Федеральное государственное автономное образовательное
учреждение высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"

Московский институт электроники и математики

Департамент компьютерной инженерии

МОДЕЛИРОВАНИЕ ПРОТОКОЛОВ КЭШ-СОГЛАСОВАНИЯ В МНОГОПРОЦЕССОРНЫХ/МНОГОЯДЕРНЫХ СИСТЕМАХ

Методические указания к лабораторной работе

Составитель: к.т.н., доцент Е.М.Иванова

Москва 2016

ЦЕЛЬ И ПРАКТИЧЕСКОЕ СОДЕРЖАНИЕ МЕТОДИЧЕСКИХ УКАЗАНИЙ

1.1. Цель работы

Данная лабораторная работа направлена на изучение моделирования работы подсистемы Кэш-памяти в многопроцессорной/многоядерной системе с использованием механизма КЭШ-согласования, реализованного с помощью различных протоколов: MSI, MESI, MOSI, MOESI/MESIF; закрепление теоретического материала по разделу «Архитектура ВС», «КЭШ-память».

1.2. Краткое содержание

В настоящих указаниях приводятся сведения об архитектуре подсистемы Кэш-памяти в многопроцессорной/многоядерной ВС, назначении ОП и КЭШ, описание протоколов КЭШ-согласования.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Многопроцессорные/многоядерные ВС (МПВС) имеют большое значение при решении множества вычислительных задач за ограниченное время. В этом случае происходит распараллеливание вычислений между несколькими процессорами. Каждый процессор в системе имеет свои операционные и управляющие устройства, что позволяет назначить каждому процессору свою задачу и ускорить обработку нескольких задач в целом [1]. Однако оперативная память в таких системах общая, что с одной стороны удобно, и позволяет каждому процессору без лишних сложностей получить совместный доступ к общим данным. Как и в любой ВС в МПВС присутствует КЭШ-память. Способы её организации могут быть различными и имеют свои достоинства и недостатки.

2.1. Архитектура памяти МПВС

Сильно связанные вычислительные системы или МПВС основаны на объединении процессоров на общем поле оперативной памяти. Это поле называется разделяемой памятью (Shared Memory) [1]. При этом достигаются более быстрый обмен информацией между процессорами, чем между ЭВМ в многомашинных вычислительных системах (комплексах), где у каждого узла ВС имеется своя локальная ОП и для работы с общими данными приходиться постоянно обмениваться сообщениями по коммуникационным каналам.

Рис. 1. Организация МПВС без кэширования

В МПВС процессоры обмениваются информацией через общую оперативную память. При этом возникают задержки из-за межпроцессорных конфликтов при доступе к общим блокам памяти. Как и в любой другой вычислительной системе, здесь для повышения эффективности работы с ОП используется КЭШ-память, т.к. КЭШ-память, сама по себе, обладает лучшими временными характеристиками.

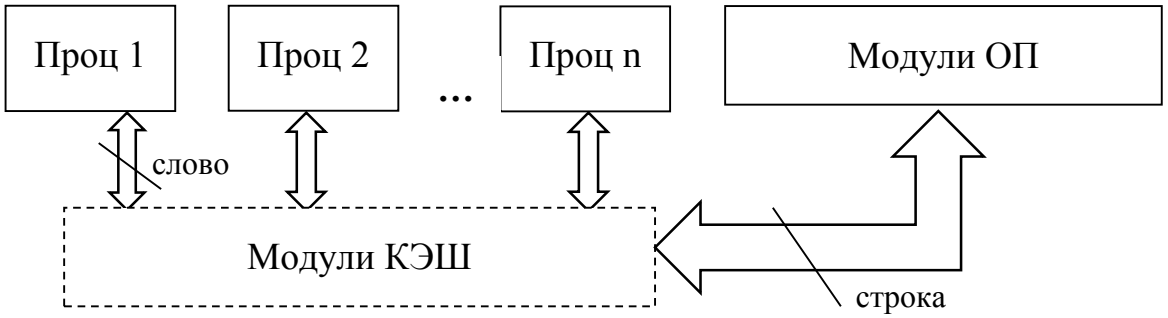


Рис. 2. Организация МПВС с использованием КЭШ

ОП служит для хранения активных программ и данных, то есть тех программ и данных, с которыми работает ВС. Из ОП в процессор поступают коды и операнды, над которыми производятся предусмотренные программой операции, из процессора в ОП направляются для хранения промежуточные и конечные результаты обработки информации.

В функциональном отношении **КЭШ-память** рассматривается как буферное запоминающее устройство, размещённое между основной (оперативной) памятью и процессором. Основное назначение КЭШ-памяти – кратковременное хранение копий данных(блоков) из ОП и выдача запрашиваемых слов из найденной копии блока процессору, что сокращает число обращений к ОП по каналам связи, скорость работы которой меньше, чем скорость КЭШ-памяти.

За единицу информации при обмене между ОП и КЭШ принята строка(блок), причём под строкой/блоком понимается набор слов, выбираемый из оперативной памяти при одном к ней обращении. Хранимая в оперативной памяти информация представляется, таким образом, совокупностью строк с последовательными адресами. В любой момент времени строки в КЭШ-памяти представляют собой копии строк из некоторого их набора в ОП, однако расположены они необязательно в такой же последовательности, как в ОП. Если при запросе от процессора строка, содержащая данные, обнаружена в КЭШ, то произошло КЭШ-попадание, иначе – КЭШ-промах.

Таким образом КЭШ как дополнительный буфер ускоряет работу всей ВС. Исходя из чего, добавляя подобные буферы, можно ещё больше повысить эффективность ВС. В итоге в п/с памяти современных ВС существует несколько уровней КЭШ-памяти (от 2 до 4): L1, L2, L3, L4 (см. рис. 3).

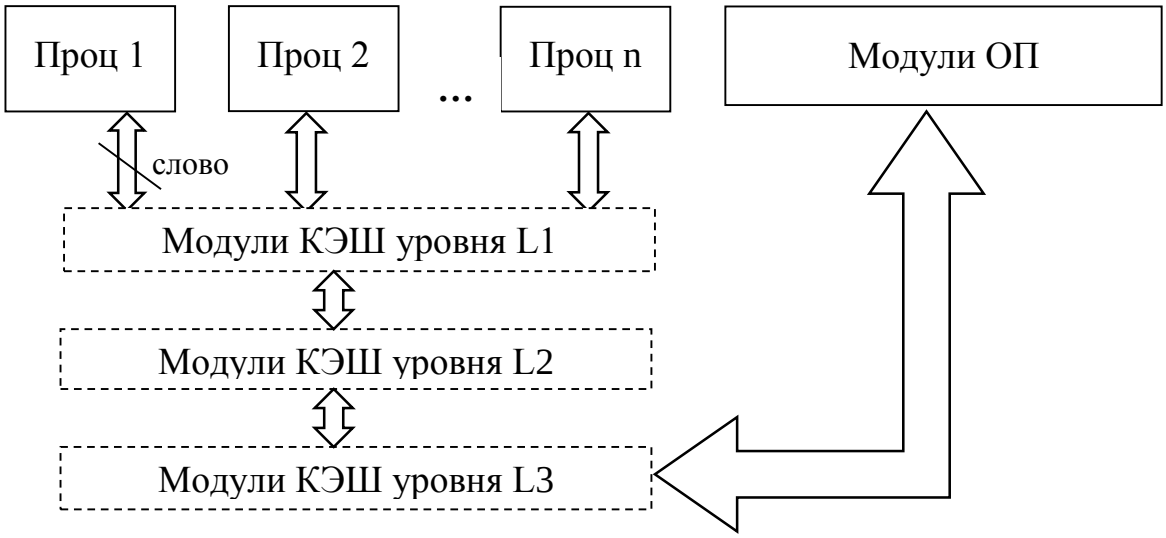


Рис. 3. Организация многоуровневой КЭШ в МПВС

Как правило все КЭШ интегрированы в ЦП и нижние уровни (L1 и L2) являются частью ядра и могут рассматриваться как локальные ЗУ, а последний уровень L3 разделяется между несколькими ядрами/процессорами (см. рис.4).

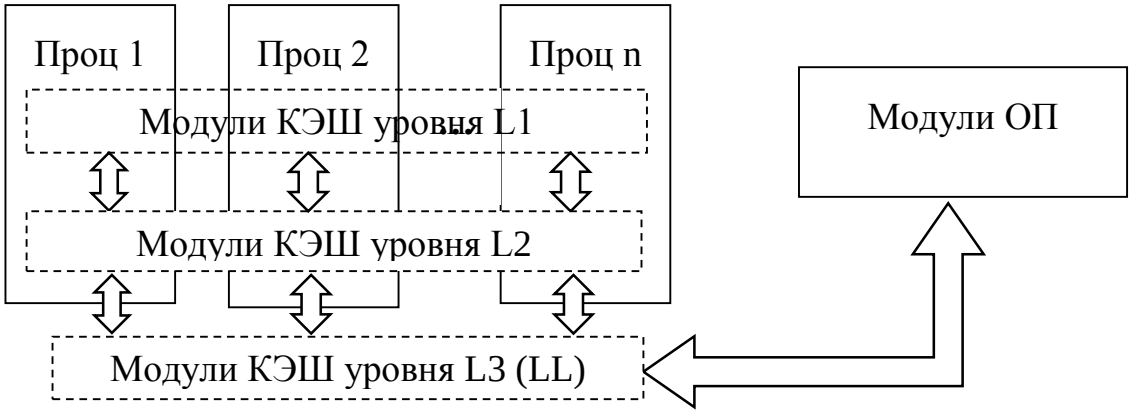


Рис. 4. Порядок размещения уровней КЭШ в ВС

В настоящей работе более детально мы будем изучать последний уровень иерархии – LL (Last Level) КЭШ, который напрямую взаимодействует с ОП. В зависимости от архитектуры это может быть уровень L2, L3 или L4.

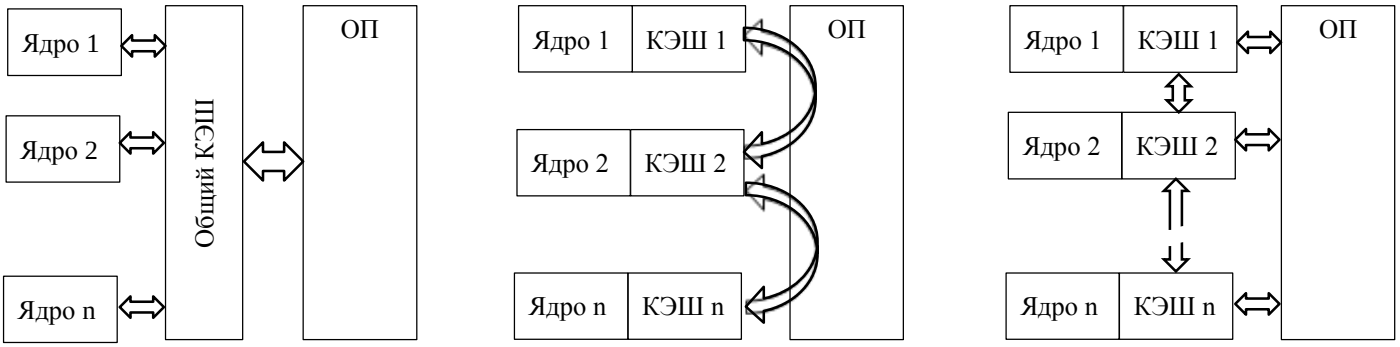
2.2. Организация КЭШ-памяти в МПВС

В МПВС уровень LL КЭШ можно сделать общим для всех процессоров/ядер системы (см. рис.5.а) или раздельным с возможностью взаимодействия. Взаимодействие можно организовать посредством специальных протоколов обмена копиями через ОП (см. рис.5.б) или посредством каналов непосредственной связи между КЭШ-модулями (см. рис.5.в). Здесь и далее модули уровня КЭШ LL обозначены просто как КЭШ для краткости.

Первый способ организации КЭШ в МПВС более прост, но менее эффективен по следующим причинам [2].

- Кэш должен быть очень большим по объёму, а такой КЭШ очень дорог, из-за сложности устройства ассоциативной памяти, поэтому стараются применять КЭШ-память малого объёма.
- Работа с объёмной КЭШ-памятью будет проходить более медленно из-за увеличенной латентности доступа, которая напрямую зависит от ёмкости модуля памяти, что тоже невыгодно.
- Процессоры/ядра конкурируют за доступ к общим областям КЭШ.

Следовательно, для МПВС эффективнее будет применять раздельный КЭШ (см. рис.5.б,в).



а) без поддержания когерентности б) MSI,MESI – протоколы в) MOSI,MOESI, MESIF- протоколы

Рис. 5. Организация Кэш-памяти в МПВС

Повышение эффективности работы вычислительной системы с использованием КЭШ-памяти достигается за счет того, что:

- Процессор работает непосредственно только с быстродействующей КЭШ-памятью.
- КЭШ-память быстрая, т.к. выполнена по технологии SRAM (что в разы быстрее модулей DRAM для ОП).
- КЭШ-память быстрая, т.к. выполнена как часть процессора и время пересылки данных из КЭШ в исполнительные блоки ЦП занимает несколько тактов (1-3) в отличие от передачи данных из ОП по шинам (десятки-сотни тактов).
- Копирование строк из ОП в КЭШ происходит большими блоками (больше, чем запрашивает ЦП при однократном обращении) «про запас на будущее».

В этом случае поиск нужной процессору информации происходит в уже ранее кэшированных строках по поисковому признаку – адресному тегу строки (это сам адрес или часть адреса). Для поиска информации процессор формирует запрос. Этот запрос может быть двух типов: чтение данных из памяти и запись данных в память. После обработки запроса либо процессор может получить найденные данные по указанному в запросе адресу (в случае чтения и КЭШ-попадания), либо инициируется процедура поиска данных в запоминающих устройствах следующего уровня иерархии: ОП, ВП (в случае КЭШ-промаха). Таким образом, реакция подсистемы памяти МПВС на запрос доступа

к данным зависит от множества параметров, и для реализации реакции на различные параметры запроса/состояния системы в любой п/с памяти существует специальный программно-аппаратный блок – КЭШ-контроллер (см. рис. 6).

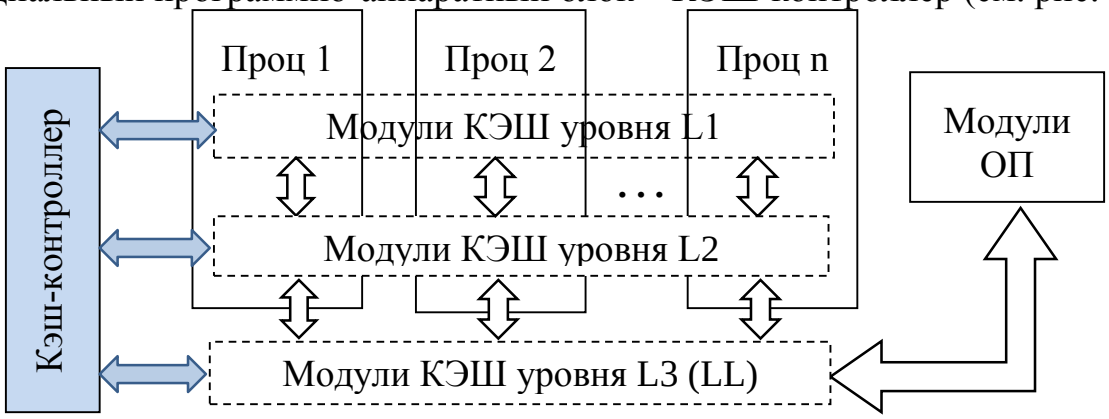


Рис. 6. Организация МПВС с использованием КЭШ

В каждой ВС КЭШ-контроллер реализует свою стратегию работы КЭШ-памяти: способы распределения данных в КЭШ (прямое, ассоциативное, частично-ассоциативное), способы размещения данных по уровням КЭШ (эксклюзивная/инклюзивная стратегия), стратегии обновления копий строк ОП, диспетчеризация запросов, стратегии удаления строк из КЭШ…

2.3. Стратегии обновления строк основной памяти

Поскольку строки КЭШ представляют собой копии соответствующих строк ОП, то в процессе работы (при выполнении запросов на запись/чтение) иногда может потребоваться эти копии обновлять. В таблице ниже приведены условия сохранения и обновления информации в ячейках КЭШ-памяти и ОП при разных режимах.

Табл. 1.Условия сохранения и обновления информации

Режим работы	Наличие копии строки ОП в КЭШ	Информация в строке КЭШ	Информация в строке ОП
Чтение	Копия есть	Не изменяется	Не изменяется
	Копии нет	Обновляется	Не изменяется
Сквозная запись без распределения	Копия есть	Обновляется	Обновляется
	Копии нет	Не изменяется	Обновляется
Сквозная запись с распределением	Копия есть	Обновляется	Обновляется
	Копии нет	Обновляется	Обновляется
Обратная запись	Копия есть	Обновляется	Не изменяется
	Копии нет	Обновляется	Не изменяется

При **чтении** информация из некоторой строки ОП и наличие копии этой строки в КЭШ, вместо оригинала процессором считывается копия данных из

КЭШ. Если копия данных в КЭШ отсутствует, то сначала она создаётся, и потом обращение происходит именно к копии. В этом случае информация ни в кэш-памяти, ни в основной памяти не изменяется.

При **записи** строк существует несколько методов обновления старой информации. Эти методы называются стратегией обновления строк основной памяти [3].

Сквозная запись. По методу сквозной записи обычно обновляется слово, хранящееся в ОП. Если в Кэш-памяти существует копия этого слова, то она также обновляется. Если же в Кэш-памяти отсутствует копия этого слова, то

- либо из ОП в КЭШ пересылается строка, содержащая это слово (сквозная запись с распределением),
- либо этого не допускается (сквозная запись без распределения).

Когда по методу сквозной записи из КЭШ-памяти удаляется блок, назначенный для хранения строки ОП, то в основную память можно не возвращать удаляемый блок, так как копия там уже есть. Недостаток метода: эффект от использования КЭШ-памяти отсутствует, т.к. обращение всё время происходит к ОП.

Обратная запись. По методу обратной записи, если адрес объектов, по которым есть запрос обновления, существует в КЭШ-памяти, то обновляется только КЭШ-память, а основная память не обновляется. Если адреса объекта обновления нет в КЭШ-памяти, то в неё из основной памяти пересылается строка, содержащая этот адрес, после чего обновляется только КЭШ.

При чтении всегда соблюдается один и тот же режим работы, а при записи возможен выбор.

При записи проще реализовать механизм сквозной записи, но тогда эффективность использования КЭШ значительно снижается, т.к. каждый раз при записи приходиться обращаться в меделенную ОП.

В случае обратной записи работа происходит всегда только с КЭШ, что значительно ускоряет обмен. Обращение к ОП происходит очень редко – тогда, когда требуется удалить строку из КЭШ (например, в случае КЭШ-промаха и необходимости размещения новой копии в КЭШ на место удаляемой), то содержимое удаляемой строки необходимо пересылать в основную память. Если результат обновления строк КЭШ-памяти не возвращается в основную память, то содержимое основной памяти становится неадекватным вычислительному процессу.

2.4. Организация КЭШ-памяти в МПВС

Одновременный доступ нескольких процессоров к одной ОП в ВС с разделённым КЭШем можно организовать посредством реализации стратегии как сквозной, так и обратной записи.

В первом случае используются коммутационные сети со сквозной записью данных. Тогда при модификации данных в локальном КЭШ процессора, нужно тут же записывать эти данные в ОП для того, чтобы другие процессоры

могли иметь доступ к изменённым данным. Это простейший алгоритм работы КЭШ-контроллера, но он резко снижает эффективность работы КЭШ [1,5].

Во втором случае дополнительно реализуются специальные программно-аппаратные механизмы (протоколы) взаимодействия. При использовании разделённого КЭШ модифицированные данные не записываются из локального КЭШа в оперативную память до того момента пока строка КЭШ с этими данными не будет удалена из КЭШ, либо пока другой процессор не произведет чтение или модификацию этих данных. Из-за этого значительно меньше приходится записывать данные из локального КЭШ в ОП, чем для первого метода, следовательно, повышается эффективность работы КЭШ.

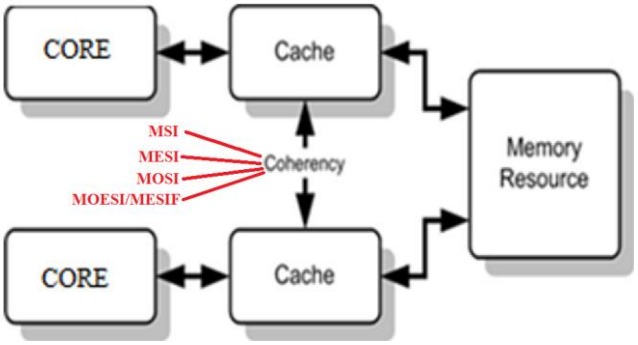


Рис. 7. Когерентность памяти в МПВС

Однако в случае обратной записи и выполнении, например несколькими ядрами/процессорами программ с одними и теми же данными (общими таблицами, массивами…) в каждой локальной КЭШ одновременно существуют несколько копий одних и тех же строк ОП. Эти копии изменяются каждым ядром/процессором произвольно. Разсогласованность копий данных может привести к нарушению когерентности памяти ВС в целом [4].

Под когерентностью следует понимать согласованное изменение всех существующих копий данных, для того, чтобы при запросе на доступ к любой копии любой процесс/процессор получал последние внесённые в эти данные (в оригинал или любую из копий) изменения.

В многопроцессорной/многоядерной системе, где у каждого процессора/ядра имеется своя локальная КЭШ-память, в каждой КЭШ может существовать своя копия одних и тех же данных. Отсутствие когерентности КЭШей будет означать, что каждый процессор/ядро/клиент видит только свои последние изменения этой копии, а изменения внесённые другими процессорами/ядрами в свои локальные копии ему не доступны.

Для борьбы с этой проблемой существует несколько методов работы с копиями или «протоколов поддержания когерентности КЭШей» [5]:

- механизм сквозной записи (Write-Through) читай (write through cache);
- MSI-протокол;
- MOSI-протокол;
- MESI-протокол;
- MOESI-протокол;
- MESIF-протокол.

Механизм сквозной записи

Рис. 8. Схема выполнения запросов ЧТ/ЗП в сетях со сквозной записью

Простейший протокол поддержания когерентности – это так называемый Write-Through, когда любые изменения сразу же записываются (write through cache) не только в КЭШ-память, но и в оперативную память компьютера (см. рис.8). Текущий процессор, вносящий изменения оповещает об этом остальных владельцев копий, а те помечают, что соответствующие строки в КЭШе отныне «недостоверные» (Invalid) и при обращении к ним данные необходимо брать не из КЭШа, а из оперативной памяти. Схема достаточно простая... но неэффективная: запись данных в оперативную память – далеко не быстрый процесс.

Другим способом согласовать копии строк в разных КЭШах является использование специальных программно-аппаратных средств (протоколов поддержания когерентности). Каждый такой протокол подразумевает хранение и изменение информации о статусе (состоянии) каждой копии строки в каждой локальной КЭШ и оригинала строки в ОП. В протоколе также прописывается алгоритм действий при чтении/записи/удалении и при определенном состоянии запрашиваемой копии. Различные архитектуры используют разные механизмы поддержания протоколов. Например, архитектуры с общей шиной часто выполняется процедура арбитража шины (snooping dog), когда запрос на чтение транслируется на все КЭШи. Другие архитектуры включают в себя каталоги КЭШа, которые хранят историю кэширования каждого конкретного КЭШ-блока.

Множество существующих протоколов различаются

- сложностью реализации,
- алгоритмами выполняемых операций,
- скоростью работы,
- наличием/отсутствием непосредственных каналов связи между локальными КЭШ.

MSI-протокол

Каждая строка/блок, содержащейся внутри КЭШа может иметь одно из трех возможных состояний: **M-S-I**

Modified - блок отличается от ОП, но достоверен (значит оригинал в ОП – недостоверен), присутствует только в одном процессоре/ядре. Т.к. копия достоверна, то операции чтения и записи разрешены. Операция ЧТ/ЗП не меняет

статус блока. При удалении такого блока из КЭШ его оригинал в ОП заменяется на изменённую копию из КЭШ.

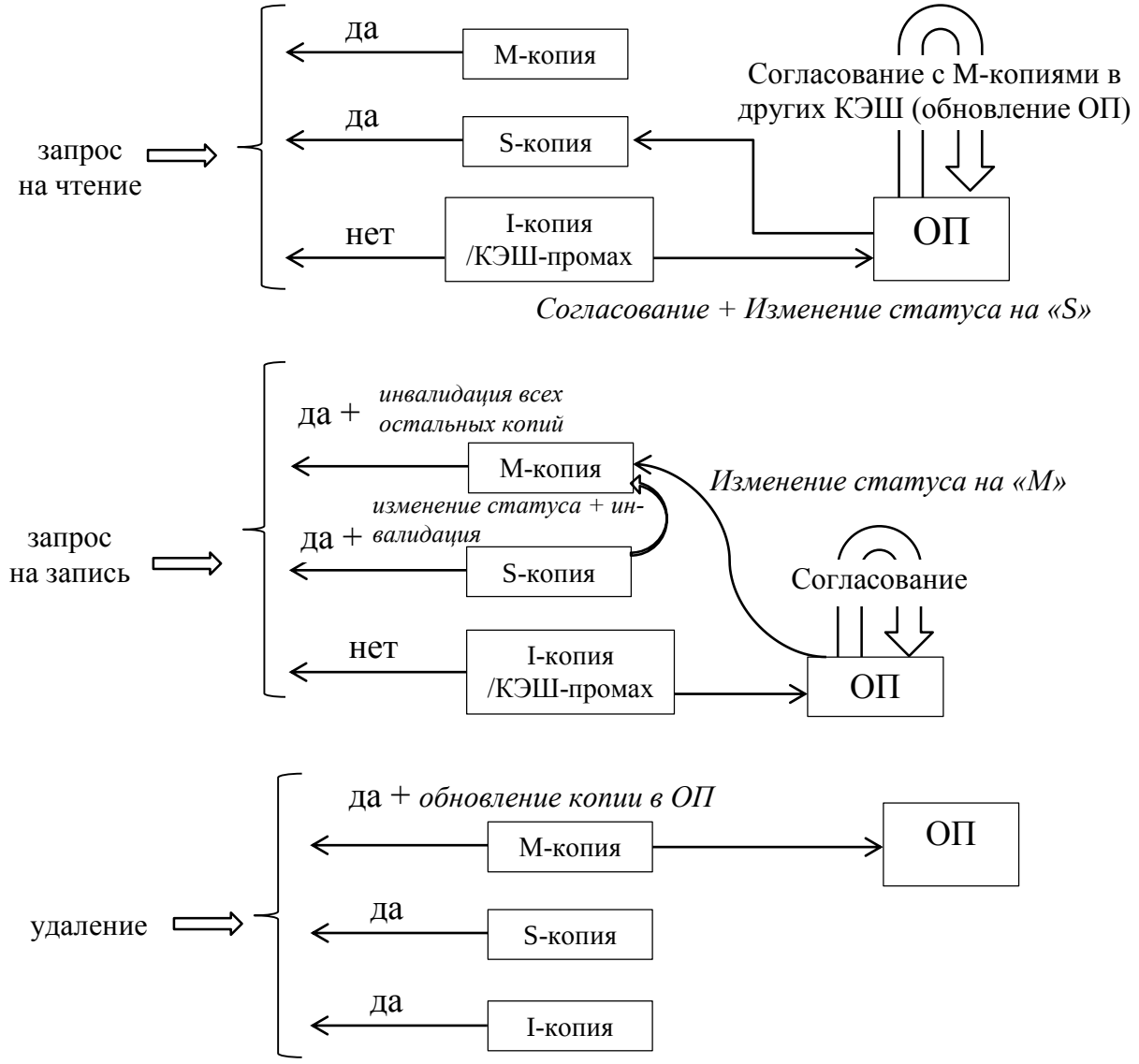


Рис. 9. Механизм MSI-согласования

Shared - этот блок совпадает с оригиналом из ОП, достоверен и существует, по крайней мере, в одной КЭШ-памяти. Т.к. копия достоверна, то операции чтения и записи разрешены. Операция ЧТ не меняет статус блока, операция ЗП меняет статус блока на «М». Его удаление из КЭШ не повлечёт никаких действий с оригиналом.

Invalid - этот блок является недостоверным (может совпадать или не совпадать с оригиналом в ОП – в зависимости от последних операций), может присутствовать в одном или нескольких процессорах/ядрах. Операции чтения/записи для «I»-блока запрещены и разрешаются только после выполнения согласования копий и повторного копирования оригинала из ОП в КЭШ-приёмник (запросивший копию). Такое чтение приводит к изменению статуса с «I» на «S». Само согласование может оказаться непростым. В этом случае анализируются остальные копии этого блока в других КЭШах. Если у кого-ни-

будь имеется измененная копия, то она должна быть скопирована в ОП, её статус у КЭШа-источника меняется на «S». И только после этого разрешается копировать оригинал в КЭШ-приёмник. Т.о. после операции согласования в системе невозможно наличие «M»-копий. Удаление «I»-копии разрешено без изменения оригинала в ОП.

Табл.2

	M	S	I
M	✗	✗	✓
S	✗	✓	✓
I	✓	✓	✓

Если требуется выполнить операцию для КЭШа, в котором искомый блок отсутствует, то выполняются те же действия, что и при наличие «I»-копии. Фактически, для любой заданной пары КЭШей, разрешены следующие состояния определенной копии (табл.2):

Все действия с копиями показаны на рис.9. Алгоритм MSI-протокола показан на рис.13.

MESI протокол

MESI: **M**odified – **E**xclusive – **S**hared - **I**nvalid.

Статусы «M», «S» и «I» аналогичны протоколу MSI.

Использующий MESI протокол процессор, четко различает КЭШ-строки, которых заведомо нет в КЭШах других процессоров (они помечаются как Exclusive) и «общие», присутствующие более чем в одном КЭШе (они помечаются как Shared). Эти копии достоверны и совпадают с оригиналом в ОП.

Операция чтения разрешена для достоверных копий блока со статусами «E», «S» и «M». При этом статус копии не меняется. При КЭШ-промахе выполняется процедура согласования:

- просматриваются другие КЭШи
- если имеется «M»-копия, то она должна быть скопирована в ОП, а затем из ОП скопирована в КЭШ процессора/ядра, запросившего доступ к данным, обе копии в КЭШе-источнике и КЭШе-приёмнике изменяют статус на «S»
- если имеется «E» или «S»-копия, то её оригинал сразу копируется из ОП в запросивший КЭШ, все копии данного блока во всех КЭШах изменяют статус на «S»
- если копий данного блока нет в других КЭШах, то его оригинал сразу копируется из ОП в запросивший КЭШ и его статус устанавливается в «E»
- выполняется операция чтения без изменения статуса копии.

Операция записи разрешена для достоверных копий блока со статусами «E», «S» и «M». При этом статус копии меняется на «M», а для всех остальных копий в других КЭШ статус меняется на «I». При КЭШ-промахе выполняется процедура согласования, а затем операция записи и изменение статусов копий. Удобство MESI-протокола состоит в том, что при изменении E-копии (а вероятность этого события очень велика из-за линейности обращений), нет необходимости делать достаточно трудоемкую операцию рассылки уведомлений другим копиям (их просто нет) и статус блока просто изменяется на «M».

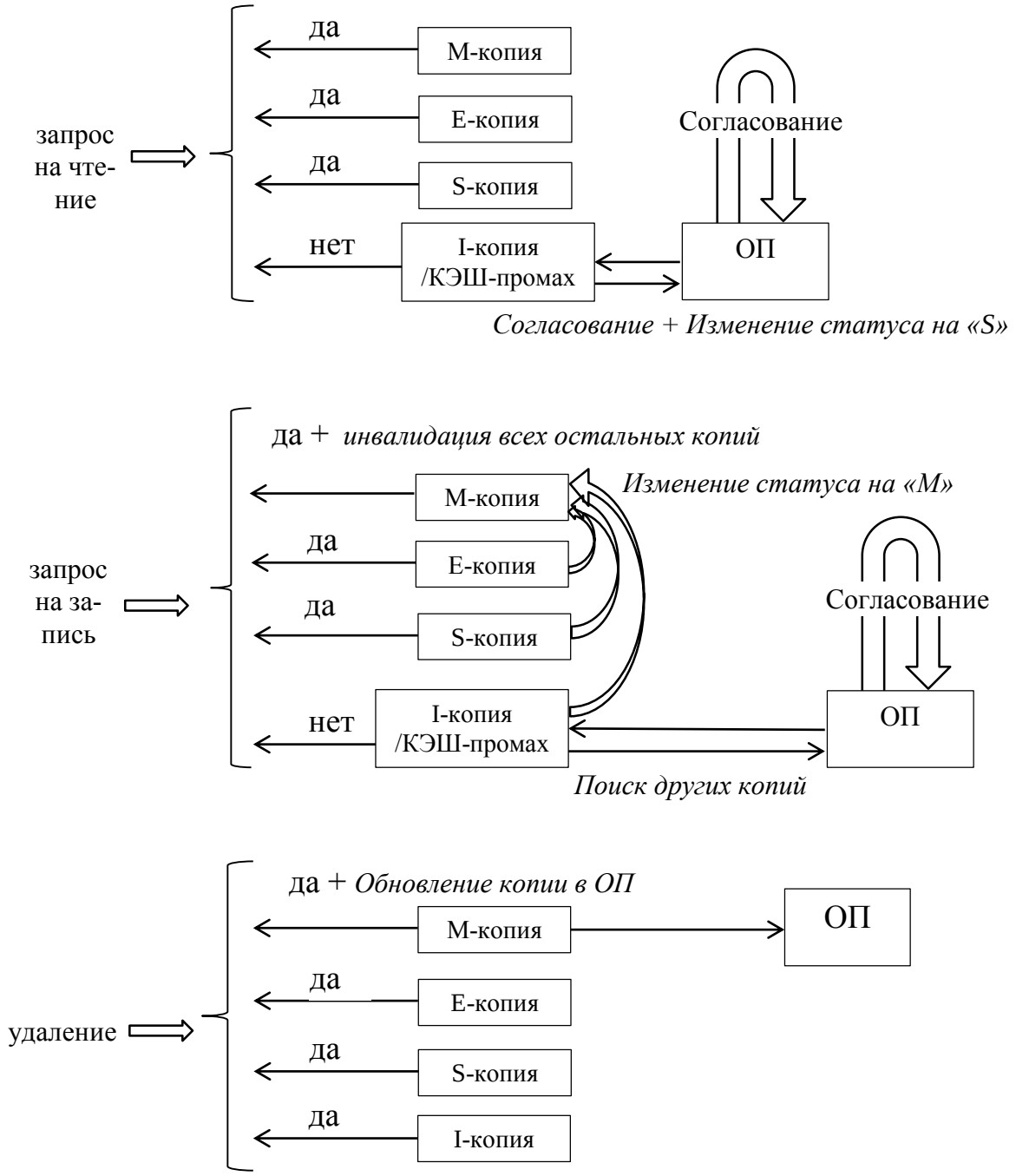


Рис. 10. Механизм MESI-согласования

Все процессоры (читай КЭШ-контроллер) бдительно наблюдают за всеми операциями чтения данных из памяти. И если один из процессоров замечает, что его сосед пытается прочитать строчку памяти, которая в его КЭШе помечена, как «Modified», то он прерывает эту операцию, сохраняет изменения в оперативной памяти, снимает со «своей» КЭШ-строки «отметку» Modified и только после этого разрешает вызвавшему «проблему» процессору завершить операцию чтения. В итоге «типовые» операции с КЭШ-памятью в MESI происходят практически с той же скоростью, что и в однопроцессорной системе.

Операция удаления разрешены для всех копий блока в любом статусе, разница лишь в том, что для М-копий содержимое блока копируется перед удалением в оригинал в ОП.

Если требуется выполнить операцию для КЭШа, в котором искомый блок отсутствует, то выполняются те же действия, что и при наличие «I»-копии. Фактически, для любой заданной пары КЭШей, разрешены следующие состояния определенной копии (табл.3):

Все действия с копиями показаны на рис.10. Алгоритм MESI-протокола показан на рис. 14.

Табл.3

	M	E	S	I
M	✗	✗	✗	✓
E	✗	✗	✗	✓
S	✗	✗	✓	✓
I	✓	✓	✓	✓

MOSI-протокол

Это расширение базового MSI протокола. Он добавляет статус **O**wned (в собственности), который указывает на то, что O-копия блока, принадлежащая текущему процессору, будет обслуживать запросы от других процессоров к этому блоку (вместо ОП). Для эффективности данного механизма необходимо наличие непосредственных связей между локальными КЭШами (см. рис.5, в), тогда как MSI- и MESI-протоколы разрабатывались в отсутствии таких связей (см. рис. 5, б).

Owned (в собственности) – одна из нескольких общих S-копий, но с исключительным правом раздавать копии вместо ОП. Копия достоверна, не совпадает с ОП и присутствует только в одном процессоре/ядре. В статус «O» может перейти только «M»-копия после появления запроса на чтение от других процессоров/ядер. КЭШ-владелец O-копии должен отвечать на все ЧТ-запросы того же блока от других процессоров/ядер. Это позволяет перемещать измененные копии в различные КЭШи без обновления ОП. Статус O-копии может быть изменен только на «I» при выполнении операции записи другим ядром/процессоров. Во всех остальных случаях статус сохраняется пожизненно (пока данная копия не будет удалена из КЭШ).

Shared (общая) копия – одна из нескольких одинаковых достоверных копий в системе, но в отличие от протокола MSI/MESI «S»-копия может отличаться от копии в ОП и при наличии «O»-копии всегда совпадает с ней. Если ни один КЭШ не перевёл свою копию в O-статус, то актуальной является копия в ОП (с которым совпадают все S-копии).

Статусы «M» и «I» аналогичны протоколам MSI/MESI.

Операция чтения разрешена для достоверных копий блока со статусами «O», «S» и «M». При этом статус копии не меняется. При КЭШ-промахе выполняется процедура согласования:

- просматриваются другие КЭШи
- если имеется «O»-копия, то её содержимое копируется в КЭШ процессора/ядра, запросившего доступ к данным,
- если «O»-копии нет, но имеется «M»-копия, то её статус меняется на «O», и её содержимое из КЭШа-источника копируется в запросивший КЭШ,
- если «O»-копии нет, «M»-копии нет, но имеется «S»-копия, то её содержимое из КЭШа-источника копируется в запросивший КЭШ (если «S»-копий несколько, то копируется, та, что ближе по коммуникационной сети, соединяющей КЭШи),

- если копий нет, то в КЭШ-приёмник копируется оригинал из ОП,
- копия в КЭШе-приёмнике получает статус «S»,
- выполняется операция чтения без изменения статуса копии.

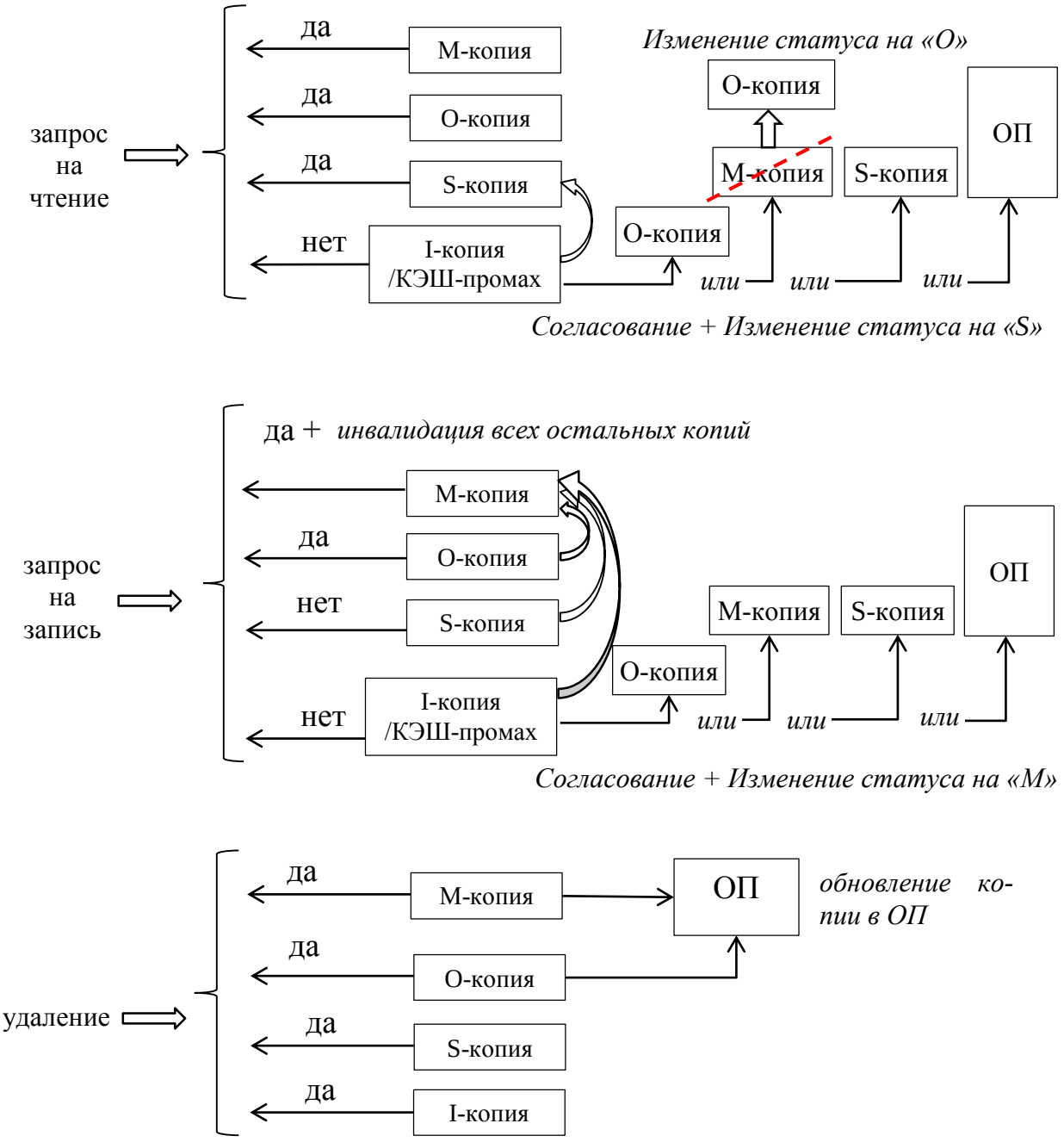


Рис. 11. Механизм MOSI-согласования

Операция записи разрешена для достоверных копий блока со статусами «O», «S» и «M». При этом статус копии меняется на «M», а для всех остальных копий в других КЭШ статус меняется на «I». При КЭШ-промахе выполняется процедура согласования, а затем операция записи и изменение статусов копий. Достоинство MOSI-протокола состоит в его скорости выполнения (обмен данными напрямую без использования оригинала в ОП.

Операция удаления разрешены для всех копий блока в

любом статусе, разница лишь в том, что для «M»- и «O»-копий содержимое блока копируется перед удалением в оригинал в ОП.

Для любой заданной пары КЭШей, разрешены следующие состояния определенной копии (табл.4):

Все действия с копиями показаны на рис.11. Алгоритм MOSI-протокола показан на рис. 15.

Табл.4

	M	O	S	I
M	✗	✗	✗	✓
O	✗	✗	✓	✓
S	✗	✓	✓	✓
I	✓	✓	✓	✓

Протокол MOESI

Является комбинацией MOSI и MESI, сочетающей достоинства обоих протоколов: высокую скорость, малое число обращений в ОП, исключение лишних проверок за счет разделения ситуации одной и нескольких копий.

Статусы «M», «O», «S» и «I» аналогичны протоколу MOSI, а статус «E» аналогичен протоколу MESI.

Табл.5

	M	O	E	S	I
M	✗	✗	✗	✗	✓
O	✗	✗	✗	✓	✓
E	✗	✗	✗	✗	✓
S	✗	✓	✗	✓	✓
I	✓	✓	✓	✓	✓

В ВС, реализующих протокол MOESI, требуется наличие высокоскоростных соединений между локальными КЭШами процессоров (предложено AMD в отличие от ранних моделей Intel, где есть связь КЭШ LL только с ОП). Т.о. процессоры могут эффективно использовать данные из КЭШ-памяти друг друга. В MESI процессоры почти не используют КЭШи своих соседей: в лучшем случае, чтение обновленных данных из памяти производится при выгрузке этих данных в память «соседом».

Для любой заданной пары КЭШей, разрешены следующие состояния определенной копии (табл.5):

MESIF протокол

Это протокол поддержания когерентности, разработанный позднее компанией Intel для CC-NUMA архитектур. Протокол состоит из пяти статусов копий: **M, E, S** и **I,** такие же, как и в MESI, а “F”-статус – это специализированная форма “S”- статуса, аналог “O”-статуса из MOESI, и указывает на то, что КЭШ должен выступать в качестве ответчика на любые запросы по данной строке.

Табл.6

	M	E	S	I	F
M	✗	✗	✗	✓	✗
E	✗	✗	✗	✓	✗
S	✗	✗	✓	✓	✓
I	✓	✓	✓	✓	✓
F	✗	✗	✓	✓	✗

Для любой заданной пары КЭШей, разрешены следующие состояния определенной копии (табл.6*):

** - если переставить соответствующие строки и столбцы в табл.5, то можно получить табл.6.*

3. МОДЕЛИРУЮЩАЯ ПРОГРАММА

В лабораторной работе моделируется четырёх-ядерная (четырёх-процессорная) система, каждый из процессоров имеет свой КЭШ. В системе присутствует одна общая оперативная память, которая условно состоит из 12 блоков.

Каждый КЭШ содержит 3 строки (одна строка может хранить в себе копию одного блока ОП). Удаление из КЭШ-памяти по переполнению осуществляется по принципу очереди – FIFO (First Input First Output). Существуют также принцип LRU (Least recently used) – удаляется строка, к которой давно не было обращений, LFU (Least Frequently Used) – удаляется строка, к которой меньше всего обращались, RND – удаляется случайная строка, или комбинированный механизм (LRU+LFU)…

С помощью программы можно смоделировать работу 4 протоколов: MSI, MOSI, MESI, MOESI/MESIF. Читая строки из ОП в КЭШ, модифицируя и удаляя их из КЭШ, ведётся постоянный контроль за состоянием локальных КЭШ каждого процессора и согласованием данных, расположенных в них, с данными общей ОП. Статусы строк всё время меняются по ситуации согласно алгоритму выбранного протокола. Блок схемы моделирующих алгоритмов приведены на рис. 13-16.

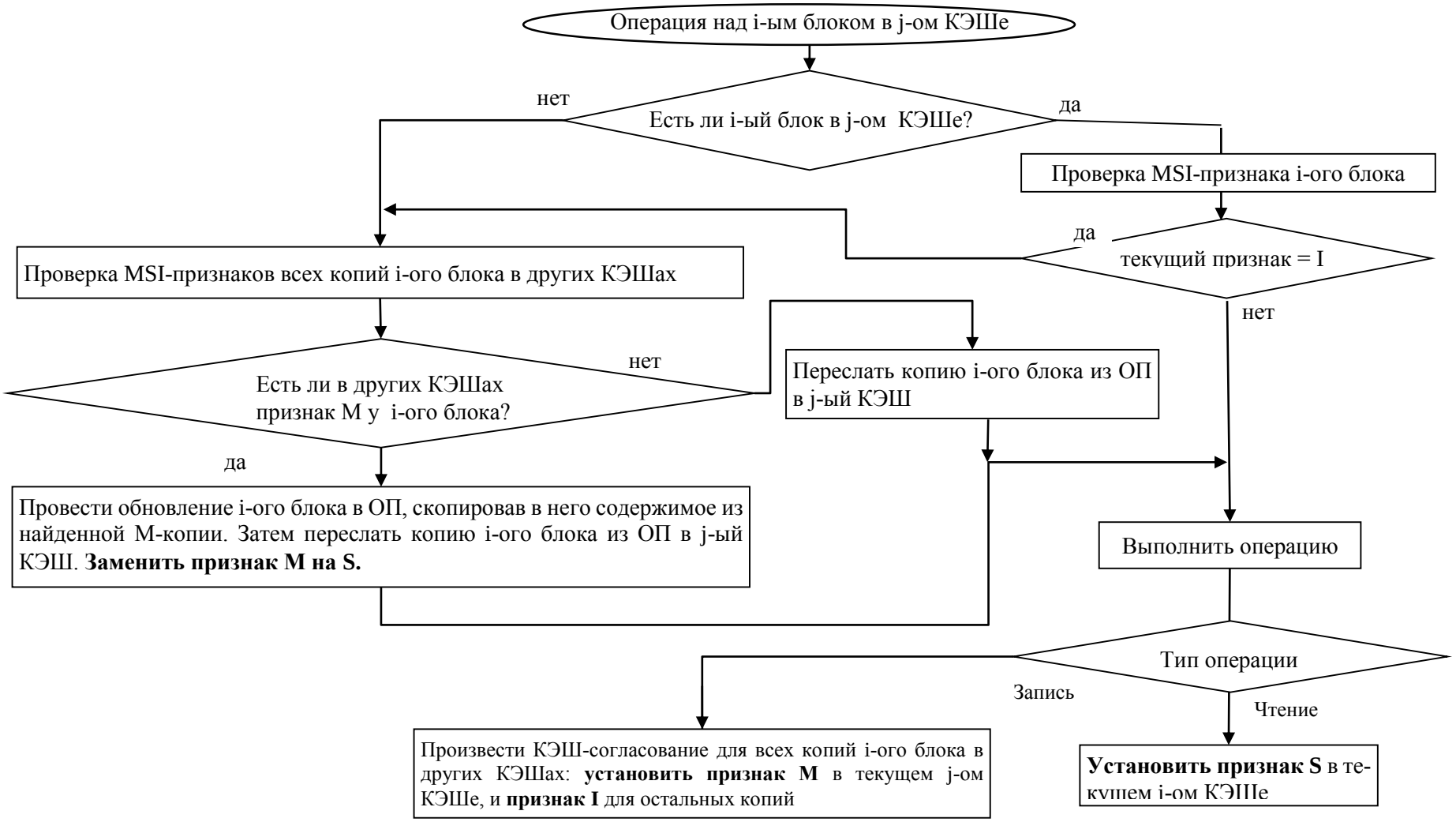


Рис.13. Алгоритм работы MSI-протокола

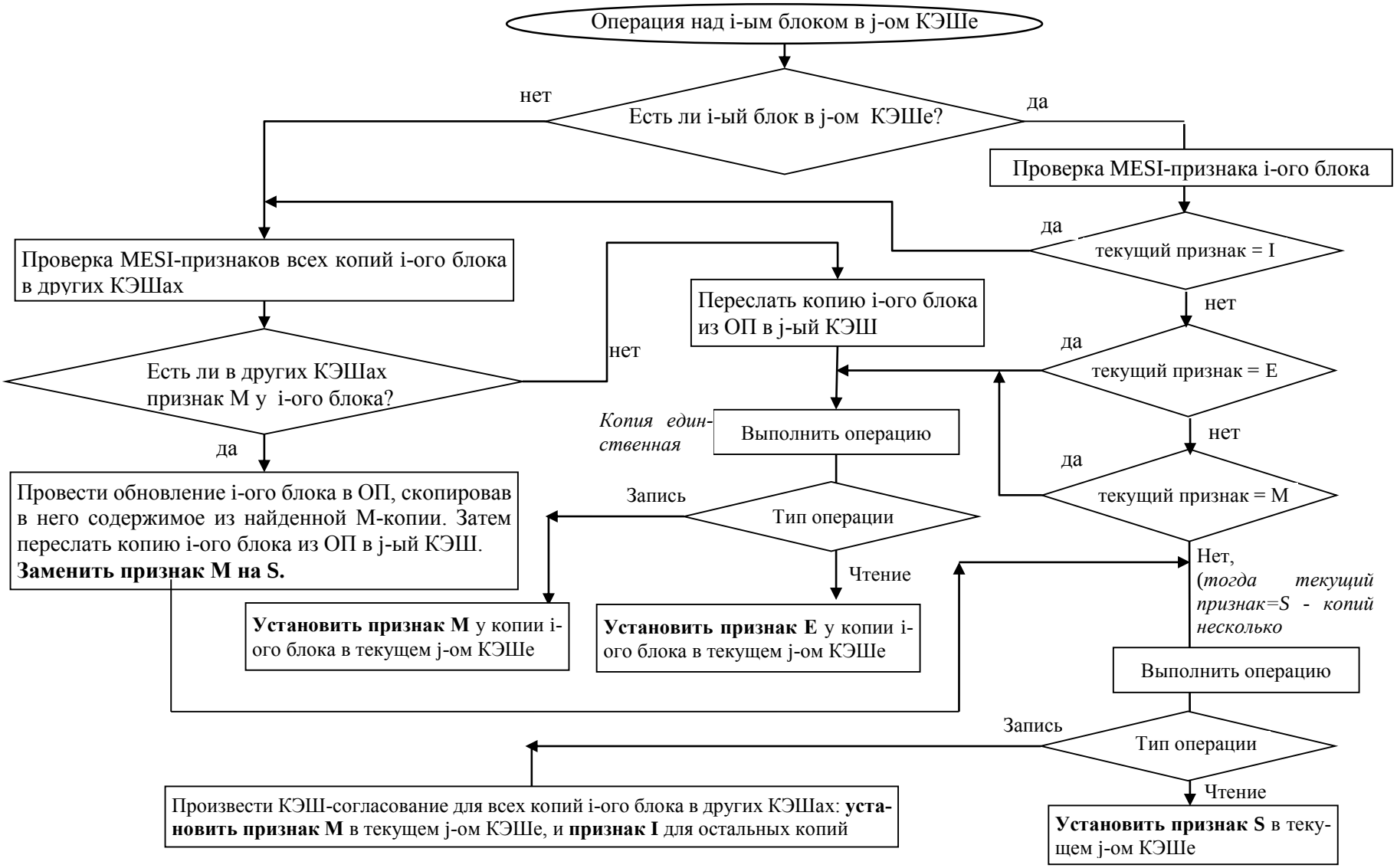


Рис.14. Алгоритм работы MESI-протокола

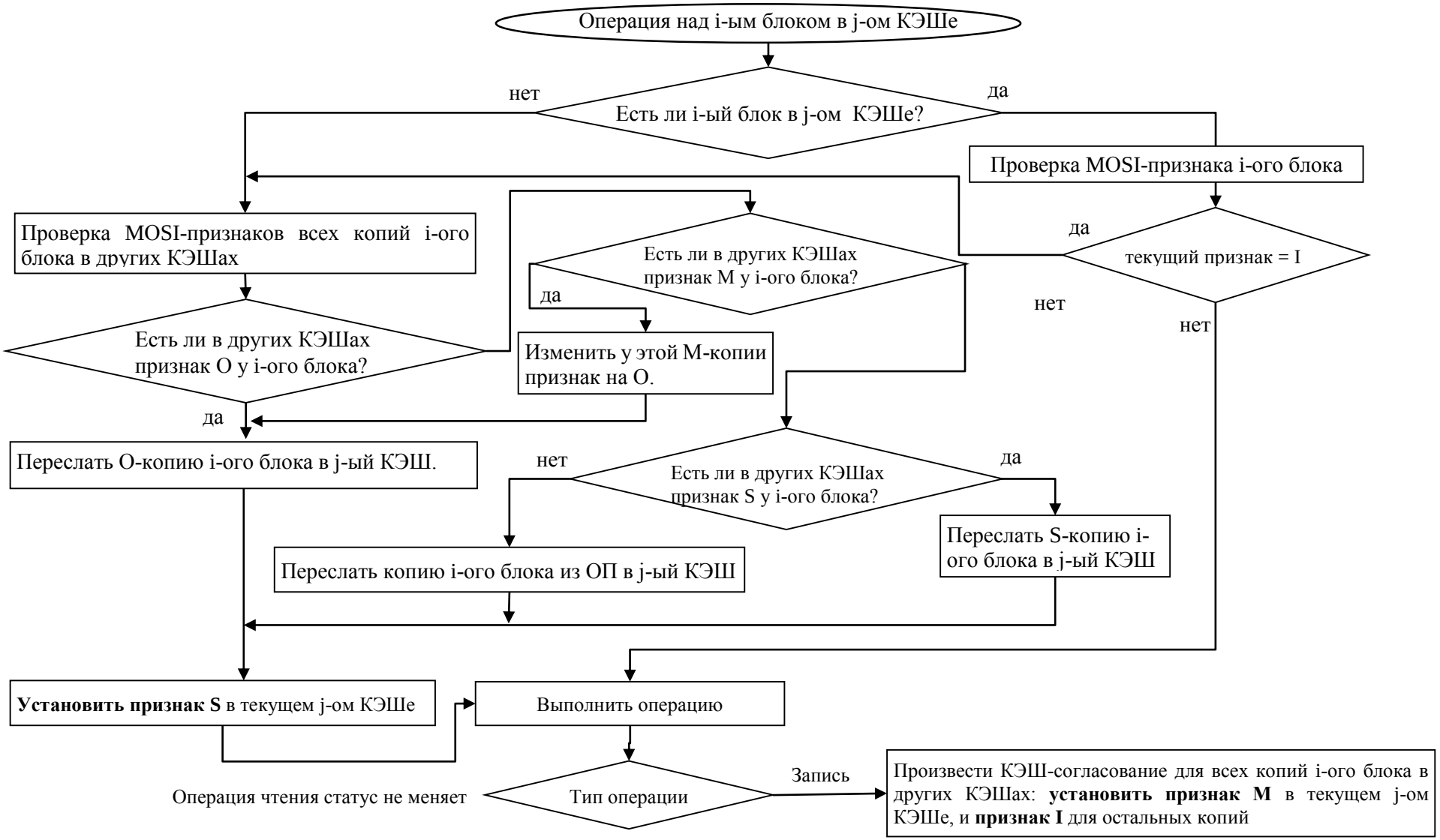


Рис.15. Алгоритм работы MOSI-протокола

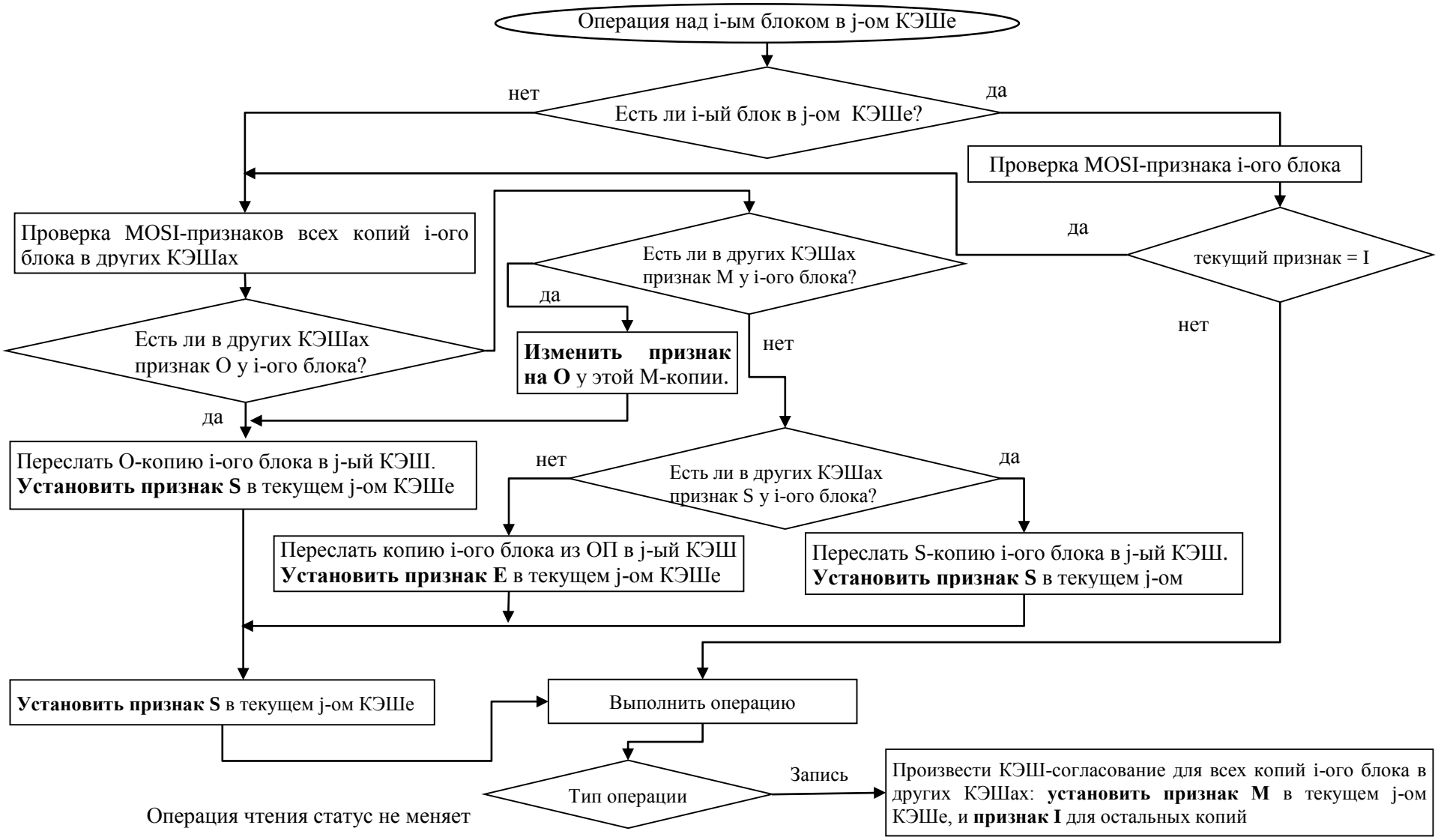


Рис.16. Алгоритм работы MOESI-протокола

Для запуска приложения необходима java-машина версии не ниже 1.8u40. Интерфейс программы показан на рис. 17.

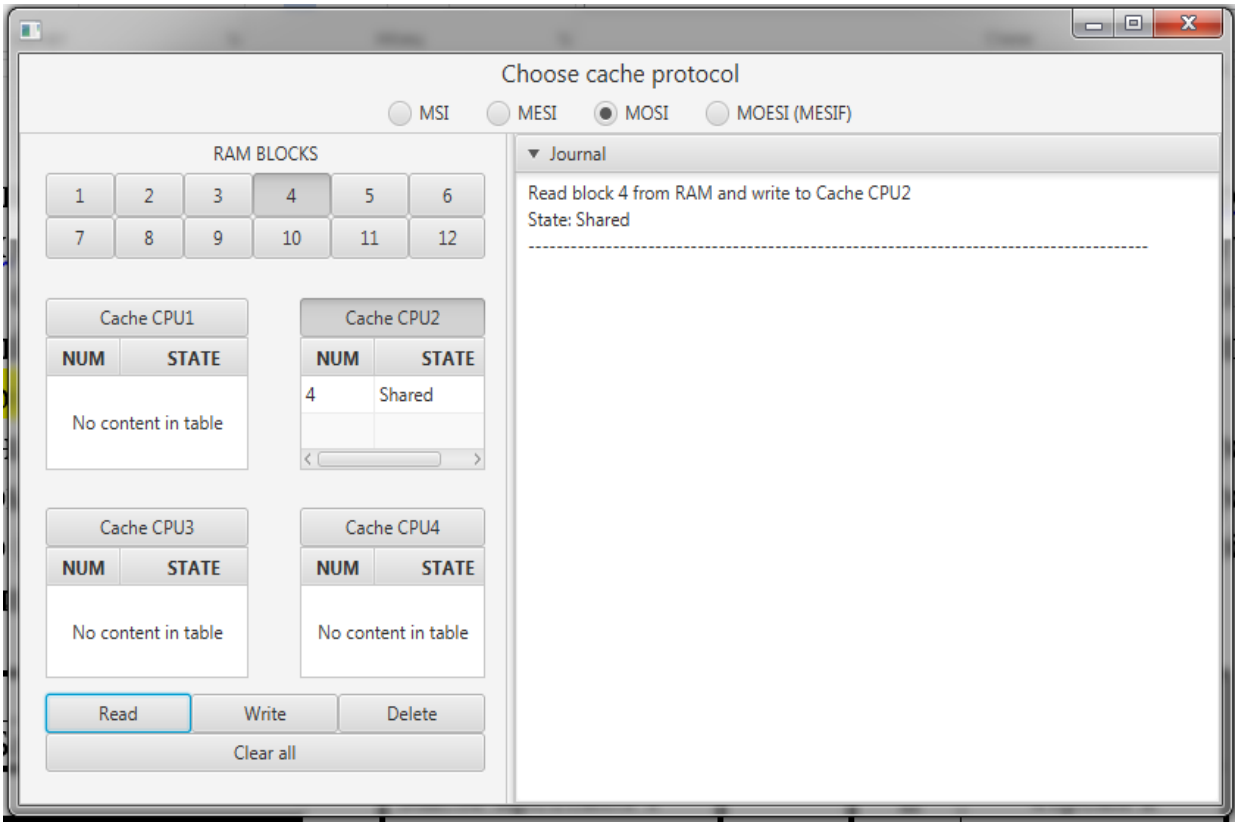


Рис.17. Интерфейс программы

В окне, появляющемся в Windows после запуска программы, есть следующие основные поля.

Вверху по центру расположены 4 переключателя для выбора протокола: MSI, MESI, MOSI, MOESI(MESIF). Кликнув мышкой один из переключателей вы выбираете указанный протокол.

Справа вверху расположены 12 кнопок, соответствующие 12 блокам ОП. однократное нажатие на одну из кнопок соответствует выбору указанного блока для выполнения операции с ним.

Ниже расположены 4 окна, соответствующие четырём процессрам ВС (CPU1÷CPU4), показывающие содержимое КЭШ-памяти каждого процессора согласно выбранному протоколу. Окно разделено на два столбца: правый – (номер блока ОП, хранимого в данной строке КЭШ) и левый – статус/признак каждого блока. Объём каждого КЭШа ограничен 3 строками. Если возникает необходимость записать четвертый блок в КЭШ, то предварительно следует удалить какую-то строку. Удаление из КЭШ-памяти по переполнению осуществляется по принципу LRU (Least recently used).

Внизу справа расположены 4 кнопки выбора операции. Возможны следующие действия процессора/ВС с блоками памяти:

- «Read» – операция считывания процессором блока данных из памяти;
- «Write» – операция записи процессором блока данных в память;

- «Delete» – удаление выбранного блока из КЭШа выбранного процессора;
- «Clear all» – очистка содержимого всех КЭШей.

Одно из этих действий может быть выбрано нажатием на соответствующую кнопку. Выполняемая операция «Read/Write/Delete/Clear all» выбирается последней после чего программа выполняет выбранное действие с выбранным блоком в выбранном КЭШе согласно выбранному протоколу. Результат можно увидеть по изменению статусов копий блоков в КЭШах, и дополнительно в окне справа выводиться краткое описание выполненных дкействий.

Выполнить работу можно очно – выполняя изменения и поясняя их преподавателю. Либо заочно, выполнив работу дома и составив отчёт о проведённых изменениях с пояснениями выполненных действий (операций чтения/записи копий) и объяснением причин изменений в статусах копий (см. примеры ниже), отчет следует отправить преподавателю на проверку. После верного выполнения работы следует её защита – ответы на теоретические вопросы из раздела 5.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

В работе требуется проследить согласно заданному варианту за изменениями в статусах/признаках строк каждого КЭШ, составить отчет, в который включить описание этих изменений и объяснение их причин. Защитить работу, ответив на вопросы по ходу выполнения моделирования и по теории (см. раздел 5). Вариант задания выбрать из табл.7. Если выполнение работы происходит очно в ДК одновременно с защитой, то писать отчет не обязательно.

Табл.7. Варианты заданий

Протокол / Операции	MSI	MESI	MOSI	MOESI (MESIF)
П1Ч1, П2Ч1, П3З1, П1У1, П4Ч1, П4Ч5, П4З2, П4Ч6, П4Ч2, П2Ч1, П3У1	1	2	3	4
П3Ч5, П2З5, П2З1, П2Ч8, П2Ч3, П4Ч1, П2У1, П3Ч1	5	6	7	8
П2З12, П2Ч7, П2Ч11, П2З5, П2Ч5, П1З5, П1У5, П2У5, П4Ч7	9	10	11	12
П4З1, П4Ч9, П4З12, П1З1, П4Ч5, П2Ч12, П4У12, П3Ч12	13	14	15	16
П4Ч8, П2З8, П1Ч8, П4Ч8, П4Ч3, П4З1, П1У8	17	18	19	20
П3Ч11, П2Ч11, П4Ч11, П1З1, П2Ч1, П2У1, П1У1, П3Ч6, П3Ч2, П3З7	21	22	23	24
П2Ч3, П2З3, П2Ч3, П3Ч3, П3З3, П2У3, П4Ч3, П3Ч4, П3З6, П3Ч5	25	26	27	28

Запись **П1Ч10** – означает «**П**роцессор **1** выполнил **Ч**тение ***из*** блока **10**», по аналогии **З5** – **З**апись ***в*** блок **5**, **У8** – **У**даление ***из КЭШ*** блока **8**.

Вы можете проделать также любые другие операции (кроме заданных в табл.7) с другим протоколом, необходимые для лучшего понимания принципов работы.

4.1. Тестовые примеры моделирования (описание в отчете)

Тест 1 **Выбран MESI-протокол.**

Исходное состояние: ни в один кэш не записаны блоки ОП, следователь-но, и MESI признаки пусты: КЭШ1:
КЭШ2:
КЭШ3:
КЭШ4:

Операция: Первый процессор произвёл чтение из 1-го блока ОП.

Действия: Т.к. в собственном КЭШе1 блок №1 отсутствует, значит его следует прочитать из ОП. Согласно MESI-протоколу проверяем другие КЭШи, в них указанный блок отсутствует, значит копируем блок сразу из ОП без КЭШ-согласования. После выполнения операции чтения данных содержимое копии блока1 в КЭШ1 не меняется, значит он единственный и совпадает с блоком ОП – ему присваивается признак E.

Результат: КЭШ1: **E1**
КЭШ2:
КЭШ3:
КЭШ4:

Тест 2

Исходное состояние: КЭШ1: E3, M1, E5
КЭШ2: E2, E12
КЭШ3:
КЭШ4:

Операция: Первый процессор произвёл чтение 2-го блока ОП.

Действия: Т.к. в собственном КЭШе1 блок №2 отсутствует, значит его следует прочитать из ОП. Согласно MESI-протоколу проверяем другие КЭШи, и находим достоверную копию в КЭШ2. Статус копии E, значит её содержимое полностью совпадает с ОП, значит копируем блок сразу из ОП без КЭШ-согласования. После выполнения операции чтения данных содержимое копии блока2 в КЭШ1 не меняется, значит он совпадает с блоком ОП, но такая же совпадающая копия имеется в КЭШ2. Несколько одинаковых копий должны быть помечены признаком S каждая.

Результат: КЭШ1: E3, M1, E5, **S2**
КЭШ2: **S2**, E12
КЭШ3:
КЭШ4:

Тест 3

Исходное состояние: КЭШ1: E3, M1, E5, S2
КЭШ2: S2, E12
КЭШ3:
КЭШ4:

Операция: Третий процессор произвёл запись в 1-й блок ОП.

Действия: Т.к. в собственном КЭШе3 блок №1 отсутствует, значит его следует прочитать из ОП. Согласно MESI-протоколу проверяем другие КЭШи, и

находим достоверную копию в КЭШ1. Статус копии М, значит её содержимое достоверно, но отличается от содержимого блока1 в ОП. Верна копия в КЭШ1. Следовательно выполняется протокол КЭШ-согласования: достоверную копия блока1 из КЭШ1 заносим в ОП (в этот момент копия и оригинал становятся одинаковыми). После этого копируем блок из ОП в КЭШ3. После выполнения операции записи данных в блок1 процессором3 содержимое копии блока1 в КЭШ3 меняется, но содержит достоверные данные – признак М. При этом содержимое всех остальных копий (КЭШ1) и оригинала в ОП становиться не достоверным – признаки I.

Результат: КЭШ1: E3, **I1**, E5, S2
КЭШ2: S2, E12
КЭШ3: **M1**
КЭШ4:

Тест 4

Исходное состояние: КЭШ1: E3, I1, E5, S2
КЭШ2: S2, E12
КЭШ3: M1
КЭШ4:

Операция: Первый процессор произвёл чтение из 1-го блока ОП.

Действия: Т.к. в собственном КЭШе1 блок №1 присутствует, проверяем его статус (I1- недостоверный), значит его следует заново прочитать из ОП. Согласно MESI-протоколу проверяем другие КЭШи, и находим достоверную копию в КЭШ3. Статус копии М, значит её содержимое достоверно, но отличается от содержимого блока1 в ОП. Верна копия в КЭШ3. Следовательно выполняется протокол КЭШ-согласования: достоверную копия блока1 из КЭШ3 заносим в ОП (в этот момент копия и оригинал становятся одинаковыми). После этого копируем блок из ОП в КЭШ1. После выполнения операции чтения данных содержимое копии блока1 в КЭШ1 не меняется, значит он совпадает с блоком ОП, но такая же совпадающая копия имеется в КЭШ3. Несколько одинаковых копий должны быть помечены признаком S каждая.

Результат: КЭШ1: E3, **S1**, E5, S2
КЭШ2: S2, E12
КЭШ3: **S1**
КЭШ4:

Тест 5

Исходное состояние: КЭШ1: E3, M1, E5, S2
КЭШ2: S2, E12
КЭШ3: I1
КЭШ4:

Операция: Первый процессор произвёл удаление 2-го блока ОП.

Действия: Проверяем его статус (S2- достоверный, совпадает с ОП), значит заносить его содержимое в ОП при удалении не обязательно. Производим удаление блок и записи о нём. Согласно MESI-протоколу проверяем другие КЭШи, и находим достоверную копию в КЭШ2. Статус копии S, значит её содержимое достоверно и совпадает с ОП. Больше копий не обнаружено. Копия блока 2 в

КЭШ2 остаётся единственной (статус E).

Результат: КЭШ1: E3, M1, E5,
КЭШ2: **E2**, E12
КЭШ3: I1
КЭШ4:

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Поясните архитектуру МПВС? Какая ВС моделируется в работе?
2. Что такое оперативная память? В чём её назначение?
3. Что такое КЭШ-память?
4. Функции КЭШ?
5. За счёт чего повышается эффективность работы процессора с ОП при применении КЭШ?
6. Что такое КЭШ-попадание и КЭШ-промах? Какие операции из вашего варианта оказались КЭШ-попаданиями, а какие КЭШ-промахами?
7. Для чего требуется удалять строки из КЭШ?
8. Какие существуют стратегии обновления строк (блоков) ОП? Поясните по таблице 1.
9. Что такое сквозная запись?
10. Что такое обратная запись?
11. Почему при сквозной записи снижается эффективность КЭШ-памяти?
12. Расскажите о мультипроцессорной системе, рассмотренной в лабораторной работе.
13. Как можно организовать КЭШ-память в мультипроцессорной системе?
14. Для чего нужно производить КЭШ-согласование?
15. Что такое когерентность памяти?
16. Расскажите о методах поддержания когерентности памяти.
17. Дайте определение протокола MESI.
18. Опишите принцип работы протокола MESI.
19. Пояснить любую операцию (Ч, З, У) с любым блоком согласно любому протоколу на усмотрение преподавателя.

6. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Таненбаум, Э. Архитектура компьютера. СПб. Питер, 2014. - 811 с.
2. Исследование эффективности совместного использования общего и разделенного L2-кэша современных двухъядерных процессоров. http://www.ixbt.com/cpu/rmmt-l2-cache.shtml
3. Болотов П. Принципы работы кэш-памяти http://alasir.com/articles/cache_principles/cache_access_write_rus.html
4. Neupane, Mahesh (April 16, 2004). "Cache Coherence"
5. http://z52107.narod.ru/02_inf/01/05.html