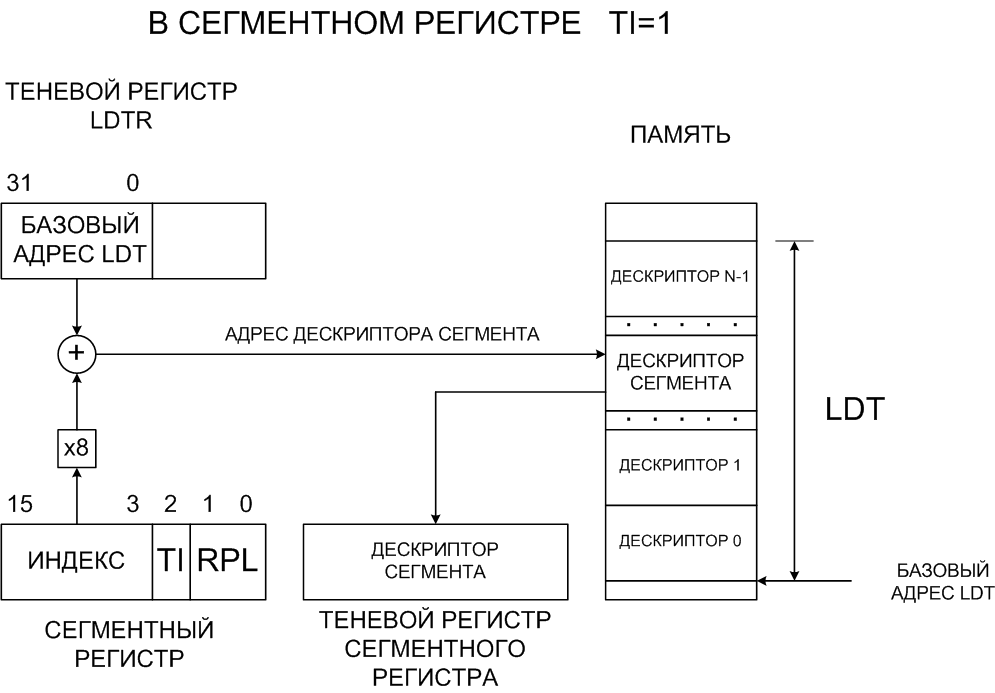
* ЭТА ПРОЦЕДУРА ВЫПОЛНЯЕТСЯ СИСТЕМОЙ АВТОМАТИЧЕСКИ ПРИ ЗАНЕСЕНИИ В СЕГМЕНТНЫЙ РЕГИСТР НОВОГО СЕЛЕКТОРА
* ДЕСКРИПТОР КЭШИРУЕТСЯ В ТЕНЕВОЙ РЕГИСТР СЕГМЕНТНОГО РЕГИСТРА
* ЭТО ПОЗВОЛЯЕТ ПРИ ФОРМИРОВАНИИ ЛИНЕЙНОГО АДРЕСА ДЛЯ ОБРАЩЕНИЯ К СЕГМЕНТУ БАЗОВЫЙ АДРЕС СЕГМЕНТА БРАТЬ НЕ ИЗ МЕДЛЕННОЙ ПАМЯТИ, А ИЗ БЫСТРОГО ТЕНЕВОГО РЕГИСТРА
* ЛИНЕЙНЫЙ АДРЕС ДЕСКРИПТОРА В ПАМЯТИ ПОЛУЧАЕТСЯ СЛОЖЕНИЕМ 32-Х РАЗРЯДНОГО БАЗОВОГО АДРЕСА GDT, КОТОРЫЙ БЕРЕТСЯ ИЗ GDTR, И 16-ТИ РАЗРЯДНОГО СМЕЩЕНИЯ - СДВИНУТОГО ВЛЕВО НА 3 РАЗРЯДА 13-ТИ РАЗРЯДНОГО ИНДЕКСА (УМНОЖЕНИЕ НА 8 НЕОБХОДИМО, ТАК КАК КАЖДЫЙ ДЕСКРИПТОР В ТАБЛИЦЕ ЗАНИМАЕТ 8 БАЙТ)

* ПЕРЕД ВЫПОЛНЕНИЕМ СЛОЖЕНИЯ СМЕЩЕНИЕ СРАВНИВАЕТСЯ С 16-ТИ РАЗРЯДНЫМ ПРЕДЕЛОМ (ГРАНИЦЕЙ GDT), КОТОРЫЙ БЕРЕТСЯ ИЗ GDTR
* ДЕСКРИПТОР С НУЛЕВЫМ АДРЕСОМ В GDT НЕ ИСПОЛЬЗУЕТСЯ

**АВТОМАТИЧЕСКАЯ ЗАГРУЗКА ТЕНЕВОГО РЕГИСТРА LDTR ИЗ LDT**

****

**АВТОМАТИЧЕСКАЯ ЗАГРУЗКА ТЕНЕВОГО РЕГИСТРА СЕГМЕНТА ИЗ LDT**

****

**ОСОБЕННОСТИ ИЗВЛЕЧЕНИЯ ДЕСКРИПТОРА СЕГМЕНТА ИЗ LDT**

* ДЛЯ УСКОРЕНИЯ ВЫБОРКИ ОПЕРАНДА ИЗ ПАМЯТИ РЕГИСТР LDTR ИМЕЕТ ТЕНЕВОЙ РЕГИСТР ДЛЯ КЭШИРОВАНИЯ ДЕСКРИПТОРА LDT ТЕКУЩЕЙ ЗАДАЧИ

* КЭШИРОВАНИЕ ВЫПОЛНЯЕТСЯ ПРОЦЕССОРОМ АВТОМАТИЧЕСКИ ПРИ ЗАГРУЗКЕ В LDTR НОВОГО СЕЛЕКТОРА LDT
* ИСПОЛЬЗОВАНИЕ СЕЛЕКТОРА LDTR ДЛЯ ДОСТУПА К LDT ДЕЛАЕТ ИХ ИДЕНТИЧНЫМИ СЕГМЕНТАМ
* УПРОЩЕНИЕ АППАРАТНОЙ РЕАЛИЗАЦИИ МЕХАНИЗМОВ ФОРМИРОВАНИЯ ФИЗИЧЕСКОГО АДРЕСА

**ТИПЫ КОМАНД ПО РАСПОЛОЖЕНИЮ ОПЕРАНДОВ В IA-32**

* РЕГИСТР – РЕГИСТР
* РЕГИСТР – ПАМЯТЬ
* НЕПОСРЕДСТВЕННЫЙ ОПЕРАНД – РЕГИСТР
* НЕПОСРЕДСТВЕННЫЙ ОПЕРАНД – ПАМЯТЬ

**ФОРМИРОВАНИЕ СМЕЩЕНИЯ В СЕГМЕНТЕ И ФОРМАТЫ КОМАНД В IA-32**

* СМЕЩЕНИЕ В СЕГМЕНТЕ КОДА ОПРЕДЕЛЯЕТСЯ СОДЕРЖИМЫМ РЕГИСТРА EIP

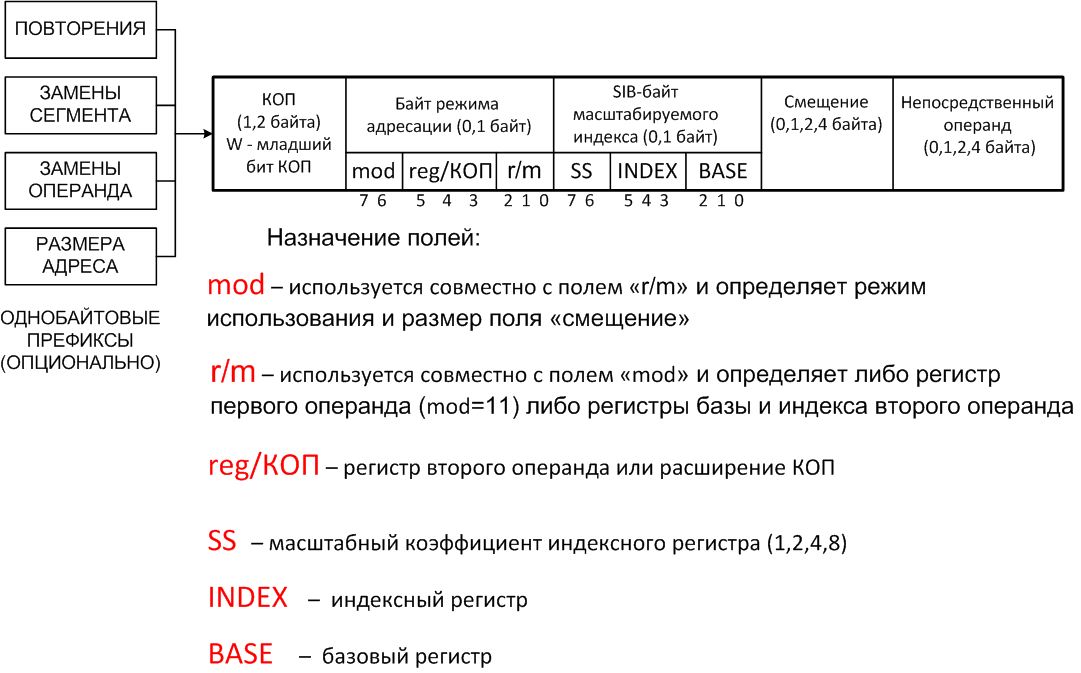
* СМЕЩЕНИЕ В СЕГМЕНТЕ ДАННЫХ (ЭФФЕКТИВНЫЙ АДРЕС - ЕА)

ОПРЕДЕЛЯЕТСЯ ИСПОЛЬЗУЕМЫМ СПОСОБОМ АДРЕСАЦИИ ОПЕРАНДА. АДРЕСНАЯ ИНФОРМАЦИЯ СОДЕРЖИТСЯ В КОДЕ КОМАНДЫ

* В IA-32 ИМЕЕТСЯ БОЛЬШОЕ КОЛИЧЕСТВО ФОРМАТОВ КОМАНД РАЗЛИЧНОЙ ДЛИНЫ, КОТОРЫЕ РАЗЛИЧАЮТСЯ ПО ТИПАМ ОПЕРАЦИЙ, И СПОСОБАМ АДРЕСАЦИИ ОПЕРАНДОВ

* ДЛЯ СИСТЕМАТИЗАЦИИ ИНФОРМАЦИИ ОБ АДРЕСНОЙ ЧАСТИ КОМАНДЫ В IA-32 ПРИНЯТО РАССМАТРИВАТЬ ОБОБЩЕННЫЙ ФОРМАТ ДВУХАДРЕСНОЙ КОМАНДЫ

**ОБОБЩЕННЫЙ ФОРМАТ КОМАНДЫ В IA-32**

****

**ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ НЕКОТОРЫХ ПОЛЕЙ ФОРМАТА КОМАНДЫ**

* Смещение в команде представлено в дополнительном коде со знаком
* Префиксы изменяют размеры адреса и операндов (16↔32 , 32↔ 16 ), установленные битом «D» в дескрипторе СЕГМЕНТА (В СОЧЕТАНИИ С БИТОМ «W» ПОЛЯ «КОП»)
* W – бит размерности операндов, позволяет определить операнд размером 8 бит

* Сочетания наличия префикса и бита «W» определяют размеры адресов и операндов (СМ. таблицУ СЛАЙД 26)

**ТАБЛИЦА ПЕРЕОПРЕДЕЛЕНИЯ РАЗРЯДНОСТИ АДРЕСА И ОПЕРАНДОВ**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Бит размерности D  в дескрипторе сегмента** | **0** | **0** | **0** | **0** | **1** | **1** | **1** | **1** |
| Префикс размерности операнда | **-** | **-** | **+** | **+** | **-** | **-** | **+** | **+** |
| Префикс размерности адреса | **-** | **+** | **-** | **+** | **-** | **+** | **-** | **+** |
| Разрядность операнда для w=1/0 (бит) | 16/8 | 16/8 | 32/8 | 32/8 | 32/8 | 32/8 | 16/8 | 16/8 |
| Разрядность адреса (бит) | 16 | 32 | 16 | 32 | 32 | 16 | 32 | 16 |

**КОНЦЕПЦИЯ АДРЕСАЦИИ ОПЕРАНДОВ В МП**

* Набор способов адресации в IA-32 ориентирован на высокоуровневое программирование
* Поддерживаются 11 способов адресации для выборки операндов
* Первый способ предполагает расположение операнда в РОН (8,16,32 бита)
* Второй способ предполагает непосредственное задание операнда в коде команды
* Остальные 9 способов предназначены для извлечения операнда из памяти с использованием 32-х разрядного эффективного адреса (ЕА)

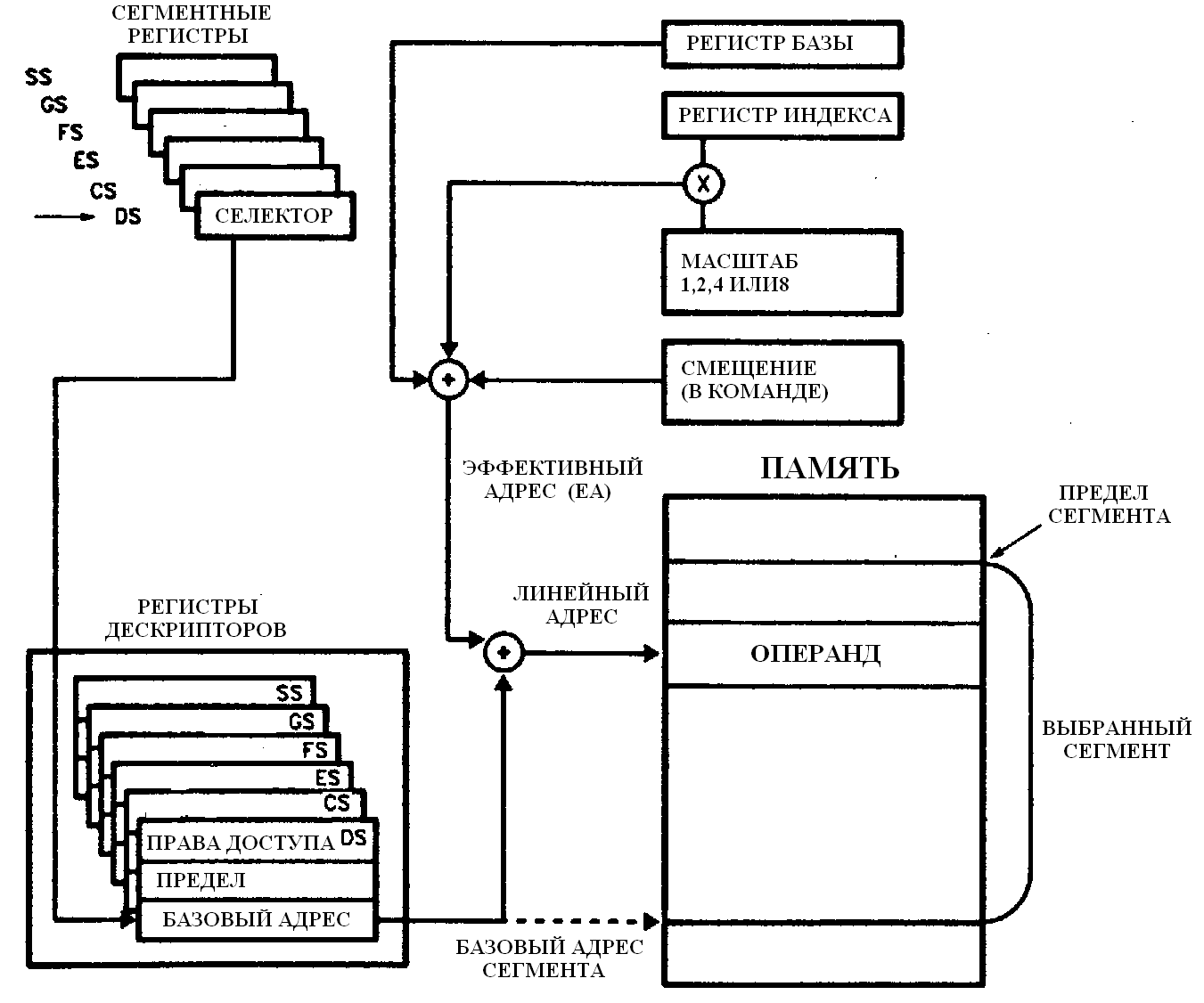
**ВЫЧИСЛЕНИЕ ЭФФЕКТИВНОГО АДРЕСА**

* ЕА = Регистр базы + Регистр индекса \*Масштаб + Смещение
* На уровне компиляции регистр базы определяет начало области памяти локальной переменной
* Регистр индекса используется для обращения к элементам массива
* Масштабирование позволяет повысить удобство для работы с массивами элементов различной длины
* В качестве регистров базы и индекса могут использоваться любые РОН

**СПОСОБЫ АДРЕСАЦИИ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **СПОСОБ АДРЕСАЦИИ** | **Используемая адресная информация** | | | | **Пример** |
| **Регистр базы** | **Регистр индекса** | **Смещение в команде** | **Масштаб** |
| **Прямая (Direct)** | **-** | **-** | **+** | **-** | **INC Word PTR[500]** |
| **Косвенная регистровая (Register Inderect)** | **+** | **-** | **-** | **-** | **MOV [ECX], EDX** |
| **Базовая (Based)** | **+** | **-** | **+** | **-** | **MOV ECX, [EAX+24]** |
| **Индексная (Index)** | **-** | **+** | **+** | **-** | **ADD EAX, TABLE[ESI]** |
| **Индексная с масштабиро-ванием**  **(Scaled Index)** | **-** | **+** | **+** | **+** | **IMUL EBX, TABLE[ESI\*4],7** |
| **Базовая индексная (Based Index)** | **+** | **+** | **-** | **-** | **MOV EAX, [ESI] [EBX]** |
| **Базовая индексная с масштабиро-ванием (Based Scaled Index)** | **+** | **+** | **-** | **+** | **MOV ECX, [EDX\*8] [EAX]** |
| **Относительная**  **Базовая индексная (Based Index with Displacement)** | **+** | **+** | **-** | **+** | **ADD EDX,[ESI] [EBP+00FFFFF0H]** |
| **Относительная**  **Базовая индексная (Based Scaled Index with Displacement)** | **+** | **+** | **+** | **+** | **MOV EAX, LOCALTABLE[EDI\*4] [EBP+80]** |

**СХЕМА ФОРМИРОВАНИЯ ЭФФЕКТИВНОГО АДРЕСА**

****

**ВИРТУАЛИЗАЦИЯ ПАМЯТИ**

* ВИРТУАЛИЗАЦИЯ ПАМЯТИ - ИСПОЛЬЗОВАНИЕ ПАМЯТИ СУЩЕСТВЕННО БОЛЬШЕГО ОБЪЕМА, ЧЕМ ОЗУ – АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ ИЕРАРХИЧЕСКОЙ ПАМЯТЬЮ
* ЕМКОСТЬ ОЗУ НЕ ПОКРЫВАЕТ ВСЕ ВИРТУАЛЬНОЕ АДРЕСНОЕ ПРОСТРАНСТВО, ЧАСТЬ ДАННЫХ НА ДИСКЕ
* НЕОБХОДИМЫЕ ДАННЫЕ ПОДГРУЖАЮТСЯ С ДИСКА СРЕДСТВАМИ ОПЕРАЦИОННОЙ СИСТЕМЫ (ОБРАЩЕНИЕ К КАРТЕ ДИСКА, В КОТОРОЙ УКАЗАНЫ ДОРОЖКА И СЕКТОР ДИСКА ДЛЯ КАЖДОЙ ВИРТУАЛЬНОЙ СТРАНИЦЫ)
* ИСПОЛЬЗУЕТСЯ ТОЛЬКО В ЗАЩИЩЕННОМ РЕЖИМЕ

**СТРАНИЧНАЯ ОРГАНИЗАЦИЯ ПАМЯТИ**

* СТРАНИЧНАЯ МОДЕЛЬ ВИРТУАЛЬНОЙ ПАМЯТИ - ВИРТУАЛЬНОЕ АДРЕСНОЕ ПРОСТРАНСТВО РАЗБИВАЕТСЯ НА БЛОКИ ФИКСИРОВАННОЙ ДЛИНЫ – СТРАНИЦЫ
* СТРАНИЦА ИМЕЕТ НОМЕР ДЛЯ ИДЕНТИФИКАЦИИ, ЛОКАЛИЗАЦИИ И ПРИВЯЗКИ К КОНКРЕТНОЙ ПРОГРАММЕ
* ПРОГРАММА РАЗРАБАТЫВАЕТСЯ В ВИРТУАЛЬНЫХ АДРЕСАХ. ПРИ ВЫПОЛНЕНИИ ТРЕБУЕТСЯ ИХ ПРЕОБРАЗОВАНИЕ В ФИЗИЧЕСКИЕ АДРЕСА
* СООТВЕТСТВИЕ ВИРТУАЛЬНЫХ И ФИЗИЧЕСКИХ СТРАНИЦ ОПРЕДЕЛЯЕТСЯ ТАБЛИЦАМИ СТРАНИЧНОГО ПРЕОБРАЗОВАНИЯ (ИХ КОНФИГУРАЦИЯ ДОЛЖНА МИНИМИЗИРОВАТЬ ЗАНИМАЕМЫЙ ОБЪЕМ ПАМЯТИ)

**СТРАНИЧНАЯ АДРЕСАЦИЯ В IA-32**

* РАЗМЕР СТРАНИЦЫ – 4 Кбайт (РАЗМЕР СЕКТОРА ДИСКА)
* СТРАНИЧНОЕ ПРЕОБРАЗОВАНИЕ ВКЛЮЧАЕТСЯ УСТАНОВКОЙ БИТА «PG»=1 В УПРАВЛЯЮЩЕМ РЕГИСТРЕ «CR0»
* ДЛЯ СТРАНИЧНОГО ПРЕОБРАЗОВАНИЯ АДРЕСА (АДРЕС СТРАНИЦЫ И СМЕЩЕНИЕ) ИСПОЛЬЗУЕТСЯ 32-Х РАЗРЯДНЫЙ ЛИНЕЙНЫЙ АДРЕС, ПОЛУЧЕННЫЙ ПОСЛЕ СЕГМЕНТНОГО ПРЕОБРАЗОВАНИЯ ЛОГИЧЕСКОГО АДРЕСА
* ДВУХСТУПЕНЧАТАЯ СХЕМА СТРАНИЧНОГО ПРЕОБРАЗОВАНИЯ АДРЕСА (ЭКОНОМИЯ ПАМЯТИ ПРИ ХРАНЕНИИ ТАБЛИЦ СТРАНИЧНОГО ПРЕОБРАЗОВАНИЯ)

**ДВУХСТУПЕНЧАТОЕ СТРАНИЧНОЕ ПРЕОБРАЗОВАНИЕ АДРЕСА**

* ИСПОЛЬЗУЮТСЯ СПЕЦИАЛЬНЫЕ ТАБЛИЦЫ ДВУХ ТИПОВ – КОРНЕВАЯ И ПОДЧИНЕННЫЕ, КАЖДАЯ РАЗМЕРОМ 4 Кбайта (РОВНО ОДНА СТРАНИЦА)
* КОРНЕВАЯ ТАБЛИЦА – КАТАЛОГ (ДИРЕКТОРИЯ) ТАБЛИЦ СТРАНИЦ (КТС) СОДЕРЖИТ 1024 32-Х РАЗРЯДНЫХ ЭЛЕМЕНТА КТС (ЭКТС – PDE page directory entry)
* ЭКТС СОДЕРЖИТ БАЗОВЫЙ АДРЕС СВОЕЙ ПОДЧИНЕННОЙ ТАБЛИЦЫ СТРАНИЦ (ТС) И ЕЕ АТРИБУТЫ
* КАЖДАЯ ТС СОДЕРЖИТ 1024 32-Х РАЗРЯДНЫХ ЭЛЕМЕНТА ТС (ЭТС – PТE page table entry)
* ЭТС СОДЕРЖИТ БАЗОВЫЙ (ФИЗИЧЕСКИЙ) АДРЕС СТРАНИЦЫ И ЕЕ АТРИБУТЫ

**ФОРМАТЫ PDE И PTE**

**Р» - БИТ ПРИСУТСТВИЯ. ПРИ Р=1 СТРАНИЦА НАХОДИТСЯ В ОЗУ.**

**ПРИ Р=0 ОБРАЩЕНИЕ К СТРАНИЦЕ ВЫЗЫВАЕТ ПРЕРЫВАНИЕ, ОБРАБОТЧИК КОТОРОГО ПОДГРУЖАЕТ НУЖНУЮ СТРАНИЦУ**

**«А» – БИТ ДОСТУПА. УСТАНАВЛИВАЕТСЯ В СОСТОЯНИЕ А=1 ПРИ**

**ВЫПОЛНЕНИИ ОБРАЩЕНИЯ К СТРАНИЦЕ ДЛЯ ЗАПИСИ ИЛИ ЧТЕНИЯ.**

**ИСПОЛЬЗУЕТСЯ ДЛЯ ЗАМЕЩЕНИЯ СТРАНИЦ В ОЗУ (ПО АНАЛОГИИ ЗАМЕЩЕНИЯ СТРОК В КЭШ-ПАМЯТИ)**

**НАЗНАЧЕНИЕ БИТОВ «PCD» И «PWT»**

* «PCD» - БИТ ЗАПРЕТА КЭШИРОВАНИЯ СТРАНИЦЫ. ПРИ PCD=1 КЭШИРОВАНИЕ ЗАПРЕЩЕНО
* «PWT» – ОПРЕДЕЛЯЕТ СПОСОБ ЗАПИСИ ИНФОРМАЦИИ ВО ВНЕШНЮЮ КЭШ- ПАМЯТЬ 2-ГО УРОВНЯ

ПРИ PWT=1 СКВОЗНАЯ ЗАПИСЬ,

ПРИ PWT=0 ОБРАТНАЯ ЗАПИСЬ

* ДАННЫЕ БИТЫ ПРИ ЧТЕНИИ «PDE» БЕРУТСЯ ИЗ РЕГИСТРА CR3
* ДАННЫЕ БИТЫ ПРИ ЧТЕНИИ «PТE» БЕРУТСЯ ИЗ «PDE»

**ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ «СВОПИНГА»**

* «D» – БИТ МОДИФИКАЦИИ (ИСПОЛЬЗУЕТСЯ ТОЛЬКО В «РТЕ»)
* ПРИ ЗАГРУЗКЕ СТРАНИЦЫ В ОЗУ «ОС» УСТАНАВЛИВАЕТ ЗНАЧЕНИЕ «D=0»

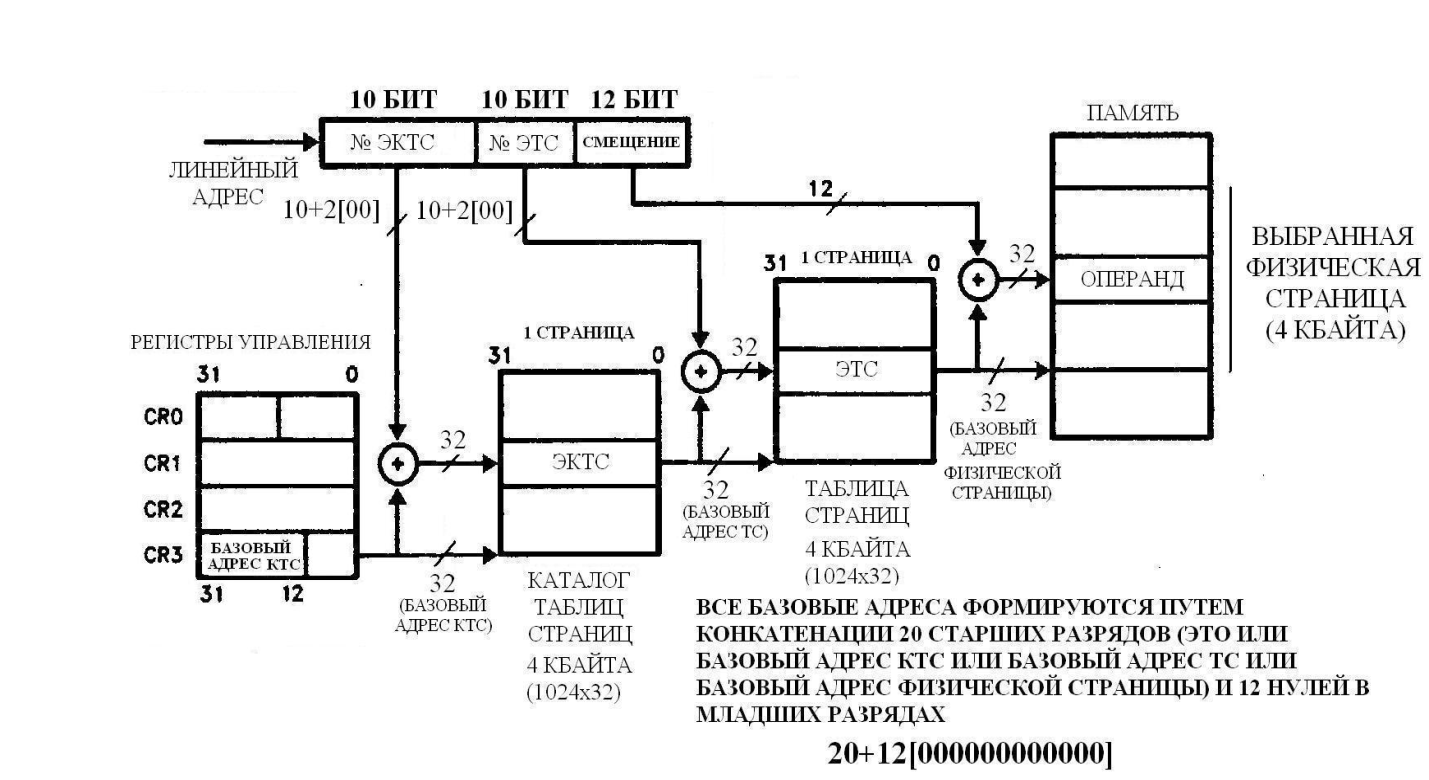
* ПРИ ЗАПИСИ НА СТРАНИЦУ УСТАНАВЛИВАЕТСЯ ЗНАЧЕНИЕ «D=1»
* ПЕРЕД ВЫГРУЗКОЙ СТРАНИЦЫ НА ДИСК ОС АНАЛИЗИРУЕТ БИТ «D»

* ЕСЛИ «D=0», ТО ЭТО ОЗНАЧАЕТ, ЧТО ЗАПИСИ НА СТРАНИЦУ НЕ БЫЛО И ПЕРЕДАЧА НА ДИСК СТРАНИЦЫ НЕ ТРЕБУЕТСЯ

**БИТЫ УПРАВЛЕНИЯ ПРАВАМИ ДОСТУПА К СТРАНИЦАМ**

|  |  |  |  |
| --- | --- | --- | --- |
| Биты R/W и U/S определяют права доступа к ТС или странице для отдельных программ с различными уровнями привилегий. Для страниц специфицированы только два уровня привилегий: **супервизор** (U/S=0, для сегмента значения DPL= 0,1,2) и **пользователь** (U/S=1, для сегмента значение DPL= 3). Допустимые операции при обращении к странице для разных уровней привилегий программ приведены в таблице | | | |
| U/S | R/W | Допустимо  для DPL=3  (пользователь) | Допустимо  для DPL= 0,1,2  (супервизор) |
| 0 | Х | Запрет | Чтение/запись |
| 1 | 0 | Чтение | Чтение/запись |
| 1 | 1 | Чтение/запись | Чтение/запись |

**СХЕМА ДВУХСТУПЕНЧАТОГО ПРЕОБРАЗОВАНИЯ АДРЕСА**

****

**КЭШИРОВАНИЕ ФИЗИЧЕСКОГО АДРЕСА СТРАНИЦЫ**

* ПРИ СТРАНИЧНОЙ АДРЕСАЦИИ С ДВУХУРОВНЕВЫМ ПРЕОБРАЗОВАНИЕМ АДРЕСА НЕОБХОДИМЫ ДВА ОБРАЩЕНИЯ К ПАМЯТИ ДЛЯ СЧИТЫВАНИЯ ЭКТС И ЭТС, ЧТО СНИЖАЕТ ПРОИЗВОДИТЕЛЬНОСТЬ ПРОЦЕССОРА
* НА ПРАКТИКЕ ПРОЦЕССОР НЕ ОЧЕНЬ ЧАСТО МЕНЯЕТ СТРАНИЦЫ, С КОТОРЫМИ РАБОТАЕТ ПРОГРАММА
* ИМЕЕТ СМЫСЛ СОХРАНЯТЬ ФИЗИЧЕСКИЕ АДРЕСА СТРАНИЦ В БЫСТРОЙ БУФЕРНОЙ ПАМЯТИ (КЭШИРОВАНИЕ ФИЗИЧЕСКИХ АДРЕСОВ СТРАНИЦ)

**БУФЕР АССОЦИАТИВНОЙ ТРАНСЛЯЦИИ АДРЕСА СТРАНИЦЫ (TLB) В IA-32**

* ДЛЯ КЭШИРОВАНИЯ АДРЕСА СТРАНИЦЫ В ПРОЦЕССОРЕ ИМЕЕТСЯ БЛОК АССОЦИАТИВНОЙ ТРАНСЛЯЦИИ АДРЕСА СТРАНИЦЫ (TLB)
* «TLB» ПРЕДНАЗНАЧЕН ДЛЯ ХРАНЕНИЯ 32 ФИЗИЧЕСКИХ АДРЕСОВ СТРАНИЦ (ОБЩИЙ ОБЪЕМ АДРЕСУЕМОЙ ПАМЯТИ 4х32=128 КБ)

* ПРЕДСТАВЛЯЕТ СОБОЙ 4-Х ВХОДОВУЮ МНОЖЕСТВЕННО-АССОЦИАТИВНУЮ ПАМЯТЬ (8 МОДУЛЕЙ / 4 СТРОКИ / 20 БИТ)

**СТРУКТУРА TLB В IA-32**

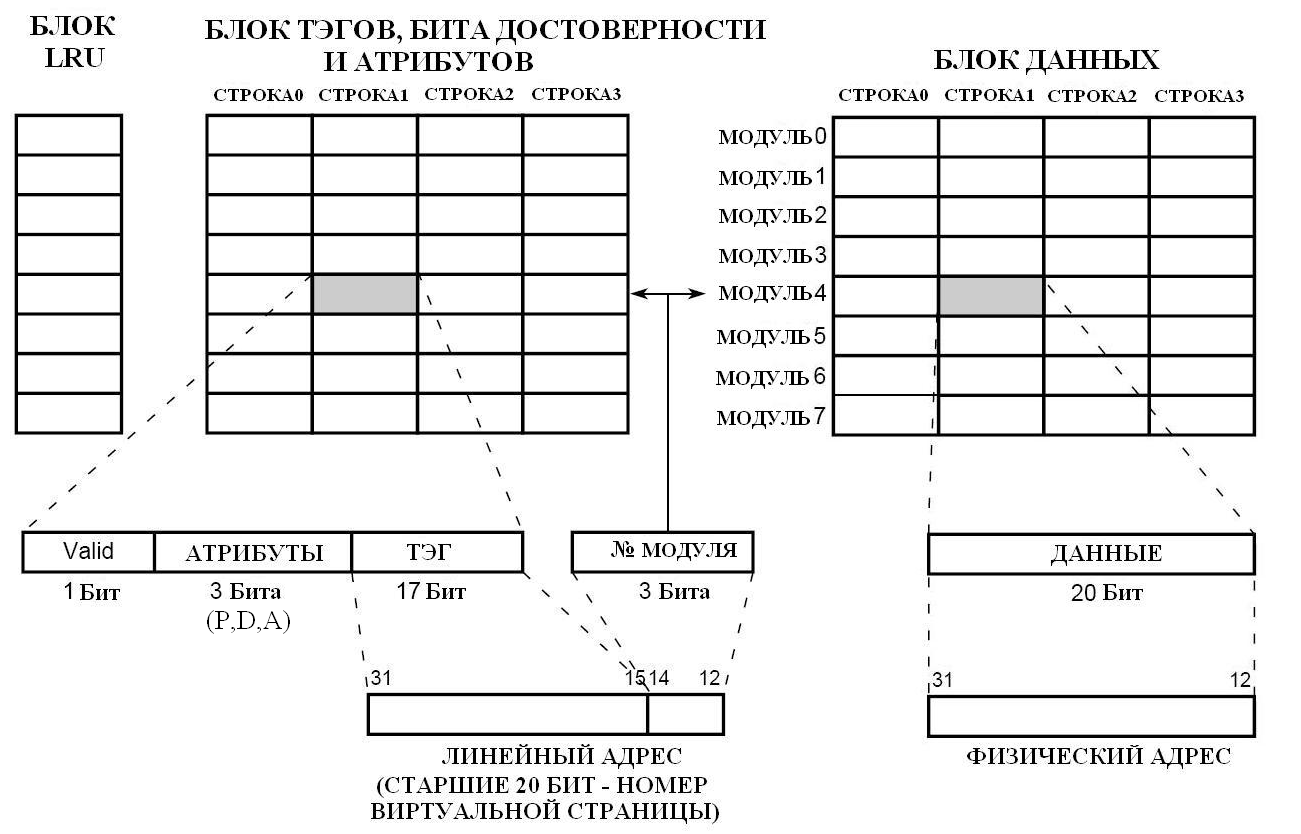
TLB СОДЕРЖИТ:

* БЛОК ДАННЫХ
* БЛОК LRU
* БЛОК ТЭГОВ, БИТА ДОСТОВЕРНОСТИ И АТРИБУТОВ

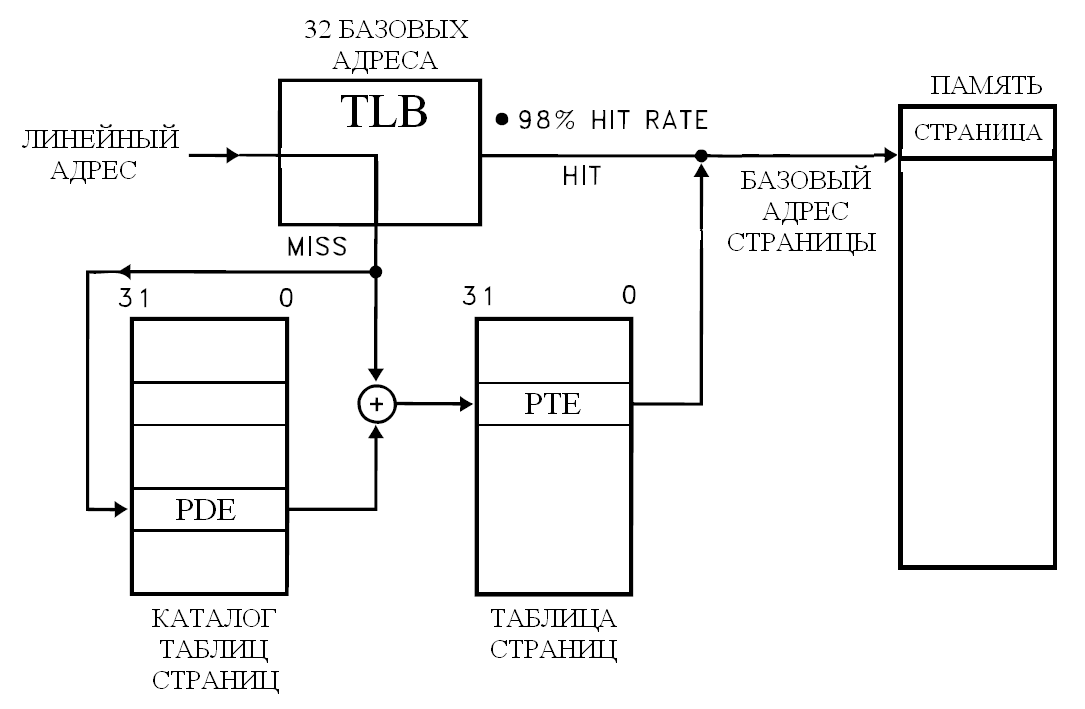
**ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ TLB**

* БЛОК MMU КЭШИРУЕТ ВСЕ ИСПОЛЬЗУЕМЫЕ «PTE» ДО ЗАПОЛНЕНИЯ TLB
* МОДУЛЬ БЛОКА ДАННЫХ , В КОТОРЫЙ ЗАПИСЫВАЕТСЯ ФИЗИЧЕСКИЙ АДРЕС СТРАНИЦЫ, ОПРЕДЕЛЯЕТСЯ ТРЕМЯ МЛАДШИМИ РАЗРЯДАМИ ЛИНЕЙНОГО АДРЕСА. СТРОКА В МОДУЛЕ ВЫБИРАЕТСЯ ПРОИЗВОЛЬНО
* В КАЧЕСТВЕ ТЭГа КАЖДОГО ХРАНИМОГО В TLB ФИЗИЧЕСКОГО АДРЕСА ИСПОЛЬЗУЮТСЯ 17 СТАРШИХ РАЗРЯДОВ ЛИНЕЙНОГО АДРЕСА
* «V (Valid)» - БИТ ДОСТОВЕРНОСТИ ИНФОРМАЦИИ В СТРОКЕ TLB. ПРИ V=0 СТРОКА НЕ ЗАПОЛНЕНА, ПРИ V=1 – ДОСТОВЕРНАЯ ИНФОРМАЦИЯ
* ИСПОЛЬЗУЕТСЯ МЕХАНИЗМ ЗАМЕЩЕНИЯ НАИБОЛЕЕ ДОЛГО НЕИСПОЛЬЗУЕМОЙ СТРОКИ PLRU (pseudo-least recently used). ВЫЯВЛЕНИЕ ТАКОЙ СТРОКИ ВЫПОЛНЯЕТСЯ ПРОВЕРКОЙ ИНФОРМАЦИИ В БЛОКЕ «LRU» (НА КАЖДЫЙ МОДУЛЬ 3 БИТА)

**ОРГАНИЗАЦИЯ TLB В IA-32**

****

**ЭФФЕКТИВНОСТЬ TLB**

ДОЛЯ КЭШ-ПОПАДАНИЙ СОСТАВЛЯЕТ 98%.

ТОЛЬКО ДЛЯ 2% ПРЕОБРАЗОВАНИЙ СТРАНИЧНОГО АДРЕСА

ТРЕБУЕТСЯ ОБРАЩЕНИЕ К КАТАЛОГУ ТАБЛИЦ СТРАНИЦ И ТАБЛИЦЕ СТРАНИЦ