**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №11**

1. **Какие процедуры выполняются в общем случае при обращении к кэш-памяти, если А П Тег для полностью ассоциативного распределения?**

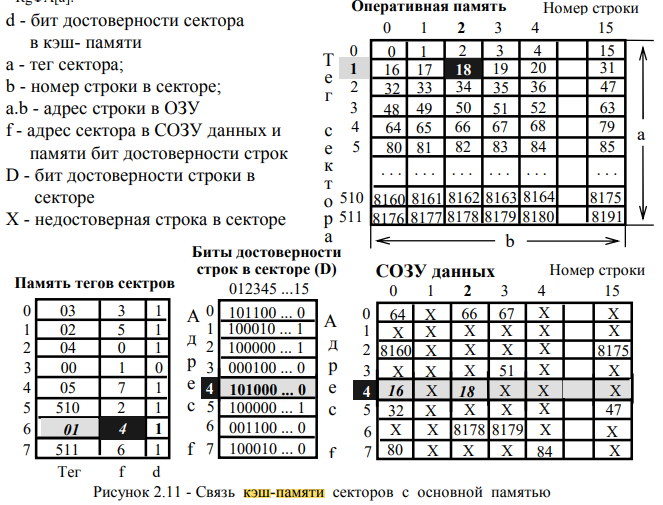
Для хранения идентификатора строк, находящихся в кэш-памяти, используют специальную память, называемую теговой памятью, а для хранения непосредственно данных (строк) используется СОЗУ данных. То есть каждому адресу строки в теговой памяти соответствует 8 слов (строка) в СОЗУ данных. Таким образом, для определения местоположения строки (в кэш-памяти или в 14 ОП) необходимо сравнить содержимое всех ячеек теговой памяти с 13-ю старшими разрядами ФА (тегом ФА) и если будет обнаружено совпадение, то значит строка располагается в быстродействующей кэш-памяти и операнд считывается из или записывается в СОЗУ данных кэш-памяти. Для получения адреса строки СОЗУ данных [b] в памяти тегов необходимо выполнить его формирование (адрес указан в скобках) или в дополнительном поле теговой памяти хранить этот адрес.



1. **Назначение бита достоверности сектора и строки кэш-памяти. Как используется значение этих бит при обращении к памяти, если А П Тег и А Î Тег для секторного распределения кэш-памяти?**

Бит достоверности данных d=1 указывает на принадлежность строки кэш-памяти, а нулевое значение означает, что данная ячейка свободна (в ней размещаются недостоверные данные, принадлежащие, например, другой программе).

При этом методе сектора в кэш-памяти распределены ассоциативно. Каждой строке, хранимой в СОЗУ данных, соответствует свой бит достоверности строки, который показывает, совпадает или нет содержимое данной строки с содержимым строки в ОП (рисунок 2.11). Выполняется проверка бит достоверности секторов d в памяти тегов, и если все di=1, то параллельно производится ассоциативный поиск сектора в памяти тегов по 9-и старшим разрядам RgФА[a]



**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №12**

1. **Приведите алгоритм свопинга страниц, если бит присутствия страницы Р=0.**

При отсутствии страницы в ОП (бит присутствия Р=0), и если в ОП нет свободного места, то она замещает одну из страниц, находящихся в ОП. При этом, если в замещаемую страницу во время ее пребывания в ОП производилась запись, она должна быть передана в дисковую память.

Эти процедуры передачи информации, называемые **свопингом страниц**, вызывают простои процессора, поэтому следует стремиться уменьшить число таких операций во время выполнения программы.

**Алгоритм по времени неиспользования**. Из ОП удаляется страница, наиболее давно неиспользовавшаяся, Для каждой страницы необходимо вычислять значения Т1, Т2, Т3, ... , Тm, характеризующие времена неиспользования страниц, размещенных в ОП. Для этого каждой странице ставится в соответствие бит обращения А (бит неиспользования), который устанавливается при каждом обращении к странице. Для наблюдения отводятся интервалы времени, в течение которых процессор выполняет R команд и вырабатывается прерывание для начисления времени неиспользования. Если бит обращения А=0 за это время (не было обращений к странице), то время неиспользования увеличивается на единицу Тн:=Тн + 1, записываемое в определенное поле дескриптора страницы. Удалению подлежит страница с максимальным временем неиспользования Тн.

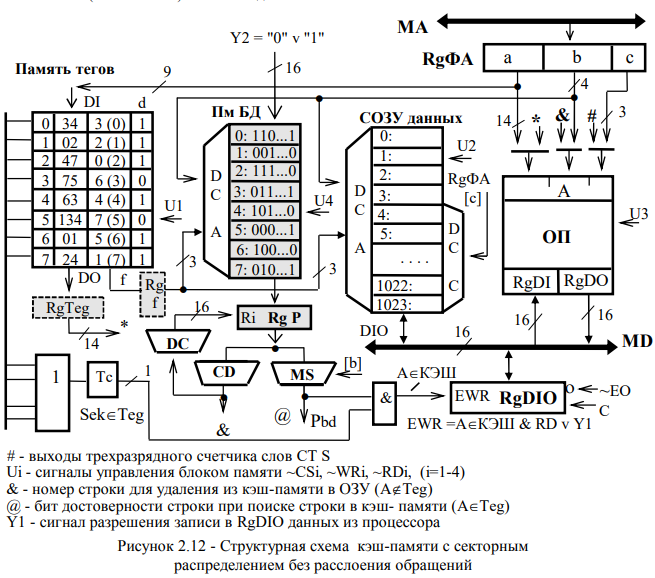
1. **Секторное распределение кэш-памяти. Определение. Выбор формата ФА. Назначение полей ФА (на что указывает поле и характеризует)**

Является развитием метода полностью ассоциативного распределения, основным недостатком которого было использование дорогостоящего АЗУ, информационная емкость и стоимость которого возрастает с увеличением емкости СОЗУ данных.

По методу распределения секторов вся ОП разбивается на секторы, каждый из которых состоит из фиксированного числа строк. То же делается с СОЗУ данных. Пусть сектор состоит из 16 строк, тогда старшие 9 бит адреса указывают на номер сектора, следующие 4 бита - адрес строки в секторе, а младшие 3 бита - адрес слова в строке.

Таким образом, при этом методе сектора в кэш-памяти распределены ассоциативно. Каждой строке, хранимой в СОЗУ данных, соответствует свой бит достоверности строки, который показывает, совпадает или нет содержимое данной строки с содержимым строки в ОП (рисунок 2.11 **выше**).

На рисунке 2.12 приведена структурная схема кэш-памяти с секторным распределением. При выполнении обращения к памяти по чтению или записи сначала выполняется ассоциативный поиск на совпадение тега сектора, хранимого в АЗУ, с тегом поля [a] RgФА. Если Sek∈Teg, то по адресу [f], сформированному или выбранному из памяти тегов выполняется обращение к памяти бит достоверности строк. В регистре RgP фиксируется 16-разрядный код, описывающий состояние строк сектора, а на выходе MS появляется признак бита достоверности запрашиваемой строки. Параллельно при чтении из СОЗУ данных считывается запрашиваемое слово по адресу [f.b.c], и если признак достоверности строки равен "1", то считанное слово защелкивается в RgDIO.



**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №13**

**1. Причины, приводящие к возобновлению работы конвейера команд, начиная с первой ступени. Методы повышения быстродействия процессора с конвейерным принципом выполнения командного цикла процессора при нарушении естественного порядка следования команд.**

Ко этой группе причин относятся: Изменение последовательности выполнения команд для команд условного и безусловного перехода, вызова подпрограмм и т.д. Возникновение прерываний как аппаратных (внешних и внутренних), так и программных.

Поэтому необходимо различать два подхода для устранения влияния перечисленных ситуаций: методы для обеспечения непрерывной работы конвейера, которые приводят к увеличению времени простоя процессора на каждой ступени конвейеризации, и методы, нарушающие последовательность выполнения команд, находящихся в буфере очереди команд и уже принятых к обработке на первых ступенях конвейера.

**Причины**: для команд условного перехода, если условие не выполняется, то порядок выполнения команд не нарушается и конвейер продолжает работу, а при выполнении условия, а также по командам безусловного перехода, CALL и при возникновении прерываний команды, введенные в конвейер и в очередь команд, считаются недействительными (сброс указателей стека очереди команд) и осуществляется перезапуск алгоритма со стадии выборки команды. При этом основной проблемой является блокировка выполнения команд, следующих за командой перехода после стадии декодирования этой команды и вычисления адреса перехода, путем перетрансляции потока микрокоманд в NOP либо после формирования адреса перехода блокировки записи результата в ОП и РОН на стадии S (операции чтения являются безобидными).

**Методы:**

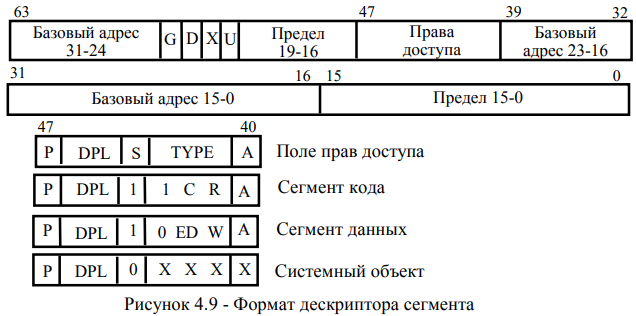
Для устранения причин второй группы существует ряд способов:

♦ с использованием буфера цикла команд достаточно большой емкости, в котором хранится весь программный цикл (подпрограмма), или нескольких таких буферов. Тогда при выполнении условия (вызове подпрограммы или прерывании) доступ осуществляется к буферу цикла, а не к ОП;

♦ множественного потока команд, при котором процессор имеет два буфера команд: основной и вспомогательный. Основной буфер используется для хранения очереди команд, если условие не задано (естественная адресация), а вспомогательный для очереди команд, если условие выполняется или нарушается естественный порядок следования команд. В ряде моделей ЭВМ используются два и более вспомогательных буфера, т.е. для каждого условия или цикла свой буфер;

♦ использование таблицы с предисторией переходов. Метод основан на использовании ассоциативной памяти, в которой в ходе выполнения программы запоминаются наиболее часто используемые переходы в командах условного перехода, циклов, вызова подпрограмм и т.д., например, по стратегии псевдо LRU-стека. Тогда при повторном выполнении некоторого участка программы осуществляется быстрый доступ к АЗУ и выполняется предвыборка команд и помещение их в очередь команд

**2. Какой формат имеет дескриптор сегмента ЦП Intel? Назначение полей.**



Формат дескриптора показан на рисунке 4.9.

Базовый адрес – определяет любой 32-разрядный начальный адрес сегмента или таблицы LDT в линейном адресном пространстве 4 Гбайт.

Предел - 20-разрядное поле, определяющее размер сегмента или таблицы LDT в байтах или в страницах минус 1 и определяет границу сегмента относительно базового адреса.

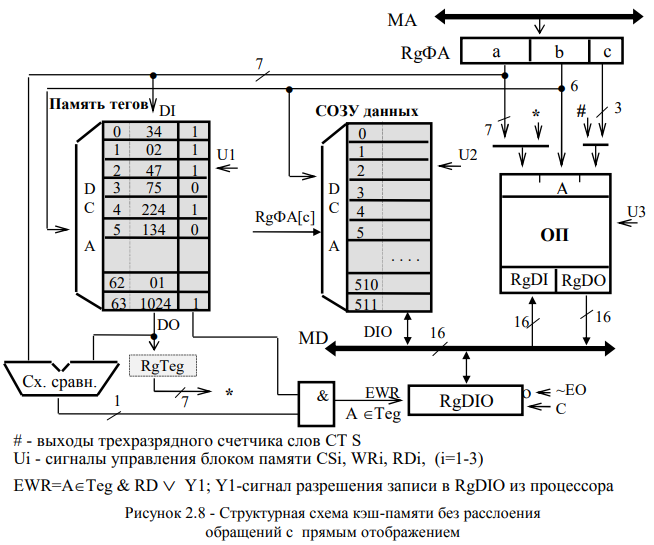
**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №14**

**1. Прямое распределение кэш-памяти. Определение. Выбор формата ФА. Назначение полей ФА (на что указывает каждое поле и что характеризует).**

С целью уменьшения аппаратурных затрат для представления адреса строки (тега) разобьем **формат ФА** на 3 части: 3-разрядное поле [c] указывает на адрес слова в строке, разряды [а.b] - на адрес строки в ОП. Тогда структуру ОП можно представить в виде матрицы [а x b] (рисунок 2.7)

Тогда 6-разрядное поле [b] (индекс) указывает на множество строк (128), отстоящих друг от друга с шагом кратным 64 (например, для RgФА[b=4] это строки с номерами 4, 68, 132, 196,..., 8132), а 7- разрядное поле [а=2] можно использовать в качестве тега для выбора одной строки из этого ряда (строка 132). Такое разбиение позволяет сократить разрядность тега с 13 до 7 разрядов. Таким образом, поиск строки осуществляется в два этапа: поле [b] напрямую указывает (адресует) группу строк, а по тегу (старшим разрядам адреса) определяется принадлежность запрашиваемой строки кэш-памяти.

Кэш-память с прямым распределением строится на основе двух блоков СОЗУ с произвольным доступом на основе обычных БИС ЗУ статического типа с произвольным доступом (рисунок 2.8): первый блок выполняет функции теговой памяти для хранения старших разрядов номера строки ОП, находящейся в данный момент времени в СОЗУ данных кэш-памяти. Таким образом, емкость теговой памяти составляет 64 7-разрядных слов, а СОЗУ данных - 64 строки по 8 n-разрядных слов.



**2. Формирование кода Хэмминга. Математический аппарат для получения вектора ошибки.**

Код Хэмминга получают путем добавления к информационным разрядам слова дополнительных контрольных разрядов, которые формируются перед записью (передачей) информации путем подсчета четности суммы единиц для определенных групп информационных разрядов.

Для образования контрольного кода Хэмминга (КХ) используется проверочная матрица Н размерности r x n, где r - число контрольных бит, а n = k + r, где k - разрядность информационного слова, а n - длина кодового слова, записываемого в память

Введем следующие обозначения:

V = (V1,V2,V3,...,Vn) - записанное в ОЗУ кодовое слово;

U = (U1,U2,U3,...,Un) - считанное из ОЗУ кодовое слово.

При записи в ОЗУ и при считывании выполняется процесс формирования контрольных бит с помощью проверочной матрицы



где Р - двоичная матрица размерности r x k, а Ir - единичная матрица размерности r x r. Первые k бит кодового слова представляют собой информационные биты, r последних - контрольные.

Проверочная матрица Н размерностью r x n определяет условия, которым должно удовлетворять считанное из ОЗУ кодовое слово. Эти условия задаются в виде бит, сумма по модулю 2 которых должна быть равна нулю (синдрома ошибок), т.е. должно выполняться условие:



где Uт - трансформированный вектор U.

Код, описываемый матрицей вида (5.1), является разделимым.

Если записанное и считанное кодовые слова отличаются хотя бы в одном бите, то разница между U и V определяет вектор ошибки Е=(Е1,Е2,....,Еn), т.е. U = V + E, т.е. если Еi ≠ 0, то i-й бит считанного слова содержит ошибку.

**Вектор ошибки:**

Проверка считанного слова заключается в вычислении синдрома ошибки S:



Если S - нулевой вектор, то считается, что ошибок нет. В противном случае синдром определяет вектор ошибки. Синдром является суммой по модулю 2 тех столбцов h матрицы Н, которым соответствуют ошибки. Если столбец hi нулевой, то ошибка в этой позиции кодового слова не окажет влияния на синдром и не позволит обнаружить ошибку. Если два столбца матрицы Н совпадают, то одиночные ошибки в каждой из этих позиций кодового слова дают один и тот же синдром.

Отсюда, матрица Н кода КО-ОД должна удовлетворять следующим условиям:

1) векторы-столбцы матрицы Н не должны быть нулевыми и должны отличаться друг от друга;

2) сумма двух векторов-столбцов матрицы Н не должна равняться нулю или любому третьему вектору-столбцу матрицы Н.

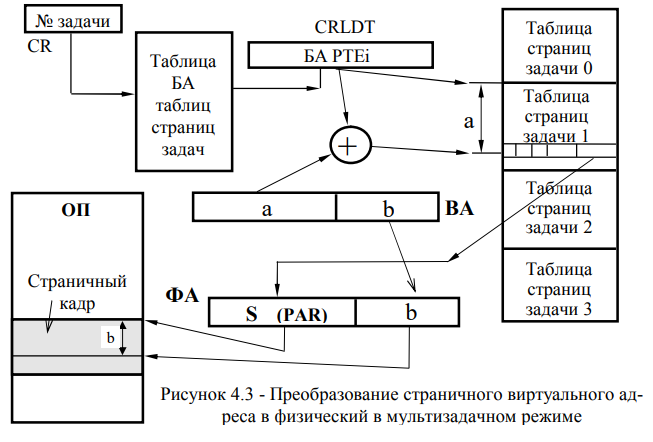
**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №15**

1. **Страничная организация виртуальной памяти. Организация доступа к страницам в мультипрограммном режиме работы через глобальную таблицу дескрипторов. Структура УУП. Алгоритм преобразования ВА в ФА через регистр настроек. Достоинства и недостатки метода.**

Внешняя дисковая память и программа разбиваются на части равной величины, называемые страницами. В начальный момент времени все страницы, представляющие программу, хранятся на дисковой памяти и по мере необходимости затребованные страницы помещаются в ОП. Размер страницы обычно выбирают в пределах 2-4 Кбайт. Таким образом, виртуальное пространство является линейным, разбитым на страницы одинакового размера, что исключает фрагментацию ОП при дозагрузке страниц из дисковой памяти в ОП. Если при запросе доступа к странице она отсутствует в ОП, то страница загружается из дисковой памяти в ОП. Такая ситуация называется страничным сбоем. Если в ОП нет свободной области памяти для дозагрузки страниц, то необходимо предварительно освободить область ОП путем удаления одной из страниц из ОП на дисковую память. Удаляемая страница определяется алгоритмом замещения страниц.

В **мультипрограммной** страничной виртуальной памяти каждой задаче отводится свое линейное виртуальное пространство. Отличие от однозадачной виртуальной памяти заключается в том, что для каждой задачи отводится своя локальная таблица страниц. В структуру УУП вводится дополнительный регистр-указатель, в который при переключении задач из быстродействующей памяти загружается базовый (начальный) адрес таблицы страниц для данной задачи (рисунок 4.3).

Данный регистр выполняет функции регистра CR в однозадачной виртуальной памяти. Дальнейший процесс преобразования ВА в ФА аналогичен предыдущему алгоритму.

**Преобразование ВА в ФА**

Основными **недостатками** данного метода являются:   
♦ низкое быстродействие, так как для доступа к операнду необходимо два обращения к памяти: к таблице страниц за дескриптором и к ОП за операндом;   
♦ таблица страниц может занимать до нескольких Мбайт памяти, так как число страниц достигает до 1 М и более дескрипторов, а каждый дескриптор требует до 4 байт памяти;   
♦ в мультипрограммном режиме работы возникает проблема перераспределения номеров страниц между задачами

**2. Достоинства и недостатки кэш-памяти с прямым распределением.**

**Достоинства**:

1. Простота реализации на основе двух СОЗУ с произвольным доступом.

2. Достаточно высокое быстродействие, т.к. блоки памяти тегов и СОЗУ данных работают параллельно при чтении.

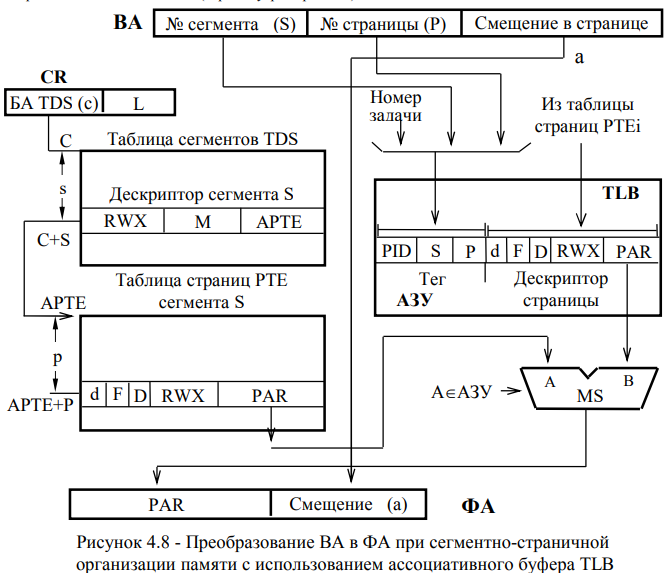
3. Память тегов и данных может быть достаточно большой емкости, т.к. кроме обычных БИС памяти необходима одна схема сравнения.

**Недостато**к: большая вероятность промахов при обращении к кэш-памяти или вероятность частого замещения строк кэш-памяти, т.к. место хранения строки в кэш-памяти однозначно определяется номером индекса строки (группы строк) (например, при частом обращении к строкам кратным 64: 63, 127, 191, 255,..., 8191 каждый раз требуется процедура обновления ОП и замещения кэш-памяти).

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №16**

**1. Что представляет собой сегментно-страничная организация памяти (схема)? Алгоритм работы**

Простейший вариант сегментно-страничной организации памяти можно представить в виде следующей структуры: программа, как и при сегментной организации памяти, состоит из сегментов кода, данных и стека переменной длины, а те в свою очередь делятся на страницы фиксированной длины



**Алгоритм**

Первоначально по полям номера задачи, сегмента и страницы осуществляется ассоциативный поиск дескриптора страницы в АЗУ. Если А∈АЗУ, то выполняется контроль по атрибутам защиты страницы, и если доступ к странице разрешен, то формируется физический адрес операцией конкатенации поля PAR (базового адреса страничного кадра) из дескриптора страницы АЗУ и смещения из ВА и устанавливаются необходимые биты в дескрипторе страницы. В регистре CR хранится начальный (базовый адрес) таблицы сегментов и ее размер. По номеру сегмента S из таблицы сегментов выбирается его дескриптор и выполняется контроль по правам доступа к сегменту и размеру, и если доступ не запрещен, то в поле АРТЕ дескриптора сегмента указан базовый номер таблицы страниц данной задачи, а по полю Р виртуального адреса из таблицы страниц выбирается ее дескриптор и также проверяются права доступа к странице и определяется ее местонахождение по биту присутствия F.

Если страница находится в ОП, то формируется физический адрес слова операцией конкатенации поля PAR дескриптора страницы и смещения [a], а дескриптор страницы и тег виртуального адреса загружаются в АЗУ на место назначенного на удаление дескриптора. Если запрашиваемая страница находится на диске (F=0), то выполняется процедура свопинга страницы по алгоритму, описанному выше.

**2. Достоинства и недостатки кэш-памяти с секторным распределением.**

Данный метод позволяет существенно снизить емкость памяти тегов на основе АЗУ при той же емкости СОЗУ данных по сравнению с полностью ассоциативным распределением либо увеличить емкость СОЗУ данных при неизменной емкости памяти тегов (для рассматриваемого примера емкость АЗУ уменьшена в 8 раз, а емкость СОЗУ данных увеличена в два раза). Остальные достоинства и недостатки метода аналогичны полностью ассоциативному распределению.

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №17**

**1. В чем заключаются недостатки использования сегментной организации виртуальной памяти?**

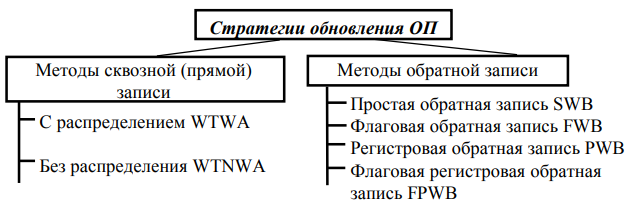
**Недостатки:**

1. Проблема настраиваемости, т.е. перераспределения номеров сегментов между задачами, так как сегменты в таблице GDT имеют сквозную нумерацию от 0 до k для всех задач, а внутри задачи адресация выполняется также с нулевого сегмента (см. страничную организацию памяти), т.е. необходимо формировать базовые номера сегментов задач при распределении ресурсов между задачами, а перед переключением задач выполнить перезагрузку регистра настроек задач.

2. В мультипрограммном режиме работы любая задача имеет доступ к любому сегменту таблицы GDT, даже к тем, которые могут использоваться только одной конкретной задачей. Это означает, что невозможно организовать эффективную защиту сегментов задач от случайного или преднамеренного доступа со стороны других пользователей.

3. После выполнения нескольких замещений сегментов с диска в ОП возникает проблема фрагментации, так как сегменты могут иметь различный размер, что приводит к неэффективному использованию адресного пространства ОП

**2. Какие стратегии обновления ОП используются при построении кэш-памяти? Классификация.**



**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №18**

**1. Достоинства и недостатки полностью ассоциативного распределения кэш-памяти.**

**Достоинства:**

1. При замещении строк кандидатами на удаление могут выступать любые строки в кэш-памяти, в зависимости от принятой стратегии, а, следовательно, эффективное использование кэш-памяти для хранения наиболее активных строк

**Недостатки:**

1. Высокая стоимость из-за сложности реализации АЗУ.

2. Малая емкость БИС тегов из-за высокой плотности монтажа на кристалле, большая потребляемая мощность и число выводов БИС АЗУ.

3. Для обращения к кэш-памяти требуется два такта и для чтения, и для записи (такт чтения памяти тегов и такт чтения или записи в СОЗУ данных), так как памяти работать параллельно не могут.

4. Высокая стоимость и ограничения по емкости АЗУ (8-64 ячеек) делают этот метод практически неосуществимым.

**2. Техническая реализация и алгоритм работы схемы КО-ОД с последовательной коррекцией для исправления двойных ошибок. Достоинства и недостатки.**

Метод основан на применении кода КО-ОД и позволяет исправлять большинство двойных неисправимых ошибок. Алгоритм работы схемы имеет много общего с первым методом исправления одиночных ошибок, но основан на запоминании синдромов жестких исправимых одиночных ошибок в специальной памяти, построенной на основе ассоциативного запоминающего устройства (АЗУ).

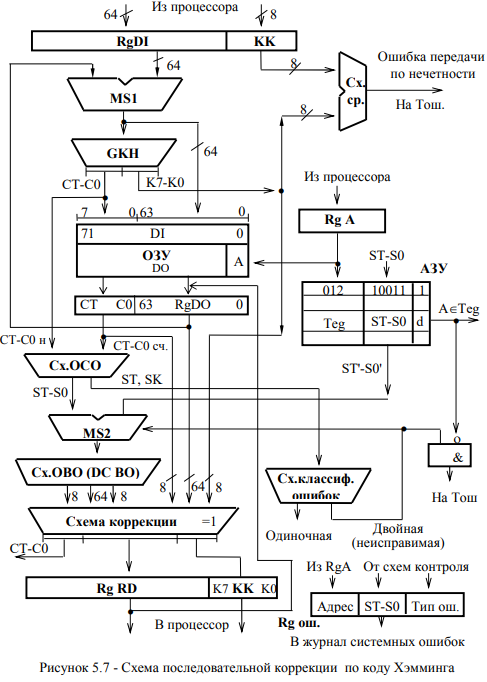
Запись в ОП выполняется аналогично предыдущим методам.

1. При считывании из ОП одновременно осуществляется обращение к АЗУ для формирования признака А∈Teg.

2. Если адрес не принадлежит тегу, то при одиночной ошибке ее синдром ST-S0 и адрес (Теg) записываются в АЗУ, а одиночная ошибка и КК нечетности исправляются средствами кода КО-ОД.

3. Если при считывании выработан сигнал двойной ошибки и A∉Teg, то вырабатывается сигнал прерывания от схем контроля о неисправимой ошибке (двойная ошибка обнаружена в момент возникновения).

4. Если A∈Teg и выработан сигнал одиночной ошибки, то она исправляется схемой коррекции кода КО-ОД, а ее синдром, если он не равен нулю, вновь записывается в АЗУ на место старого синдрома, так как при первой записи синдрома ошибка могла быть вызвана случайным самоустранимым сбоем и в дальнейшем произойдет неправильное исправление ошибки. Если новый синдром равен нулю, то в ячейке АЗУ соответствующий бит достоверности данных сбрасывается. Также выполняется коррекция КК нечетности одного из байт слова.



**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №19**

**1. Что такое привилегированные команды? Какие группы привилегированных команд Вы знаете?**

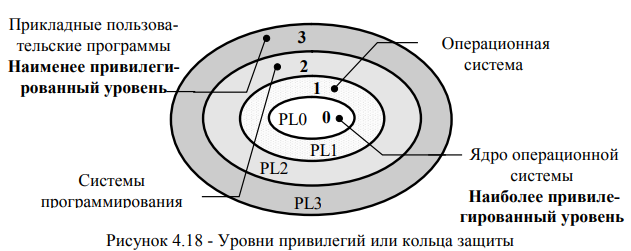
**Привилегированные команды**.

При выполнении почти каждой команды осуществляется проверка защиты по привилегиям.

В систему команд процессора i486 входит 19 привилегированных команд, которые разрешено выполнять только ОС с разным уровнем привилегии.

К первой группе команд относятся команды, изменяющие сегментацию (содержимое сегментных регистров) или влияют на сам механизм защиты (загрузки дескрипторов, селекторов таблиц и т.п.). Эти команды разрешены только на уровне привилегии 0 (PL0-программы). При попытке выполнения этих команд на других уровнях генерируется нарушение общей защиты (прерывание 13) и ОС прекращает работу.

Вторую группу привилегированных команд составляют команды, модифицирующие состояние флажка прерываний IF и команды ввода-вывода IN и OUT. Для этих команд программа не обязательно должна иметь уровень привилегий 0.



**2. Какие процедуры выполняются в общем случае при обращении к кэш-памяти, если найдено совпадение по полю тега для частично-ассоциативного распределения?**

Если хотя бы один тег совпал, то вырабатывается сигнал A∈Teg и на выходе шифратора CD формируется двоичный номер модуля СОЗУ данных, в котором находится запрашиваемая строка. Так как память тегов и СОЗУ данных работают параллельно, то при чтении на выходе MS/DMS появляется запрашиваемое слово строки (из модуля 1, адрес слова определяется 9-ью младшими разрядами регистра адреса [b.c]).

Кандидатом на удаление из кэш-памяти однозначно назначается группа из четырех строк полем [b] RgФА, а конкретная строка определяется ассоциативно по одному из методов замещения строк

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №20**

**1. Какие системные регистры входят в состав устройства сегментного преобразования адресов в ЦП Intel 486?**

**Сегментные** **регистры** CS, SS, DS, ES, FS, GS используются для **сегментной** адресации.

**2. Регистровая обратная и флаговая регистровая обратная запись при обновлении ОП. Сущность метода. Достоинства и недостатки. Оценки метода в сравнении.**

**Регистровая обратная запись PWB**

Является модификацией метода SWB, при котором к магистрали данных между кэш-памятью и ОП подключается регистровый буфер (типа FIFO), который позволяет сократить время простоя кэшпамяти на выполнение процедур обновления ОП и замещения кэш-памяти.

В регистр помещается строка, подлежащая удалению из кэш-памяти, а ее запись в ОП выполняется после обновления кэш-памяти из ОП. Таким образом, обращение к кэш-памяти со стороны процессора возможно сразу после обновления строки в кэш-памяти, а запись удаленной из кэш-памяти строки в ОП будет отложена и выполнена параллельно с работой кэш-памяти. При этом выигрыш во времени составляет от одного обращения к ОП при использовании расслоения обращений к ОП до k х Тзу при использовании ОП без расслоения обращений

**Флаговая регистровая обратная запись FPWB** Является комбинацией методов PWB и FWB и не требует дополнительных комментариев. 

Достоинства и недостатки в подкорках вашего сознания, нужно лишь задуматься