**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №1**

1. 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). Если в памяти тегов хотя бы один бит d=0, то при обнаружении несовпадения тегов при сравнении процедура обновления ОП не выполняется, а затребованная строка из ОП перемещается в кэш-память для дальнейшего использования.
2. Расширение сферы применения средств вычислительной техники (СВТ), особенно их применение для решения научно-технических задач, задач автоматизации производства, САПР, работы с базами данных, обработки статистической информации и управления производством привело к необходимости создания вычислительных систем (ВС), отличающихся составом технических средств (ТС) и техническими характеристиками: производительностью, емкостью оперативной памяти (ОП), внешних запоминающих устройств (ВЗУ) и т.д. Это привело к необходимости создания ЭВМ с единой архитектурой, но с переменным составом оборудования, который определяется выполняемыми ВС функциями. Такой подход означает выполнение отдельных функциональных устройств в виде модулей, которые могут объединяться в необходимом количестве в одной ЭВМ или говорят о ЭВМ проблемно-ориентированных на решение определенного класса задач. При этом существенное место для реализации такого подхода занимает сокращение числа типов (номенклатуры) выпускаемых семейств ЭВМ.

**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №2**

1. Механизмы преобразования адресов Выделяют 4 способа размещения данных в кэш-памяти или механизма преобразования адресов строк: 1. Полностью ассоциативное распределение. 2. Прямое распределение. 3. Частично-ассоциативное распределение. 4. Распределение секторов. Пусть емкость кэш-памяти равна 512 слов, размер строки - 8 слов, а емкость основной памяти 64 Кслова. 2.3.1 Полностью ассоциативное распределение ОП разбивается на строки по 8 слов в каждой (рисунок 2.3). Разрядность физического адреса слова (ФА) составляет 16 бит, следовательно, адрес строки определяется 13-ю старшими битами ФА, которые однозначно идентифицируют любую строку, хранимую в ОП. В кэш-памяти может быть размещено 512:8=64 строки. 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. 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). Если в памяти тегов хотя бы один бит d=0, то при обнаружении несовпадения тегов при сравнении процедура обновления ОП не выполняется, а затребованная строка из ОП перемещается в кэш-память для дальнейшего использования. Кэш-память строится на основе двух блоков памяти: первый выполняет функции теговой памяти для хранения номера строки ОП, находящейся в данный момент времени в СОЗУ данных. Таким образом, емкость теговой памяти составляет 64 13-разрядных слов, а кэш-данных - 64 строки по 8 nразрядных слов или 64 х 8 = 512 слов. Память тегов строится на основе ассоциативного ЗУ (АЗУ), в котором старшие 13 бит ФА (поле а) строки используются в качестве адреса теговой памяти, которые параллельно сравниваются с содержимым всех ячеек памяти тегов. Если хотя бы один тег совпал, то это означает, что строка, содержащая текущий адрес i, находится в СОЗУ данных. В качестве старших разрядов адреса СОЗУ данных выступают 6 разрядов, считываемых из соответствующего поля [b] памяти тегов, к которым присоединяются три младших разряда ФА (поле [c]) и осуществляется обращение к СОЗУ данных по чтению или записи (рисунок 2.4). Если ни один тег не совпадает со значением поля [a] RgФА (А∉Teg), то выполняется процедура обновления строки из кэш-памяти в ОП и замещения кэш-памяти согласно принятой стратегии. Данный метод допускает размещение каждой строки [bn] ОП на месте любой строки [bc] кэшпамяти, так как в качестве ассоциативного признака для поиска строки используется 13-разрядный тег, однозначно определяющий любую и только одну строку ОП.
2. Основными требованиями для создания ВС с переменным составом оборудования, различной производительностью и стоимостью является их: ♦ информационная совместимость; ♦ программная совместимость снизу доверху; ♦ аппаратная совместимость; ♦ конструктивная совместимость; ♦ эксплуатационная совместимость. В настоящее время широкое применение нашли несколько семейств ЭВМ различных западных фирм: Intel, Motorolla, DEC. При этом в каждом семействе для различных поколений микропроцессоров (МкПр) фирмы выпускают несколько типов процессоров и обрамления к ним (периферийных БИС). Например, фирмой Intel выпускаются МкПр следующих типов: ♦ высокопроизводительные центральные процессоры, предназначенные для сложных математических расчетов, но имеющие высокую стоимость; ♦ удешевленные варианты данного поколения МкПр с несколько худшими техническими характеристиками и предназначенные для использования в ЭВМ для учрежденческих применений, периферийных ЭВМ в составе сетей, во встроенных микроконтроллерных системах, где не требуется высокое быстродействие; ♦ МкПр с малым потреблением мощности, предназначенные для применения в компьютерах типа Note Book. С развитием элементной базы и появлением новых типов МкПр существенно улучшаются технические характеристики ВС, но при этом для преемственности (совместимости) использования ранее разработанного программного обеспечения (ПО) любая модель семейства должна удовлетворять вышеперечисленным требованиям, т.к. стоимость разработки нового ПО на порядок и выше превышает стоимость аппаратных средств. Информационная совместимость ЭВМ предполагает использование единых способов кодирования информации, форматов и типов данных, одинаковые или кратные длины машинных слов в различных моделях. Программная совместимость означает, что программы, написанные для одной модели, должны выполняться для других моделей семейства. Это предполагает наличие единой системы и форматов 3 команд, режимов адресации, что позволяет использовать общие ОС и прикладное ПО для моделей одного семейства снизу доверху. Программной совместимостью объясняется наличие большого числа форматов команд и системы команд для старших моделей МкПр, что затрудняет их изучение. Аппаратная совместимость заключается в возможности подключения к любой модели ЭВМ, состоящей из центрального процессора (ЦП) и ОП любых контроллеров периферийных устройств (ПУ), общих для всех моделей ряда. Это достигается за счет использования унифицированных интерфейсов ввода-вывода и единых протоколов обмена между ПУ и ЦП. Однако следует заметить, что практически каждая новая модель МкПр имеет свой состав аппаратных средств БИС, но которые программно и аппаратно совместимы с предыдущими версиями БИС. Конструктивная совместимость подразумевает использование унифицированных панелей, блоков и ТЭЗов (плат) с единой системой назначения контактов разъемов и типов разъемов, конструктивного исполнения системного блока и разбивки ТС на конструктивные модули. Эксплуатационная совместимость предполагает общие методы технической и математической эксплуатации и обслуживания, т.е. преемственность языков программирования, единых ОС, программ технического обслуживания и диагностики, единые методы профилактики ТС и т.д., что не требует переквалификации и дополнительного обучения обслуживающего персонала. Все рассмотренные особенности и требования к семействам ЭВМ позволяют создавать системы переменной конфигурации с возможностью постепенного, по мере необходимости, наращивания вычислительной мощности ВС путем замены ЦП более производительным, расширения емкости ОП, подключения новых и замены устаревших ПУ. При этом за счет программной совместимости моделей одного семейства ЭВМ все ранее существующее программное обеспечение сохраняется. Улучшение технико-экономических характеристик моделей семейства ведется в следующих направлениях: \* совершенствования элементной базы, т.е. применения новых более быстродействующих БИС и СБИС; \* повышения производительности ВС за счет применения новых технологий и структурных решений как для ЦП, так и для периферийных БИС; \* увеличения объема ОП и ВЗУ, совершенствования организации хранения данных, реализации виртуальной памяти, использования как внутренней на кристалле, так и внешней кэш-памяти; \* дальнейшего развития системы ПО как в области прикладных программ, так и ОС; \* создания мультипрограммных и многомашинных вычислительных систем, работающих в реальном времени, локальных и глобальных ВС; \* развития системы ввода-вывода и расширения номенклатуры ПУ; \* повышения надежности ВС, развития эффективных систем контроля и диагностики; \* за счет совершенствования технологии. Технические средства семейств ЭВМ постоянно совершенствуются и развиваются в следующих направлениях: \* модификации - создании нескольких моделей одного и того же устройства, отличающихся друг от друга значением какого-либо параметра или набором функции. Как правило, модификации появляются вследствие доработки предыдущих версий устройства для устранения каких-либо недостатков или дополнения функциональных возможностей устройства без изменения его технических характеристик. Например, таймер ВИ54 является модификацией предыдущей модели ВИ53, в котором реализована дополнительная функция фиксации и считывания приказа обратного считывания; 4 \* модернизации, т.е. замены устаревших образцов устройств новыми с улучшенными техническими, функциональными, конструктивными и эксплуатационными характеристиками. Например, для матричных принтеров выпускается множество модернизаций по ширине каретки, по количеству и расположению иголок, быстродействию и т.д.; \* создания принципиально новых устройств. Например, лазерный принтер является принципиально новой разработкой среди принтеров, как по принципу функционирования, так и по техническим характеристикам.

**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №3**

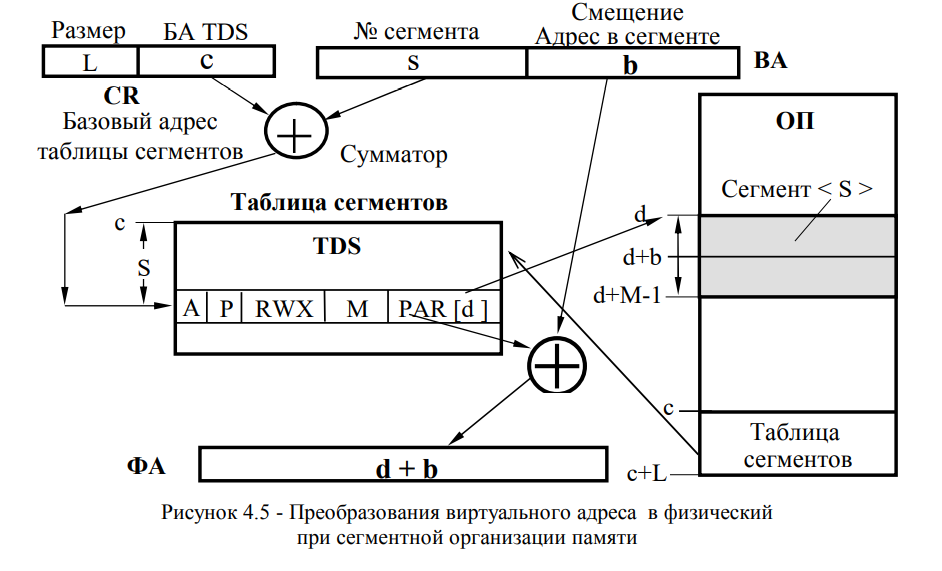
1. Стратегии произвольного замещения не учитывают частоту обращений к строке со стороны ЦП, и велика вероятность, что кандидатом на удаление будет назначена строка, к которой ЦП часто обращается, что снижает общую производительность системы. Однако эти методы наиболее просты с точки зрения технической реализации. Счетчик адресов в качестве кандидата на удаление (КНУ) определяет последовательные ячейки памяти тегов от 0 до N. По стратегии FIFO удаляется строка, которая самой первой была переслана в кэш-память (по времени пребывания в кэш-памяти). Как правило, стратегия используется совместно с другими методами, например, со стратегией по биту неиспользования. Методы, учитывающие активность строк, позволяют сократить число промахов при назначении кандидата на удаление из кэш-памяти, а следовательно, и повысить эффективность использования кэш-памяти. Стратегия LRU: удаляется наиболее давняя по использованию строка. Идея организации псевдо LRU-стека заключается в следующем: при каждом обращении к строке ее номер помещается в стек, в результате замене подлежит строка, хранящаяся в наиболее глубокой позиции стека, и эта строка подлежит удалению первой. Методы замещения кэш- памяти Методы произвольного замещения (наугад) Методы, учитывающие активность строк С использованием счетчика адресов замещения Стратегия стека FIFO По признаку неиспользования Стратегия псевдо LRU-cтека 27 Обращение к строке: b d a a abcd ⇒ bacd ⇒ dbac ⇒ adbc ⇒ adbc ⇒ Техническая реализация метода псевдо LRU на основе стека весьма проблематична, т.к. сдвиг информации в стеке выполняется не во всех ячейках, а носит случайный характер. Обычно метод псевдо LRU-стека реализуют на основе управляющего автомата с жесткой логикой. Реализация автомата усложняется по мере увеличения числа строк в кэш-памяти, т.к. число возможных комбинаций расположения данных в LRU (число состояний автомата) в общем случае без минимизации составляет n!, где n число замещаемых строк в кэш-памяти. Для 4 строк получим 4!=24 состояния автомата. Поэтому этот метод используется только при частично-ассоциативном распределении, когда число модулей не превышает 2-4, т.е. LRU имеет глубину 4 (для каждой группы строк (модулей)). Ниже приведен пример графа переходов состояний автомата для LRU-стека глубиной 3. В ЦП i486 реализован следующий алгоритм замещения строк. Пусть В0, В1 и В2 - код текущего состояния автомата с учетом минимизации, однозначно определяющий номер строки в группе, которая подлежит замещению, а L0, L1, L2 и L3 замещаемые строки в группе строк. При каждом попадании в кэш-память биты состояния В0-В2 модифицируются путем изменения кода состояния. В таблице представлен алгоритм выбора замещаемой строки и формирования функций возбуждения для перехода в следующее состояние. После замещения строки новое состояние автомата заносится в блок LRU. Код состояния В0 В1 В2 Замещаемая строка Номер последней замещаемой строки Изменение бита состояния 0 0 X 0 1 X 1 X 0 1 X 1 L0 L1 L2 L3 Если L0 или L1, Если L2 или L3, Если L0, Если L1, Если L2, Если L3, то В0=1 то В0=0 то В1=1 то В1=0 то В2=1 то В2=0 При технической реализации данного метода необходимо иметь память LRU состояний автомата для каждой группы строк, емкостью равной емкости одного модуля памяти тегов. На рисунке 2.13 приведен фрагмент схемы памяти бит достоверности и LRU состояний для кэшпамяти с частично-ассоциативным отображением. Рассмотрим алгоритм работы схемы. При чтении из памяти тегов параллельно по номеру индекса группы строк (поле b) из памяти LRU состояний считываются и загружаются в регистры бит достоверности строк Rgd и состояний RgS автомата содержимое бит достоверности строк данной группы (4 бита) и 3-разрядный код B0-B2 LRU состояния автомата для группы строк подлежащих замещению. 012 102 201 120 021 210 201/1 Удаляется строка 1: она выбирается как наиболее давняя по использованию 0 2 1 1 1 1 0 2, r 2 2 0 0 2 1, r 0, r 0, r 1, r 2, r r - показывает запрос на замещение строки 28 Если хотя бы один бит достоверности d=0, то удаление строки из кэш-памяти не производится, а запрашиваемая строка считывается из ОП и записывается в модуль, выбираемый сигналом ~CSi, формируемым с выходов дешифратора DC. После замещения строки в память тегов в одноименный модуль записывается новый тег загруженной строки, а в регистре бит достоверности Rgd устанавливается соответствующий бит. Переход к пункту 4. Если все биты достоверности строк в группе равны "1" и А∉Teg, то на выходе КС1 формируется код (номер) модуля памяти тегов строки, подлежащей замещению в двоичном или унитарном коде и на выходах MS2 формируется сигнал ~CSi выбора модуля из группы. В зависимости от принятой стратегии обновления ОП (сквозная или обратная запись) выполняется процедура удаления выбранной (одной из четырех) строк из кэш-памяти в ОП (при простой обратной записи), а затем процедура замещения затребованной строки из ОП в кэш-память. 4. Выполнить требуемую операцию чтения или записи информации с кэш-памятью и параллельно на выходах КС2 в зависимости от набора входных сигналов ~CSi формируется код следующего состояния автомата, определяющего нового кандидата на замещение, и этот код состояния записывается в RgS, а затем в память LRU состояний параллельно со считанными ранее битами достоверности группы строк. По признаку неиспользования: для каждой строки кэш-памяти вводится бит активности строки, который устанавливается при каждом обращении к строке при чтении или записи в кэш-память. При частично-ассоциативном распределении память бит неиспользования строится по аналогии памяти бит достоверности в предыдущем примере (рисунок 2.13). Для полностью ассоциативного и распределения секторов реализация данного метода требует значительных аппаратурных затрат и может использоваться только для памяти тегов небольшой емкости (16-64 ячеек). На рисунке 2.14 приведен пример реализации схемы для полностью ассоциативной кэш-памяти с флагов ой обратной записью и назначением кандидата на удаление по признаку неиспользования. Регистр битов активности можно построить на асинхронных RS-триггерах. Унитарный код, формируемый с выходов схем сравнения АЗУ при ассоциативном поиске, служит источником для установ1111 1110 1111 1000 1 0 1 1 0 0 0 0 1 1 1 1 • • • Di Rg d Rg S KC 1 KC 2 • & MS 1 b MS2 С выходов схем сравнения памяти тегов ~CSi Функции возбуждения ~CS0 ~CS1 ~CS2 ~CS3 Выбор модуля кэш-памяти Рисунок 2.13 - Фрагмент схемы памяти бит достоверности и LRU состояний для кэш-памяти с частично-ассоциативным отображением • CD DC 4 4 3 4 А∈Teg DI Память LRU-состояний и бит достоверности строк Биты di LRU-состояния 0 ••• 4 63 0111 101 29 ки бита активности по входу Si Rg BN. Биты активности, установленные в "0", определяют строки кэшпамяти, как кандидата на удаление из кэш-памяти в ОП или на замещение. 1. Если А∈Teg, то по адресу, сформированному схемой CD1 или выбираемому из памяти тегов (поле [f]), при выполнении записи в СОЗУ данных параллельно выполняется запись в память флагов кода "1". 2. Если А∉Teg, то по адресу, формируемому схемой CD2, назначается строка в качестве кандидата на удаление из кэш-памяти. По этому адресу из памяти флагов считывается значение бита w в триггер Tw для определения необходимости выполнения процедуры обновления ОП (при Tw=0 обновление ОП не выполняется). 3. Если Tw=1, то по этому же адресу строка считывается из СОЗУ данных и записывается в ОП по адресу RgTeg[a]. 4. Из ОП считывается новая строка по адресу из RgФА[a] и загружается на место удаленной строки, а в памяти флагов соответствующий бит w сбрасывается, и параллельно в память тегов загружается новый тег из RgФА. Выполнить пункт 1. Если в регистре признака неиспользования все биты будут установлены в "1", то произойдет автоматический сброс всех бит Rg BN и подсчет активности строк начинается сначала. При увеличении емкости кэш-памяти для полностью ассоциативного и секторного распределений биты активности строк можно также хранить в отдельной одноразрядной памяти бит неиспользования, построенной на основе АЗУ. В качестве эталона для сравнения выступает код нуля, а с выходов АЗУ (поле [f]) будет сниматься код адреса строки, подлежащей замещению, если А∉Teg. Однако, такой подход требует значительных аппаратурных затрат и имеет высокую стоимость. Выбор стратегии замещения тесно связан с реализованным методом распределения кэш-памяти. Например, при прямом распределении кандидат на удаление однозначно определяется номером индекса, метод псевдо LRU-стека и биты неиспользования, как правило, применяется только для частичноассоциативной кэш-памяти, FIFO - совместно с битом неиспользования для частично-ассоциативного Память тегов 0 34 3 (0) 1 1 02 2 (1) 1 2 47 0 (2) 1 3 75 6 (3) 1 4 63 4 (4) 1 5 134 7 (5) 1 6 01 5 (6) 0 k 24 1 (k) 0 DI DO Сх.ср. • • • • Сх.ср. & & & 1 & C D 1 C D 2 M S D C A S0 S1 S2 Sk R Сброс Rg BN Флаг w на УУ для анализа А∈Teg WR 1 1 0 1 0 0 Память флагов d0 dk D C A Адрес АЗУ (..) или поле f f V АКНУ Адрес КНУ Рисунок 2.14 - Схема памяти тегов на основе АЗУ с памятью бит неиспользования и памятью флагов Тег для обновления ОП di RgTeg Tw Rg B f RgФA[a] Rg BN k k Tc [b] • 30 распределения и для полностью ассоциативного или секторного распределений при небольшой емкости АЗУ и т.д. На рисунках 2.4, 2.5, 2.6, 2.8 и 2.12 на входах ОП показаны схемы мультиплексирования адреса: при сквозной записи операнда в ОП или при выполнении процедуры обновления ОП старшие разряды адреса выбираются из памяти тегов, а остальные в зависимости от метода распределения кэш-памяти (поле [b]) из RgФА и организации расслоения обращений ОП (счетчик слов в строке СТ S). При выполнении процедуры замещения кэш-памяти адрес строки берется из RgФА. При использовании расслоений обращений в ОП (рисунок 2.5) для организации записи слова в ОП при сквозной записи и при вводе данных в ОП из ПУ в структуру УУП введен дешифратор выбора кристалла, позволяющий выполнять операции чтения/записи либо всей строки по ширине выборки для всех модулей памяти параллельно, либо в один из выбираемых дешифратором модулей памяти.

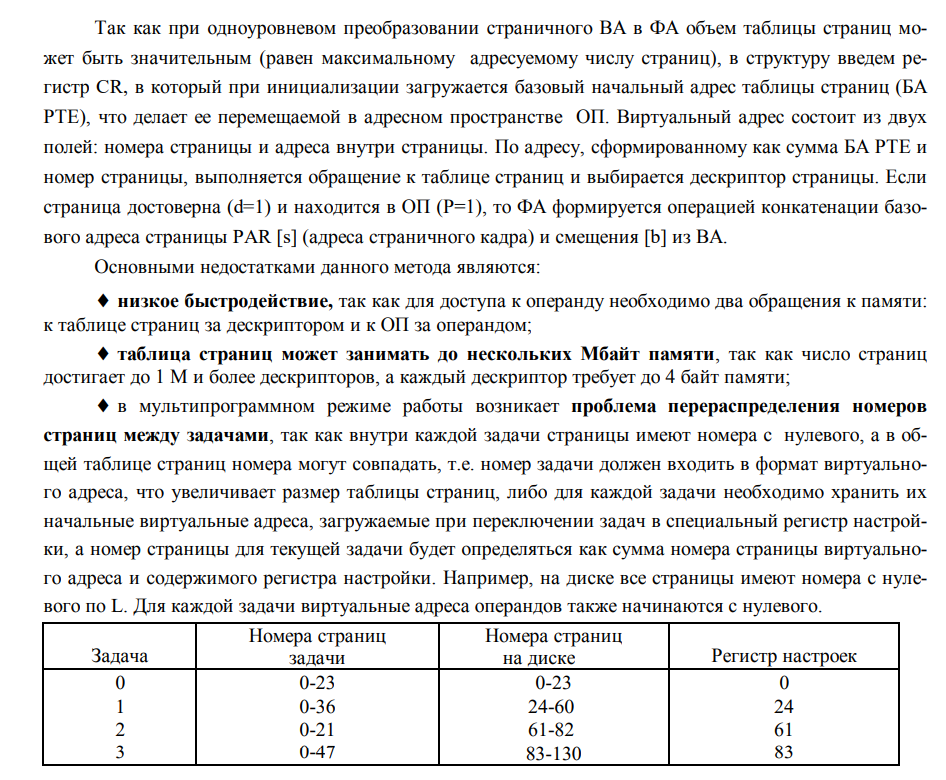
**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №4**

1 . Таким образом, для повышения эффективности работы средств поддержки обмена необходимо иметь средства, обеспечивающие прерывание текущего процесса (программы) и переход к обслуживанию запросов на обмен или обработку специфических ситуаций, называемые подсистемой прерывания. Все прерывания можно разделить на: ♦ внутренние ; ♦ программные ; ♦ внешние. Прерывания, происходящие в результате событий, локализованных внутри модуля системы, например, в процессоре при делении на ноль, обрабатываются с помощью средств внутренних прерываний, в функции которых обычно входит распознавание причины прерывания (формирование вектора прерывания), сохранение текущего состояния процессора (SR, PC и т.п.) и загрузка в программный счетчик начального адреса подпрограммы, обрабатывающей данную ситуацию. Некоторые авторы (программисты) внутренние прерывания относят к программным (так как они не требуют наличия аппаратуры прерывания) и называют их особыми случаями. Особые случаи возникают, например, при нарушении защиты по привилегии, превышении размера сегмента, выходе за границы массива, делении на ноль и т.д. Возникновение данных ситуаций носит случайный характер и особые случаи невозможно предсказать. Все особые случаи (внутренние прерывания) можно разделить на три группы: ♦ нарушение - это такой особый случай, который процессор может обнаружить до возникновения фактической ошибки (например, нарушение правил привилегий, превышение размера сегмента, нарушение атрибутов доступа к сегменту, недействительного кода операции и т.д.). После обработки нарушения (выполнения ППОП) можно продолжить программу путем повторного выполнения (рестарта) виноватой команды; ОП Прикладные программы Стек пользователя ОС Системный стек Область пользователя Системная область Системный стек ОС 44 ♦ ловушка - это такой особый случай, который процессор обнаруживает после окончания выполнения виноватой команды (например, прерывание при переполнении, большинство отладочных команд INT n). После обработки прерывания особого случая процессор возобновляет действия с той команды, которая находится после "захваченной" команды; ♦ авария - это ситуация, когда ошибка настолько серьезна, что ее невозможно устранить и продолжить выполнение программы. При аварии вычислительный процесс прекращается. К таким видам ошибок относятся аппаратные ошибки, обнаруживаемые ОС, или недопустимые значения в системных таблицах. Также заметим, что маскирование особых случаев (внутренних прерываний) невозможно даже запрещением прерываний в процессоре. Обработка при нарушении правил привилегии необходима для защиты ОС. Привилегированной называется команда, выполнение которой разрешается только в системном режиме. К ним относятся: команды изменения содержимого регистра состояния и системных регистров процессора, команда возврата RTI из системного режима, команды останова HALT и сброса RESET, команды ввода-вывода IN и OUT и другие. С помощью внешних (аппаратных) прерываний осуществляется взаимодействие процессора с ПУ (клавиатурой, дисками, таймером и т.д.), сообщается о возникновении ошибок в устройствах от схем контроля (ошибки в памяти, на шине, аварийное выключение питания и т.д.). Функциями средств, обслуживающих внешние прерывания, поступающих от ПУ (аппаратуры прерывания), являются: ♦ фиксация запросов на прерывание от внешних источников; ♦ определение номера приоритетного незамаскированного запроса для обслуживания; ♦ запоминание в стеке состояния текущего процесса (SR, PC и т.д.); ♦ передача управления подпрограмме обслуживания (обработки) данного запроса (ППОП) (поместить символ с клавиатуры в буфер, считать сектор с диска и т.п.); ♦ возврат - восстановление состояния прерванного процесса (программы) и передача ему управления. В зависимости от способа реализации каждой из перечисленных функций подсистемы внешних прерываний могут классифицироваться: ♦ с маскированием входов запросов на прерывание; ♦ без маскирования; ♦ бесприоритетные (запросы на прерывание обслуживаются в порядке поступления); ♦ приоритетные (обслуживание запросов происходит в соответствии с назначенными приоритетами, которые могут быть фиксированными или циклически изменяемыми); ♦ одноуровневые (без вложенности); ♦ многоуровневые, допускающие вложение ППОП в соответствии с назначенным приоритетом; ♦ динамически маскируемые, при которых допускается обслуживание запроса на прерывание от источника с меньшим приоритетом (специального маскирования); ♦ безвекторные, при которых передача управления осуществляется по фиксированному адресу независимо от источника прерывания; ♦ векторные (запрос от каждого устройства обслуживается своей ППОП, для которой служит вектор точек входа (начальный адрес ППОП)). 45 В большинстве процессоров реализована система векторных прерываний, включающая 256 векторов (от 0 до 255), которые закреплены за всеми возможными причинами прерываний: внутренние, внешние, немаскируемые, программные и часть векторов зарезервирована для расширения.

2 . Организация виртуальной памяти (ВП) имеет цель: Предоставить программисту достаточно большое (кажущееся) адресное пространство памяти. Предоставить программисту область памяти, физически не используемую из-за фрагментации, в виде логически непрерывного пространства. При сегментной адресации все пространство адресов делится на множество сегментов различной длины, определяемой необходимым размером данного сегмента. Начальный адрес сегмента называется 38 базовым, а за каждым сегментом закреплен соответствующий номер. Порядок разбиения на сегменты может быть произвольным, а исполнительный адрес определяется номером сегмента (базовым адресом) и смещением внутри сегмента. Отсюда ЛА можно представить в виде двух целочисленных величин (полей): номера сегмента и смещения. При этом разрядность смещения определяет максимальный размер сегмента в байтах. В ряде 16-разрядных процессоров для удобства преобразования ЛА в ФА сегмент разбивается на блоки, кратные 2k-1 байт, т.е. смещение разбивается еще на два поля: номера блока в сегменте и адреса байта в блоке. При сегментной адресации сегмент ограничен разрядностью поля смещения ЛА или максимальным размером, который является величиной постоянной для каждого типа процессора.

В первом варианте используется метод преобразования ВА в ФА через общую для всех задач таблицу дескрипторов сегментов (ее часто называют глобальной GDT). Данному методу присущи серьезные недостатки: 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.



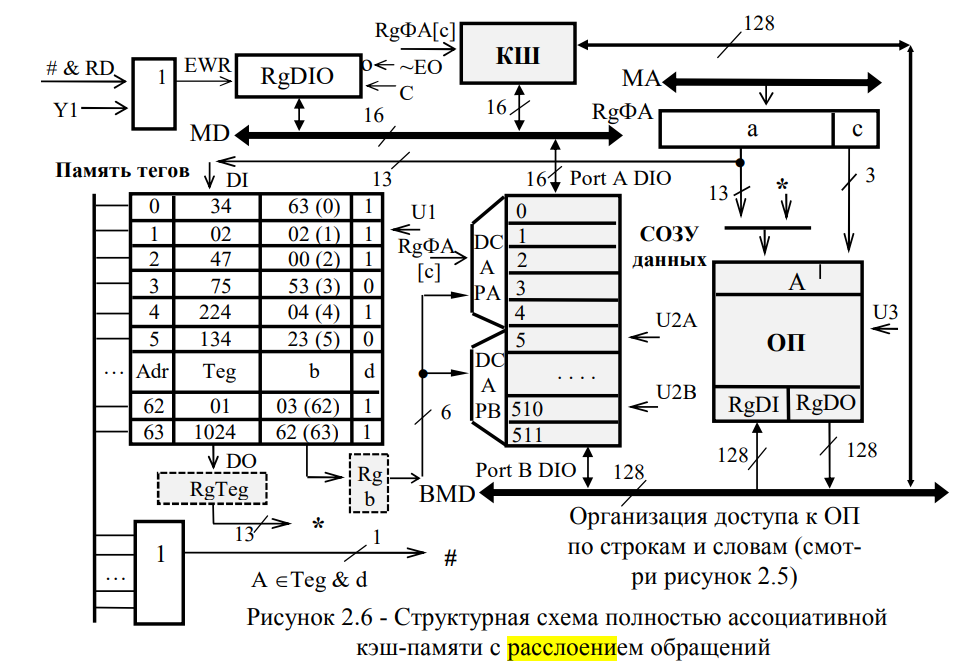


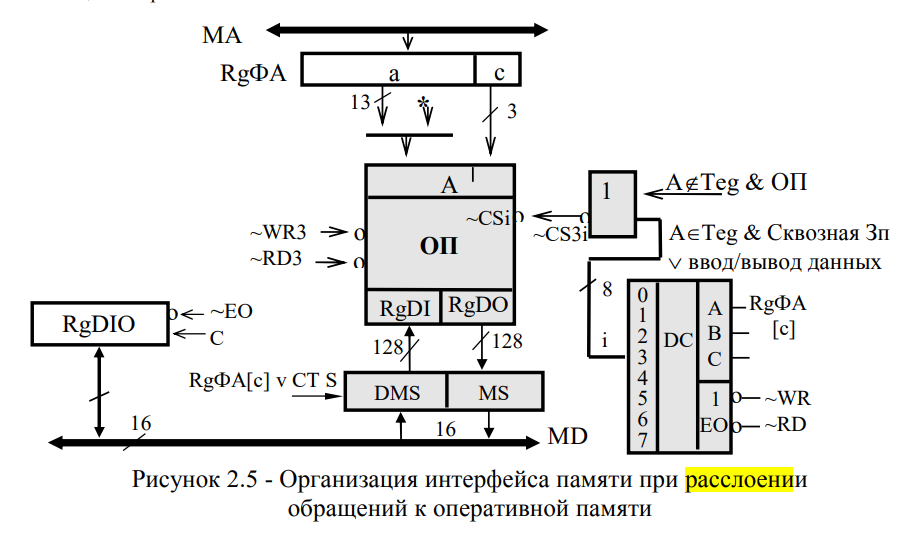
**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №5**

1 . 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. Память тегов 0 34 63 (0) 1 1 02 02 (1) 1 2 47 00 (2) 1 3 75 53 (3) 0 4 224 04 (4) 1 5 134 23 (5) 0 62 01 03 (62) 1 63 1024 62 (63) 1 СОЗУ данных 0 1 2 3 4 5 510 511 RgTeg 13 U1 RgФА [c] 6 1 \* EWR A ∈Teg & d DI DO Port B DIO DC A РА Рисунок 2.6 - Структурная схема полностью ассоциативной кэш-памяти с расслоением обращений . . . . MA RgФА а с 3 • А ОП RgDI RgDO U3 U2A 128 ~EO C RgDIO \* # 128 128 13 13 1 16 U2B 16 Port A DIO КШ 128 DC A РВ • Организация доступа к ОП по строкам и словам (смотри рисунок 2.5) o . . . . . . ВМD 16 MD # & RD 1 Y1 RgФА[c] Rg b Adr Teg b d 18 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. СОЗУ данных 0 1 2 3 4 5 62 63 192 129 258 126 . . . . 191 132 0 03 1 1 02 1 2 04 0 3 00 1 4 02 1 5 104 0 62 01 1 63 02 1 Память тегов Тег d Рисунок 2.7 - Связь кэш-памяти с основной памятью d - бит достоверности строки в кэш-памяти а - тег; b - индекс (адрес группы строк); а.b - адрес строки в ОЗУ Оперативная память 0 1 2 3 4 5 62 63 0 1 2 3 4 127 Номер тега И н д е к с . . . . . . . . . . . . . . . . . . . . . . 3 8128 8129 8130 8131 8132 8133 8190 8191 0 1 2 3 4 5 62 63 64 65 66 67 68 69 126 127 128 129 130 131 132 133 190 191 192 193 194 195 196 197 254 255 256 257 258 259 260 261 318 319 b a ⇒ 02 ⇒ 132 2 34 Память тегов 0 34 1 1 02 1 2 47 1 3 75 0 4 224 1 5 134 0 62 01 63 1024 1 СОЗУ данных 0 1 2 3 4 5 510 511 RgTeg 7 U1 RgФА[c] \* EWR A ∈Teg DI DO DIO D C A # - выходы трехразрядного счетчика слов CT S Ui - сигналы управления блоком памяти CSi, WRi, RDi, (i=1-3) EWR=A∈Teg & RD ∨ Y1; Y1-сигнал разрешения записи в RgDIO из процессора Рисунок 2.8 - Структурная схема кэш-памяти без расслоения обращений с прямым отображением . . . . MA RgФА а с 3 • А ОП RgDI RgDO U3 U2 16 ~EO C RgDIO \* # 16 16 7 1 7 b D C A • Сх. сравн. • • 6 & ο • MD 19 Следующие младшие 6 бит адреса (поле b) строки используются в качестве адреса теговой памяти и СОЗУ данных, указывающие на одну группу из 128 строк ряда, хранимых в ОП. Из теговой памяти считывается тег (номер столбца ОП) и сравнивается с 7-ью старшими разрядами адреса. Если они совпадают, то это означает, что строка, содержащая текущий адрес i, находится в СОЗУ данных. Параллельно считыванию тега осуществляется чтение данных из СОЗУ данных по адресу, образованному 9-ью младшими разрядами адреса i (поля [b.c]), и если A∈Teg, то данные выдаются (записываются) из/в СОЗУ данных на MD в/из RgDIO. Если считанный тег отличается от поля [a] (А∉Teg), то выполняется процедура обновления строки ОП из кэш-памяти и замещения кэш-памяти из ОП согласно принятой стратегии. Таким образом, если А∈Teg, то при чтении время доступа к кэш-памяти составляет один такт, а время такта равно: tкэш= (max tтеg = (τтеg+ τсх.ср. + τ&), tсозу дан. = (τсозу дан. + τrgdio)). При записи: tкэш = (2 × max (tтеg , tсозу дан.)) (или два такта). 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. Недостаток: 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). Принципы организации доступов к памяти с расслоением обращений и использование системного интерфейса на ширину выборки применимы к любому механизму преобразования адресов, так как данные методы являются универсальными и в дальнейшем не рассматриваются.

2 .

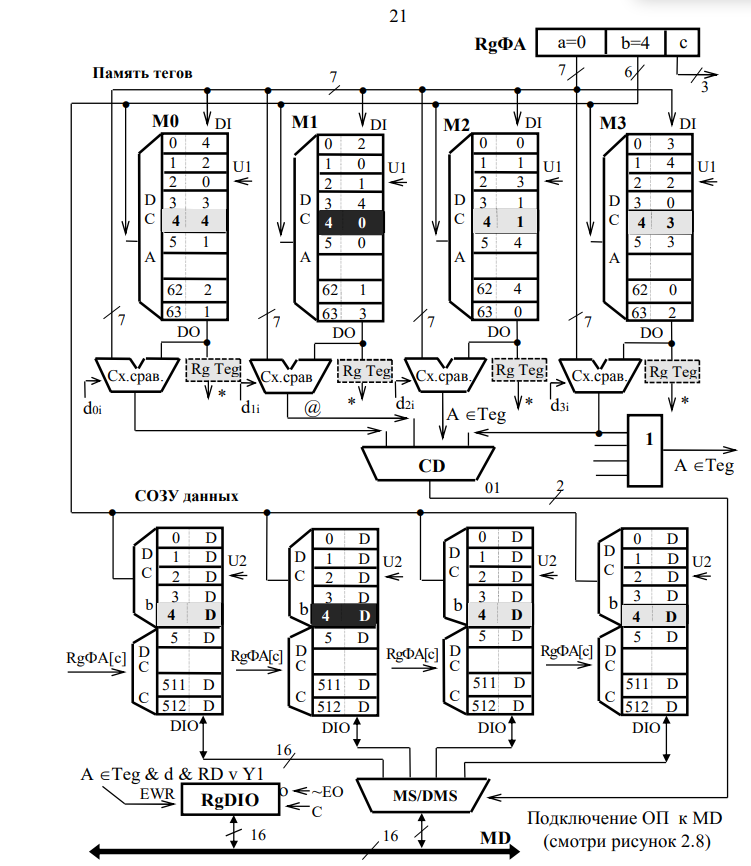
Рассмотрим четыре варианта: ♦ ОП и СОЗУ данных без расслоения обращений; ♦ ОП с расслоением обращений на ширину выборки строки, СОЗУ данных без расслоения обращений; ♦ ОП и СОЗУ данных с расслоением обращений на ширину выборки строки. ♦ ОП с расслоением обращений на ширину выборки строки, СОЗУ данных с расслоением обращений и двусторонним доступом. Первый вариант схемы включения представлен на рисунке 2.4. Процедура обновления ОП требует 8 обращений к СОЗУ данных и 8 обращений для записи строки в ОП, т.е. потребуется 8 х Tозу+ 8 х tсозу дан. и столько же обращений для замещения строки кэш-памяти. Второй вариант (рисунок 2.5) требует наличия на выходах ОП RgDO с выходами с zсостоянием или мультиплексора на 8 входов, а на входах соответственно демультиплексора и регистра RgDI. Тогда процедура обновления ОП потребует 8 x tсозу для чтения строки из СОЗУ данных и записи в RgDI через DMS и одного обращения Tозу к ОП для записи строки и столько же обращений для замещения строки кэш-памяти. При третьем варианте организации ОП и СОЗУ данных потребуется 8 х n-разрядная MD, что приводит к нарушению стандарта системного интерфейса (разрядность системной магистрали данных должна быть n-разрядной). Поэтому этот вариант является неприемлемым и не рассматривается. Четвертый вариант организации магистралей, ОП и СОЗУ данных представлен на рисунке 2.6. Схема включает двухпортовое СОЗУ данных, где порт А связан через МD с процессором и RgDIO. MD имеет разрядность n бит (т.е. без расслоения обращений), а порт В выполнен с расслоением обращений на ширину строки и связан с входами/выходами ОП через 8 х n-разрядную внутреннюю магистраль данных для обмена строками за одно обращение к ОП и СОЗУ данных. MD и BMD связаны между собой через контроллер шины (КШ) для выполнении сквозной записи данных в кэш-память и ОП и загрузки в ОП программ и данных из ВЗУ. КШ представляет собой схему мультиплексора/демультиплексора, управляемого контроллером кэш-памяти и ЦУУ. Процедура замещения строки А ОП RgDI RgDO 128 128 MA RgФА а с \* 3 13 о о 1 DC A B C 1 EO o o 0 1 2 3 4 5 6 7 A∉Teg & ОП DMS MS 16 RgФА[c] v CT S о о RgФА [c] ~WR ~RD ~WR3 ~RD3 16 ~CS3i Рисунок 2.5 - Организация интерфейса памяти при расслоении обращений к оперативной памяти MD ~EO C RgDIO o ~CSi A∈Teg & Сквозная Зп ∨ ввод/вывод данных i 8 17 выполняется за одно обращение к ОП и одно обращение к кэш-памяти. Однако этот метод необходимо рассматривать только для перспективных разработок, когда ОП конструктивно будет размещаться на одной плате (ТЭЗе) с ЦП или между ОП и внутренней кэш-памятью подключается внешняя быстродействующая кэш-память большой емкости. Иначе надежность работы системы резко снижается, а реализация внешней дополнительной 128-разрядной шины данных не дает существенного выигрыша в быстродействии, так как при вероятности попадания в кэш-память 90% выигрыш составит 0,1 х 7 тактов = 0.7 такта на команду для рассматриваемого примера. Использование дополнительной шины данных оправдано только в мультипроцессорных системах при подключении к этой шине внешней кэш-памяти большой емкости для обеспечения независимого доступа к внешней кэш-памяти со стороны нескольких процессоров по приоритету с арбитражем шины.



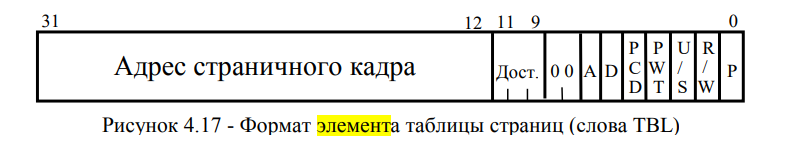


**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №6**

1 . 

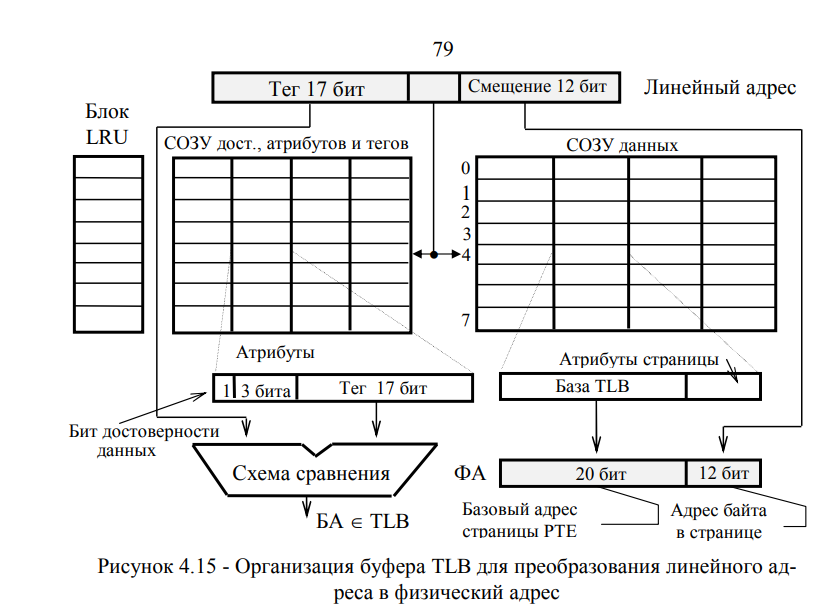


Является развитием метода прямого и ассоциативного распределения. При этом методе кэшпамять строится из модулей, образующих группу строк - не менее двух модулей, например 4. При данном распределении для выбора группы строк используется метод прямого распределения, а для выбора модуля в группе (строки) - метод полностью ассоциативного распределения (рисунок 2.9). Поле [b], как и при прямом распределении, адресует множество строк, отстоящих друг от друга с шагом кратным 64, а старшие разряды адреса также используются в качестве тега для выбора одной строки из этого множества. При этом в кэш-памяти для каждого адреса поля [b] можно хранить группу до четырех строк из этого множества. Выбор модуля (или одной строки из группы (четырех строк)) определяется по совпадению тегов (старших разрядов адреса) в данной группе и запрашиваемой строки. Для доступа к массиву тегов, в котором хранятся 7 старших разрядов адреса (поле а), используются 6 младших разрядов адреса [b] строки (индекс группы строк), при этом одновременно осуществляется обращение и к СОЗУ данных. Из четырех модулей теговой памяти выбираются 7-разрядные теги (4 тега) строк и сравниваются с адресом тега из RgФA[а]. 20 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]) (рисунок 2.10). 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. Достоинства: 1. Высокое быстродействие: чтение за один такт. Время такта равно: tкэш={max tтеg=(τтеg +τсх.ср. +τcd + τлэ); tсозу.дан.=(τсозу дан. + τms + τrgdio)}, а запись за 2 такта tкэш = 2 × max {tтеg, t созу.дан.}; 2. Незначительные аппаратурные затраты (использование БИС памяти с произвольным доступом статического типа); 3. Возможность получения достаточно большой емкости кэш-памяти (при неизменной разрядности полей RgФА емкость кэш-памяти увеличилась в 4 раза); 4. Сокращение числа промахов по сравнению с методом прямого распределения, так как кандидат на замещение строки выбирается как один из четырех модулей по одной из стратегий замещения). Модификацией структуры кэш-памяти может служить вариант, исключающий из структуры шифратор, а выходы схем сравнения соединяются с соответствующими входами выбора кристалла модулей ~CSi СОЗУ данных. Однако при этом время обращения к кэш-памяти и для чтения, и для записи будет составлять два такта.

2 . 

Защита на уровне страниц в основном предназначена для предотвращения взаимодействия программ друг с другом. Контроль достоверности обращения к памяти осуществляется параллельно со страничным преобразованием адреса (т.е. не ухудшает характеристики производительности процессора) и является более простым, чем при сегментации памяти, так как не использует полей типа страниАдресная часть команды 15 Селектор CS 0 31 Смещение 0 85 цы (кода или данных) и предела (размер всех страниц одинаков и определяется разрядностью смещения). Различают две разновидности контроля на уровне страниц: ограничение адресуемой области памяти и контроль типа обращений. Доступом к страницам управляют биты элемента таблицы страниц (рисунок 4.17): PCD, PWT, U/S, R/W. Защита применяется к таблицам страниц первого и второго уровня (элементу каталога и страницы). Ограничение адресуемой области памяти. Для страниц используется не четыре, а только два уровня привилегий: Уровень супервизора (U/S=0) для операционной системы, других системных программ (драйверов устройств и т.д.) и защищенных системных данных (таблиц страниц и т.д.). Уровень пользователя (U/S=1) для прикладных программ и данных. При CPL= 0, 1 или 2 процессор работает на уровне супервизора и ему доступны все страницы, а при CPL=3 процессор работает на уровне пользователя и ему доступны только страницы уровня пользователя. Механизм защиты состоит в сравнения бита U/S с полем CPL. Контроль типа обращений. В механизме защиты предусмотрены только два типа доступа к памяти: ♦ разрешение доступа только для считывания из страницы (R/W=0); ♦ разрешение доступа для считывания/записи из/в страницу (R/W=1). Таким образом, используется двухуровневая защита страниц: ♦ если процессор работает на уровне пользователя, то записать информацию можно только в страницы, которые а) принадлежат уровню пользователя (U/S=1, CPL=3): б) отмечены как допускающие считывание и запись (R/W=1); в) страницы уровня супервизора недоступны с уровня пользователя ни для считывания, ни для записи (U/S=1, CPL=0, 1, 2). При попытке нарушения этих правил генерируется прерывание особого случая общей защиты. Следует отметить, что атрибуты защиты элемента каталога (таблицы страниц первого уровня PDE) могут отличаться от атрибутов ее элемента таблицы страниц второго уровня РТЕ, так как процессор контролирует атрибуты защиты элементов PDE и PTЕ раздельно, начиная с первого уровня, что подразумевает четыре возможные комбинации атрибутов защиты R/W для пользовательского режима. Совместный эффект можно описать выражением R/W[TBL]= R/W[PDE] & R/W[PTE]. Данное значение атрибута защиты и устанавливается в бите защиты элемента таблицы страниц, загружаемого в TLB буфер при промахе. Атрибут защиты страниц, используемых в режиме супервизора всегда имеет атрибут разрешения считывания и записи (R/W=1). Когда разрешено страничное преобразование, сначала реализуется защита сегментов, а затем защита страниц и только в том случае, если при сегментации не обнаружено нарушений защиты памяти. Бит PCD управляет кэшированием страницы во внутреннюю кэш-память: при PCD=1 кэширование страницы запрещено. Это бывает необходимо для страниц, которые содержат порты ввода-вывода с отображением на память или для страниц, кэширование которых не дает выигрыша в быстродействии (линейные участки программ или программы инициализации). Бит PWT можно использовать для запрещения сквозной записи при обращениях только к внешней кэш-памяти (PWT=1 - сквозная запись), так внутренняя кэш-память работает только со сквозной записью при любом значении бита PWT. При PWT=0 для страницы разрешается обратная запись только при обращении к внешней кэш-памяти.

**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №7**

1 . 

Кроме того, если таблицу страниц хранить в ОП, время преобразования линейного адреса в физический будет несоизмеримо с преобразованием логического адреса в линейный при сегментации. Для ликвидации этого недостатка в структуру процессора входит устройство страничного преобразования, составной частью которого является ассоциативный буфер преобразования TLB, реализованный в виде частично-ассоциативной памяти, состоящей из четырех групп (модулей) общей емкостью 32 32- разрядных слова (рисунок 4.15). Работа буфера TLB основана на общей идеологии кэш-памяти. Основная таблица страниц хранится в ОП, а базовые адреса активных страниц (наименее давно используемых), с которыми работают в текущий момент времени программы в блоке данных частично-ассоциативной памяти. В устройстве страничного преобразования из линейного адреса по 3-разрядному номеру индекса из памяти тегов выбираются четыре тега, базовые адреса которых загружены в блок данных, и сравниваются с тегом из линейного адреса (старшие 17 разрядов линейного адреса). Если один из тегов совпадает, то из СОЗУ данных параллельно выбирается 20-разрядный адрес страничного кадра, отображаемого на физическую память (базовый адрес страницы в ОП) и 12 бит, описывающих страницу (назначение битов смотри рисунок 4.17). ФА ОП получается операцией конкатенации выбранного 20-разрядного базового адреса страницы и 12-разрядного смещения из линейного адреса. В типичных системах TLB удовлетворяет до 99% запросов на доступ к таблицам страниц. В качестве стратегии замещения в буфере TLB применяется алгоритм псевдо-LRU, как и во внутренней кэш-памяти. Если при страничном преобразовании в TLB не обнаружено совпадение тегов, то выполняется процедура замещения информации из таблицы страниц, находящейся в памяти. Кратко рассмотрим этот процесс. В процессоре реализовано двухэтапное или двухуровневое страничное преобразование линейного адреса в физический на уровне ОП, представленное на рисунке 4.16, что позволяет существенно сократить емкость ОП для хранения таблиц страниц по сравнению с одноэтапным преобразованием (рисунок 4.14). В состав ассоциативного буфера TLB также входит регистр управления CR3, в котором хранится 20-разрядный физический базовый адрес каталога страниц текущей задачи, он называется регистром базового адреса каталога страниц PDBR. Каталог страниц постоянно находится в ОП и не участвует в свопинге. 31 Номер виртуальной страницы 12 11 Смещение 0 31 Номер физической страницы 12 11 Смещение 0 Таблица страниц Линейный адрес Физический адрес Рисунок 4.14 - Одноэтапное преобразование линейного адреса в физический 79 Корневая таблица называется таблицей страниц первого уровня или просто каталогом страниц, содержит 1024 32-разрядных дескриптора, называемых элементами каталога страниц PDE. Каждый элемент таблицы PDE адресует подчиненную таблицу страниц (таблицу страниц второго уровня), т.е. всего допускается иметь до 1024 подчиненных таблиц страниц. Каждая из таблиц страниц содержит 1024 32-разрядных дескриптора, называемых элементами таблицы страниц - РТЕ, и каждый из элементов РТЕ, в свою очередь, адресует страничный кадр в физической памяти. Собственно преобразование линейного адреса в физический состоит из следующих этапов: ♦ старшие 10 бит 31-22 линейного адреса, сдвинутые на два разряда влево, логически складываются с содержимым регистра базового адреса каталога страниц CR3, и по этому адресу из каталога страниц выбирается один из 1024 элементов PDE, который определяет 20-разрядный адрес таблицы страниц (одну из 1024 таблиц страниц); ♦ средние 10 бит 21-12 линейного адреса, сдвинутые на два разряда влево, логически складываются с содержимым выбранного элемента PDE каталога страниц и по этому адресу из таблицы страниц выбирается один из 1024 элементов PТE, который определяет 20-разрядный адрес страничного кадра в физической памяти (базовый адрес страницы в ОП); ♦ 20-разрядный базовый адрес страничного кадра РТЕ совместно с 12 младшими разрядами атрибутов страницы загружаются в ассоциативный буфер TLB страничного преобразования на место назначенной на удаление страницы по алгоритму псевдо LRU-стека, а также выбранные 20 бит базового адреса страничного кадра совместно с 12 битами 11-0 линейного адреса образуют 32-разрядный физический адрес памяти, по которому производится обращение. Тег 17 бит Смещение 12 бит Линейный адрес Блок LRU 0 1 2 3 4 7 СОЗУ дост., атрибутов и тегов СОЗУ данных 1 3 бита Тег 17 бит База TLB Схема сравнения БА ∈ TLB ФА 20 бит 12 бит Адрес байта в странице Базовый адрес страницы PTE Рисунок 4.15 - Организация буфера TLB для преобразования линейного адреса в физический адрес Атрибуты Бит достоверности Атрибуты страницы данных • 80 При обращениях к элементам PDE и PTE параллельно производится несколько проверок, в которых принимают участие младшие 12 разрядов элементов таблиц каталога и страниц, имеющие одинаковый формат: Поле - адрес страничного кадра. В таблице каталогов (элемент PDE) в этом поле указывается адрес таблицы страниц, а в таблице страниц (элемент РТЕ) -базовый адрес страницы, содержащей данные или команды. Биты системного программиста. Биты 11-9 аппаратно не устанавливаются, а могут быть использованы разработчиками операционных систем, например, для хранения информации о активности страниц, загруженных в ассоциативный буфер TLB (т.е. о том, как часто они используются). Биты обращения А (Accessed) и записи D в страницу (Dirty) (неудачный перевод "грязный") содержат информацию об использовании страницы. Бит А устанавливается аппаратно при каждом обращении к странице при записи или при чтении из нее, т.е. при обращении к странице первого и второго уровней до выполнения операции чтения или записи в таблицу, а сбрасывается программно (сбросом бита А может управлять программист). R / W P P W T U / S P C D Дост. 0 0 A D 31 12 11 9 0 Адрес страничного кадра Рисунок 4.17 - Формат элемента таблицы страниц (cлова TBL) PDRB CR3 Каталог Таблица Смещение 31 12 31 22 21 12 11 0 Линейный адрес (ВА) 31 0 31 0 1023 0 1023 0 Каталог страниц 1023 0 Таблицы страниц Физическая память Страничный кадр 4095 0 БА+См БАстр № РТЕ . . . . . . . . . . . . . . . . БА РТЕ № РDЕ БА РDЕ Рисунок 4.16 - Двухэтапное преобразование линейного адреса в ФА PDE PTE Байт \* 81 Бит D устанавливается также аппаратно только при записи в страницу и только в элементе РТЕ (в элементе каталога PDE этот бит не определен и не участвует в алгоритмах работы страничного преобразования). Периодически проверяя и сбрасывая биты А во всех элементах таблиц страниц, ОС может определить наиболее активные (часто используемые) страницы с привлечением поля достоверности [11-9], в котором можно фиксировать число обращений к странице между каждым сбросом бита А. ОС привлекает бит D при возвращении страницы на диск. Бит D сбрасывается в 0 при загрузке страницы в память, и при необходимости освобождения места в ОП ОС принимает решение о необходимости удаления данной страницы на диск перед загрузкой новой затребованной страницы с диска в ОП (прототип бита флага для обновления ОП в кэш-памяти). Состояние D=0 показывает, что содержимое страничного кадра в ОП не изменялось (не было записи в страницу) и на диске имеется точная копия этой страницы, поэтому не требуется и свопинг для нее. А D=1 указывает на необходимость свопинга страницы, т.е. предварительную перезапись страницы на диск на место старой страницы. Бит присутствия Р (Present) показывает местоположение страницы: в физической памяти (Р=1) или на диске (Р=0). При Р=0 в таблице страниц хранится информация о местоположении отсутствующей страницы на диске в формате: Если бит Р=0 в каком-либо элементе (PDE или PTE), то при обращении к этому элементу со стороны программы возникает особый случай страничного нарушения и в режиме виртуальной памяти реализуется следующий алгоритм замещения страниц (свопинга): 1. Если в ОП нет свободной области памяти для дозагрузки страниц с диска, то ОС по значению бита обращения А (возможно с привлечением поля достоверности (биты 11-9)) определяет страницу в ОП с Р=1 в качестве кандидата на удаление (потерявшего активность) на диск. 2. По значению бита D данной страницы принимается решение о необходимости выполнения процедуры свопинга страницы. Если бит D=1, то ОС копирует страницу из ОП на диск и отмечает ее местоположение в таблице страниц. Далее ОС копирует затребованную страницу с диска в ОП, загружает адрес страничного кадра в элемент таблицы страниц и устанавливает бит Р=1. Также могут устанавливаться другие биты, например, атрибут защиты страницы R/W. 3. Так как в буфере TLB может остаться копия старого элемента таблицы страниц, то ОС очищает его (сброс бита достоверности в TLB). Текущий элемент страницы загружается в TLB буфер на место, определяемое нулевым значением бита достоверности или кодом, формируемым блоком LRU в качестве кандидата на замещение. 4. Осуществляется рестарт команды, вызвавшей особый случай прерывания. Биты считывания/записи R/W и пользователь/супервизор U/S применяются в механизме защиты страниц и рассматриваются ниже. Биты управления кэшированием. Биты PCD запрещения кэширования страницы и PWT сквозной записи применяются для управления кэшированием на уровне страниц и также рассматриваются ниже.

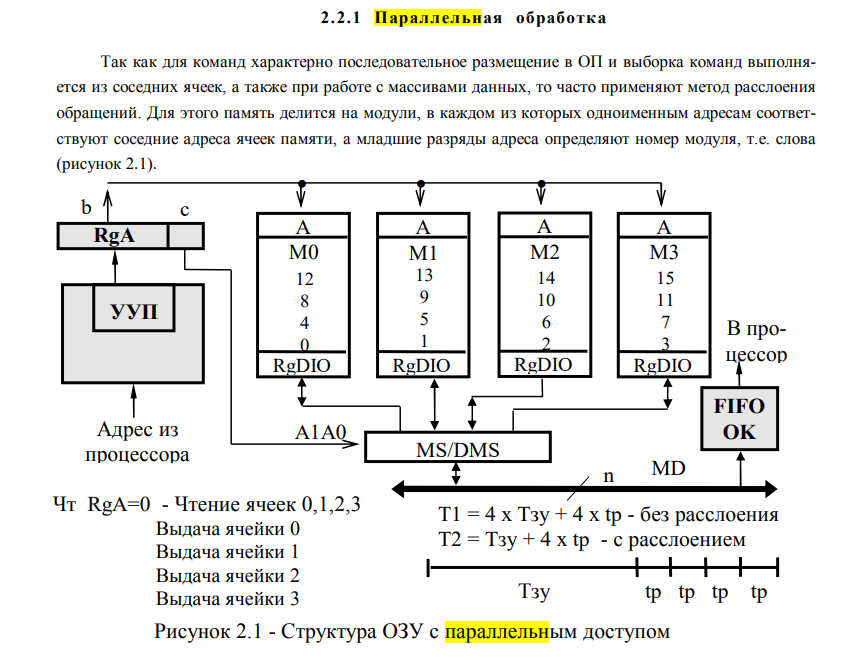
2 .

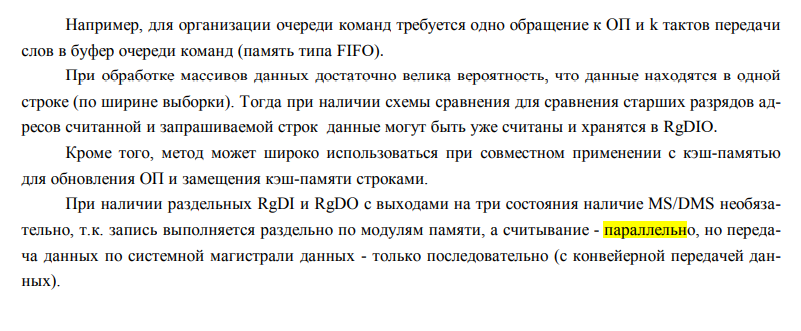
**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №8**

1 .

2 .

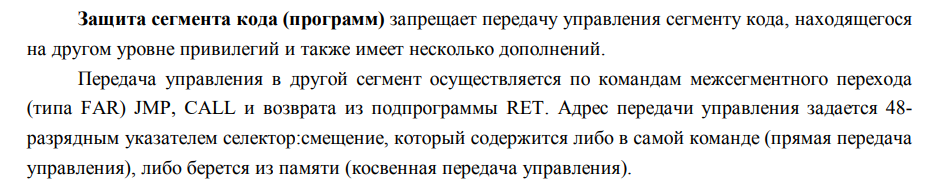
**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №9**

1 . 



2 . Отсюда, в зависимости от того, где применяется тот или иной тип ЦП, состав ТС и технические характеристики, включая стоимость, существенного отличаются. Основными особенностями принципов организации ЭВМ являются: \* проблемная ориентация систем в достаточно широком диапазоне с возможностью гибкого изменения конфигурации системы и переориентации ее на другие классы задач; \* организация параллельной и конвейерной обработки информации как на уровне ЦП, так и на уровне модулей ЦП; \* использование табличных методов обработки данных и принятия решений, особенно ярко проявляется на примере преобразования виртуальных адресов в физические, организации средств защиты памяти и т.д.; \* в ряде систем реализована возможность создания адаптивно перенастраиваемых ВС, конфигурация которых изменяется в процессе решения задачи с целью наиболее эффективной организации вычислительного процесса и обеспечения живучести системы; \* аппаратная реализация ряда функций математического обеспечения (сопроцессоры), что позволяет в существенной степени повысить производительность ВС; \* многоуровневая иерархическая организация памяти ЭВМ с различными техническими характеристиками по емкости памяти и быстродействию. Для МПС в структурной реализации получил дальнейшее развитие принцип 3М - модульность, магистральность и микропрограммируемость. Модульная организация систем. Принцип модульной организации предполагает построение ЭВМ и ВС на основе набора модулей. Под модулем понимается конструктивно, функционально и электрически законченное устройство, позволяющее самостоятельно или в совокупности с другими модулями решать задачи заданного класса или выполнять определенные функции. При этом различают функциональные и конструктивные модули. К функциональным модулям, как правило, относятся БИС, выполняющие строго определенные функции. Например, БИС операционных устройств, БМУ и т.д. для секционированных МкПр, БИС ЦП, блоки приоритетных прерываний, прямого доступа в память для однокристальных микропроцессоров, БИС микроконтроллеров. Из этих модулей собираются конструктивные ТЭЗы (платы) путем объединения функциональных модулей и наращивания разрядности обрабатываемых данных. Конструктивные ТЭЗ (платы) также выполняют строго определенные функции, например, материнская плата, плата контроллера НМД, слоты памяти и т.д., или в систему может входить несколько ТЭЗ с одно- или разнотипными процессорами, на основе которых можно создавать любые конфигурации ЭВМ, отличающиеся друг от друга функциональными возможностями и техническими характеристиками. Модульный подход также способствует сокращению затрат и сроков проектирования, упрощает наращивание мощности и реконфигурацию систем, отодвигает время морального старения ТС. 6 При решении вопроса о функциональном составе модулей существуют две диалектические противоположности: многофункциональность (универсальность) и специализация модулей. Многофункциональные модули позволяют обеспечить: \* сокращение номенклатуры модулей; \* снижение затрат на проектирование и их изготовление; \* высокую серийность, а следовательно, и низкую стоимость. Специализация модулей позволяет исключить избыточность структуры за счет оптимизации схемных решений, реализуемых алгоритмов и функций. Однако специализация модулей низшего конструктивного уровня ведет к необходимости иметь большое число разнотипных модулей, хотя и с высокими техническими характеристиками, за исключением создания систем специализированного назначения. Примерами многофункциональных модулей могут служить ОУ, БМУ секционированных МПК, однокристальные ЦП, программируемые периферийные БИС, реализующие по несколько разнотипных программно настраиваемых режимов работы. Специализированные модули позволяют получать хорошие характеристики по быстродействию, надежности, потребляемой мощности. Магистральный способ обмена информацией. Выделяют два способа взаимосвязей модулей: принцип произвольных связей типа "каждый с каждым" и принцип упорядоченных связей - магистральный, позволяющий минимизировать число связей. При магистральном принципе выделяют три типа шин: данных, адреса и управления, что позволяет обеспечить регулярность структуры МПС как на уровне БИС, так и на уровне связей между конструктивными модулями МПС. Достоинства магистрального обмена: \* минимизация числа связей между конструктивными модулями; \* обеспечение стандартизации интерфейсов (Multibus, ISA, EISA и т.д.); \* сокращение числа выводов БИС; \* единые способы подключения и протоколы обмена между модулями. Микропрограммная организация управления. Для современных МПС характерна многоуровневая организация управления на уровне микрокоманд: с жесткой и программируемой логикой. Принцип микропрограммного управления обеспечивает: \* наибольшую гибкость при организации многофункциональных микропроцессорных модулей за счет использования вызова подмикропрограмм, являющихся общими для нескольких алгоритмов; \* позволяет осуществлять проблемную ориентацию ЭВМ за счет программной настройки на требуемую систему команд или гибкость использования устройств за счет смены микропрограмм путем их загрузки в ОЗУ микропрограмм с внешнего носителя; \* использование макроопераций в МПС, т.е. часть алгоритмов, выполняемых с помощью подпрограмм, может выполняться по отдельным командам на микрокомандном уровне; \* увеличивает регулярность структур за счет использования ПЗУ и не требует при проектировании или доработке УУ в существенной степени изменять схему БМУ, а любые изменения сводятся к корректировке текстов микропрограмм; \* повышает надежность устройств за счет применения БИС памяти; \* упрощает контроль функционирования УУ, который сводится к контролю чтения содержимого ПЗУ микропрограмм. 7 Регулярность структуры. Принцип регулярности предполагает закономерную повторяемость элементов структуры и связей между ними. Применение данного принципа позволяет: \* увеличить плотность интегрального исполнения БИС; \* сократить время топологического и схемотехнического проектирования БИС; \* сократить число типов функциональных и конструктивных элементов; \* повысить серийность, а следовательно, снизить стоимость БИС. Принцип регулярности структур наиболее ярко проявляется при использовании структур и устройств типа памяти (РОН, ОЗУ, ПЗУ, ПЛМ), при использовании магистрального способа обмена, стандартизации интерфейсов, использовании принципа микропрограммного управления и т.д.

**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №10**

1 . 

2 . билет 6.1

**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №11**

1 .

2 .

**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №12**

1 .

2 .

**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №13**

1 .

2 .

**ЭКЗАМЕНАЦИОНЫЙ БИЛЕТ №14**

1 .

2 .