

Организация когерентности многоуровневой иерархической памяти

Мультипроцессорная когерентность кэш-памяти

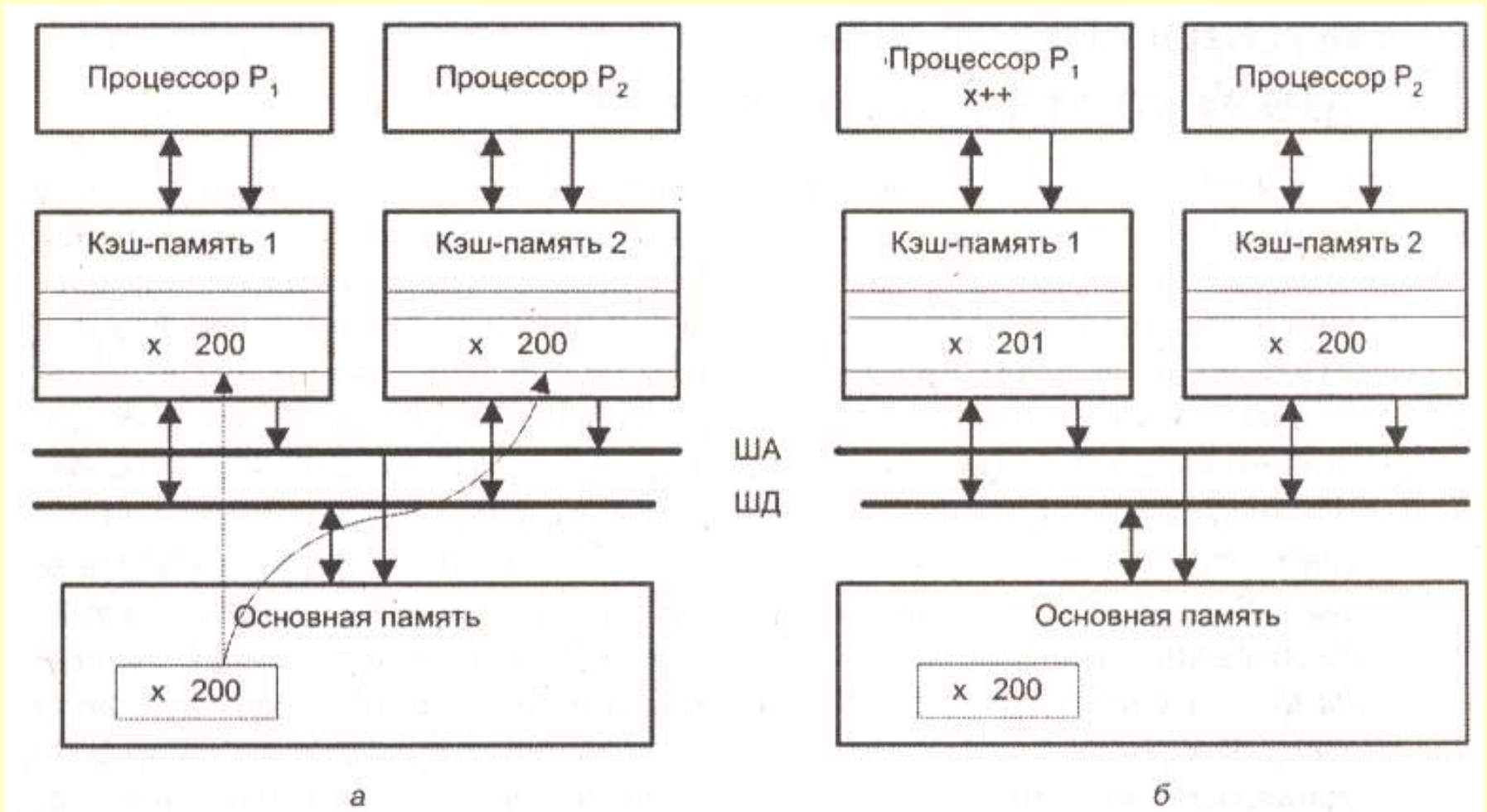
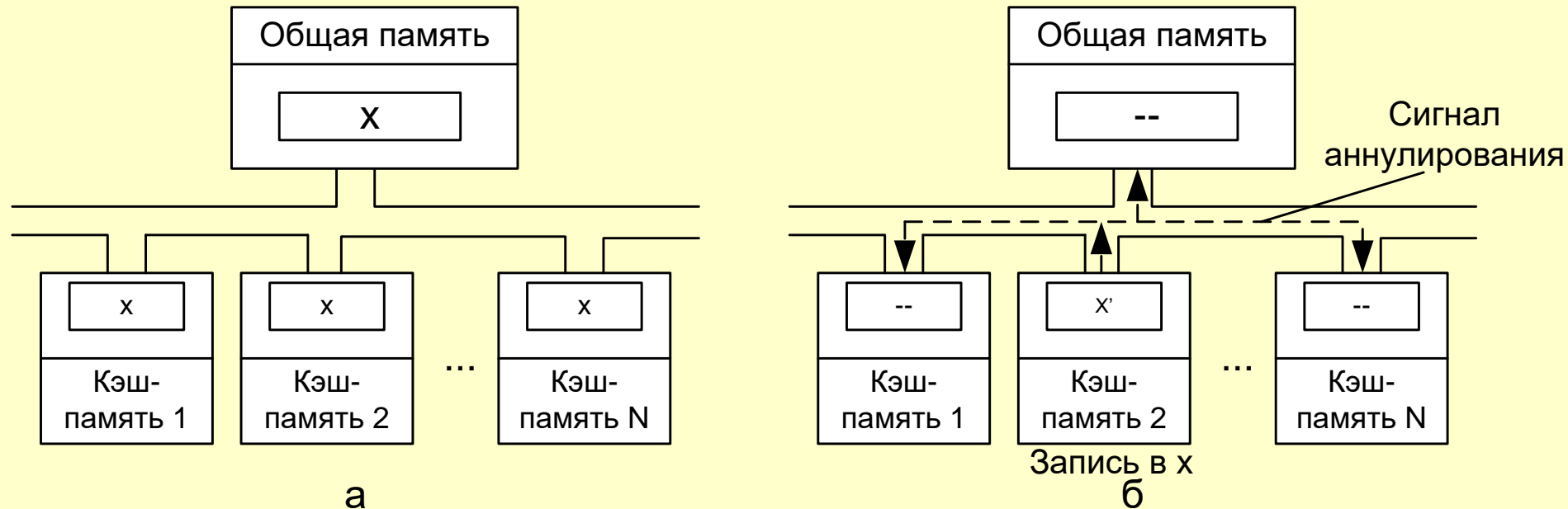


Иллюстрация проблемы когерентности памяти: а — содержимое памяти до изменения значения x ; б — после изменения

Аппаратные способы решения проблемы когерентности

1. Запись с аннулированием (write invalidate)

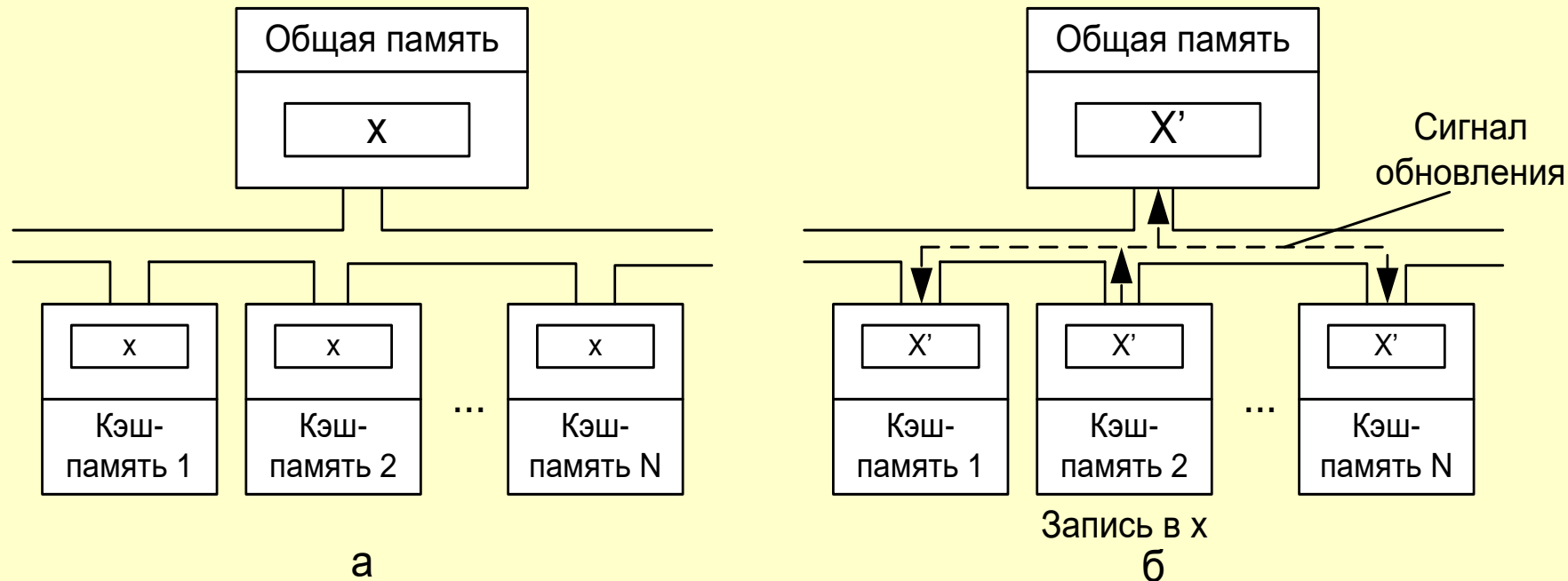


Запись с аннулированием: а - исходное состояние;

б - после изменения значения x в кэш-памяти 2

Аппаратные способы решения проблемы когерентности

2. Запись с обновлением (write update) или запись с трансляцией (write broadcast).



Запись с аннулированием: а - исходное состояние;
б - после изменения значения x в кэш-памяти 2

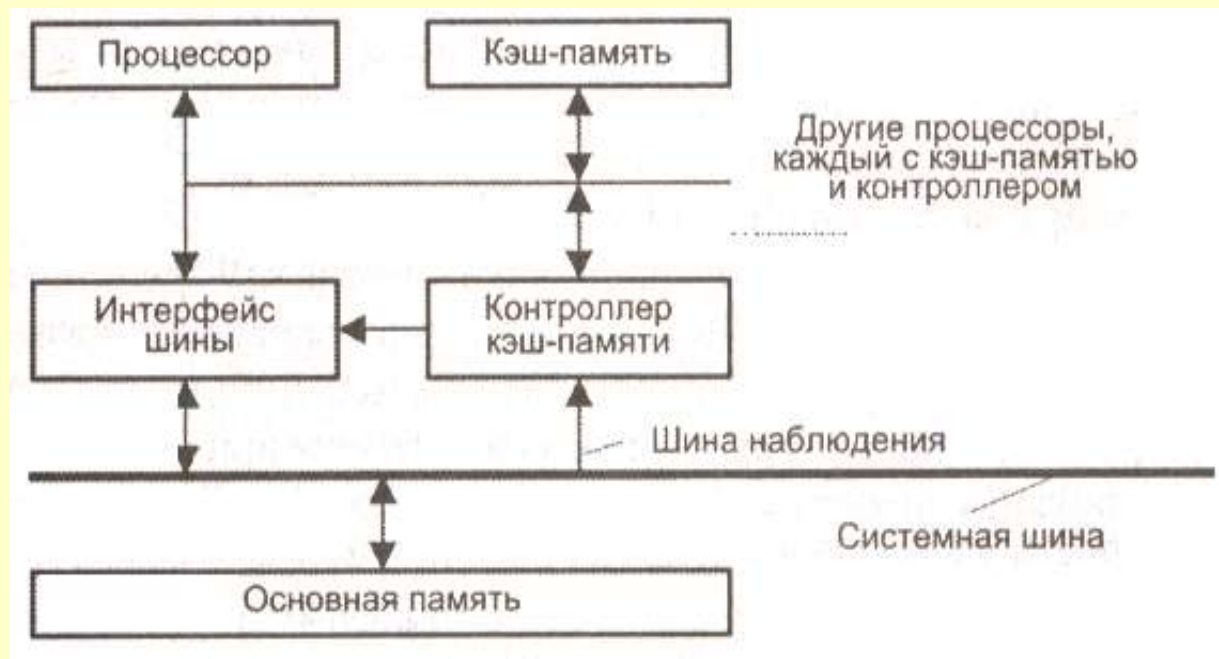
Способы обеспечения когерентности

В общем случае для поддержания когерентности в мультипроцессорных системах имеются следующие возможности:

- некешируемые данные
- совместно используемая кэш-память
- широковещательная запись
- протоколы наблюдения
- протоколы, на основе справочника

Протоколы наблюдения

В протоколах наблюдения (*snoopy protocols* или просто *snooping*) ответственность за поддержание когерентности всех кэшей многопроцессорной системы возлагается на контроллеры кэшей.



Протоколы наблюдения характерны для мультипроцессорных систем на базе общей шины, поскольку общая шина достаточно просто обеспечивает как наблюдение, так и широковещательную передачу сообщений.

Протоколы наблюдения

Наиболее распространенные протоколы наблюдения:

- **Протокол сквозной записи**
- **Протокол обратной записи**
- **Протокол однократной записи**
- **Протокол Synapse**
- **Протокол Berkeley**
- **Протокол Illinois**
- **Протокол Firefly**
- **Протокол Dragon**
- **Протокол MESI**

Протоколы наблюдения

В большинстве протоколов стратегия обеспечения когерентности кэш-памяти расценивается как смена состояний в конечном автомате.

При таком подходе предполагается, что любой блок в локальной кэш-памяти может находиться в одном из фиксированных состояний.

Обычно число таких состояний не превышает четырех, поэтому для каждой строки кэш-памяти в ее теге имеются два бита, называемые **битами состояния** (SB, Status Bit).

Следует также учитывать, что некоторым идентичным по смыслу состояниям строки кэша разработчиками различных протоколов присвоены разные наименования.

Например, состояние строки, в которой были произведены локальные изменения, в одних протоколах называют Dirty («грязный»), а в других — Modified («модифицированный» или «измененный»).

Протокол сквозной записи

- Протокол представляет собой расширение стандартной процедуры для SISD систем.
- Протокол считается простым, но при большом числе процессоров приводит к значительному трафику шины (даже при *записи с аннулированием*).
- Также производительность падает из-за простоя процессоров в ходе выполнения записи в ОП.

Протокол обратной записи

- Протокол представляет собой расширение стандартной процедуры для SISD систем.
- Отличие – блок модифицированных данных будет переписан в ОП при выполнении одного из двух условий:
 - блок удаляется из кэш-памяти, где он был изменён
 - другой процессор обратился к своей копии изменённого блока
- Протокол по эффективности превосходит схему сквозной записи (перезапись только изменённых блоков)
- Однако, изменения в блоке не всегда наблюдаемы со стороны других процессоров (до момента перезаписи блока в ОП)
- Berkeley – кэши, которые «собираются изменить» содержимое блока, должны получить эксклюзивные права на этот блок
- ОП не всегда содержит последнее значение элемента

Протокол MESI

Согласно протоколу MESI, каждая строка бывает в одном из четырех возможных состояний (в дальнейшем будем ссылаться на эти состояния с помощью букв M, E, S и I):

- * **Модифицированная (M, Modified)** — данные в кэш-строке, помеченной как M, были модифицированы, но измененная информация пока не переписана в основную память.

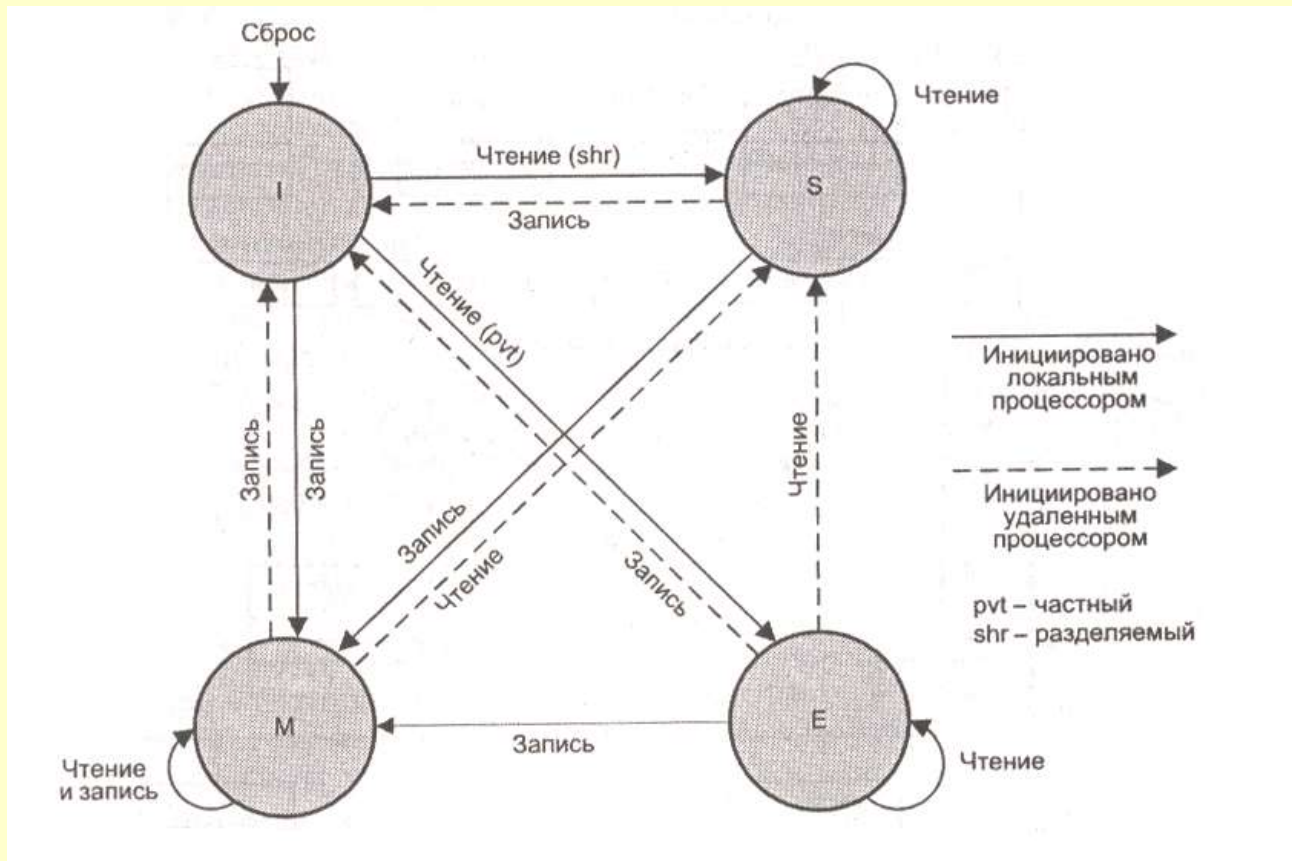
- * **Эксклюзивная (E, Exclusive)** — данная строка в кэше не подвергалась изменению посредством запроса на запись, совпадает с аналогичной строкой в основной памяти, но отсутствует в любом другом локальном кэше.

- * **Разделяемая (S, Shared)** — строка в кэше совпадает с аналогичной строкой в основной памяти (данные достоверны) и может присутствовать в одном или нескольких из прочих кэшей.

- * **Недействительная (I, Invalid)** — кэш-строка, помеченная как недействительная, не содержит достоверных данных и становится логически недоступной

Протокол MESI

Безусловно, среди известных протоколов наблюдения самым популярным является протокол MESI (Modified/Exclusive/Shared/Invalid). Протокол MESI широко распространен в коммерческих микропроцессорных системах, например на базе микропроцессоров Pentium и PowerPC.



Протокол MESI

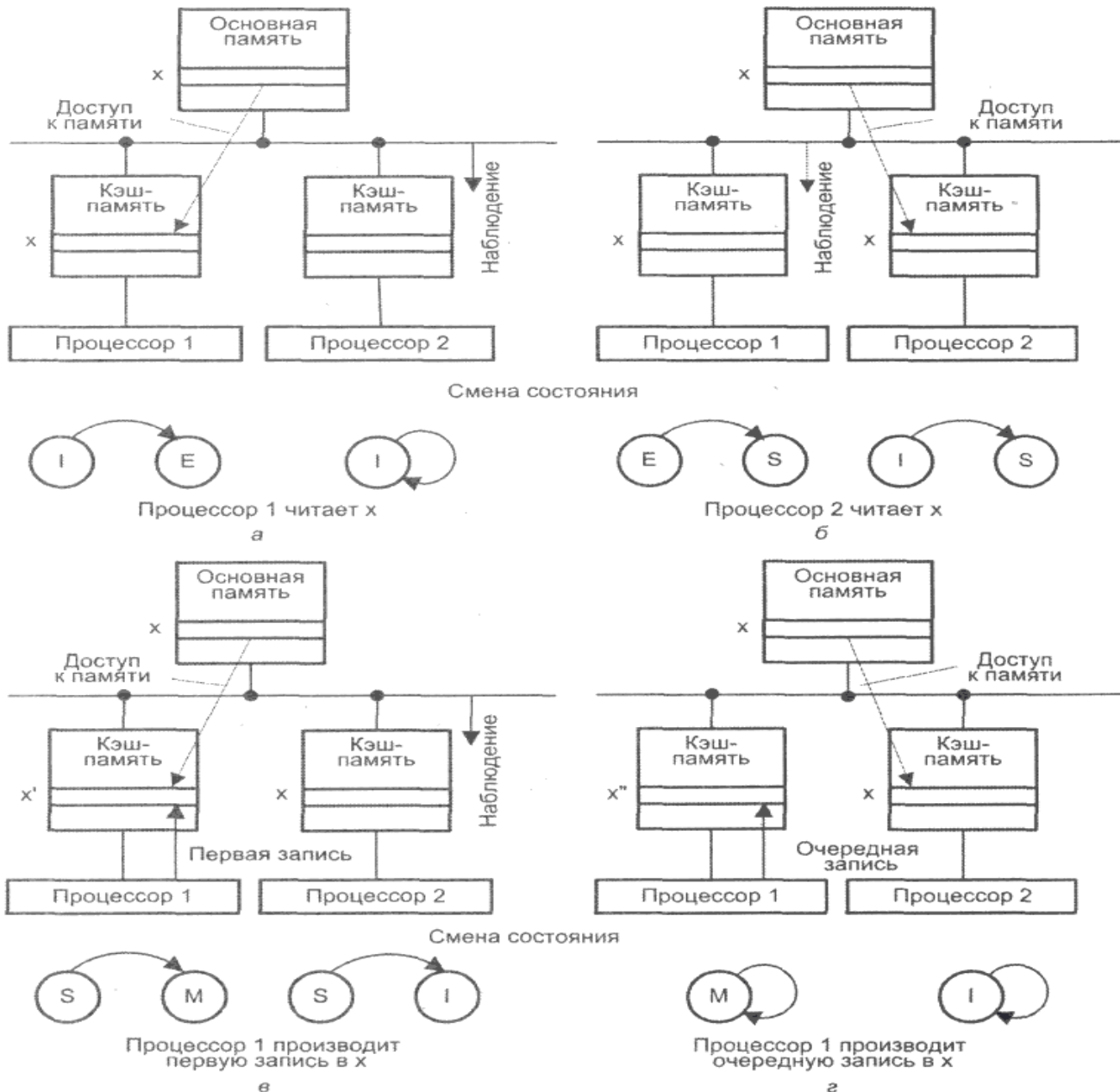
Запрос на **чтение из строки** (промах при чтении):

- строка считана из ОП (статус E)
- строка считана из другой кэш-памяти (статус S)

Запрос на **запись в строку** (промах при записи) – формируется последовательность операций RWITM (чтение с намерением модификации):

- копия в другой кэш-памяти не M – копия I, считывание из ОП
- копия в другой кэш-памяти M – прерывание RWITM, запись в ОП, копия I, возобновление RWITM (повторное чтение из ОП)

Протокол MESI



Протокол MESI

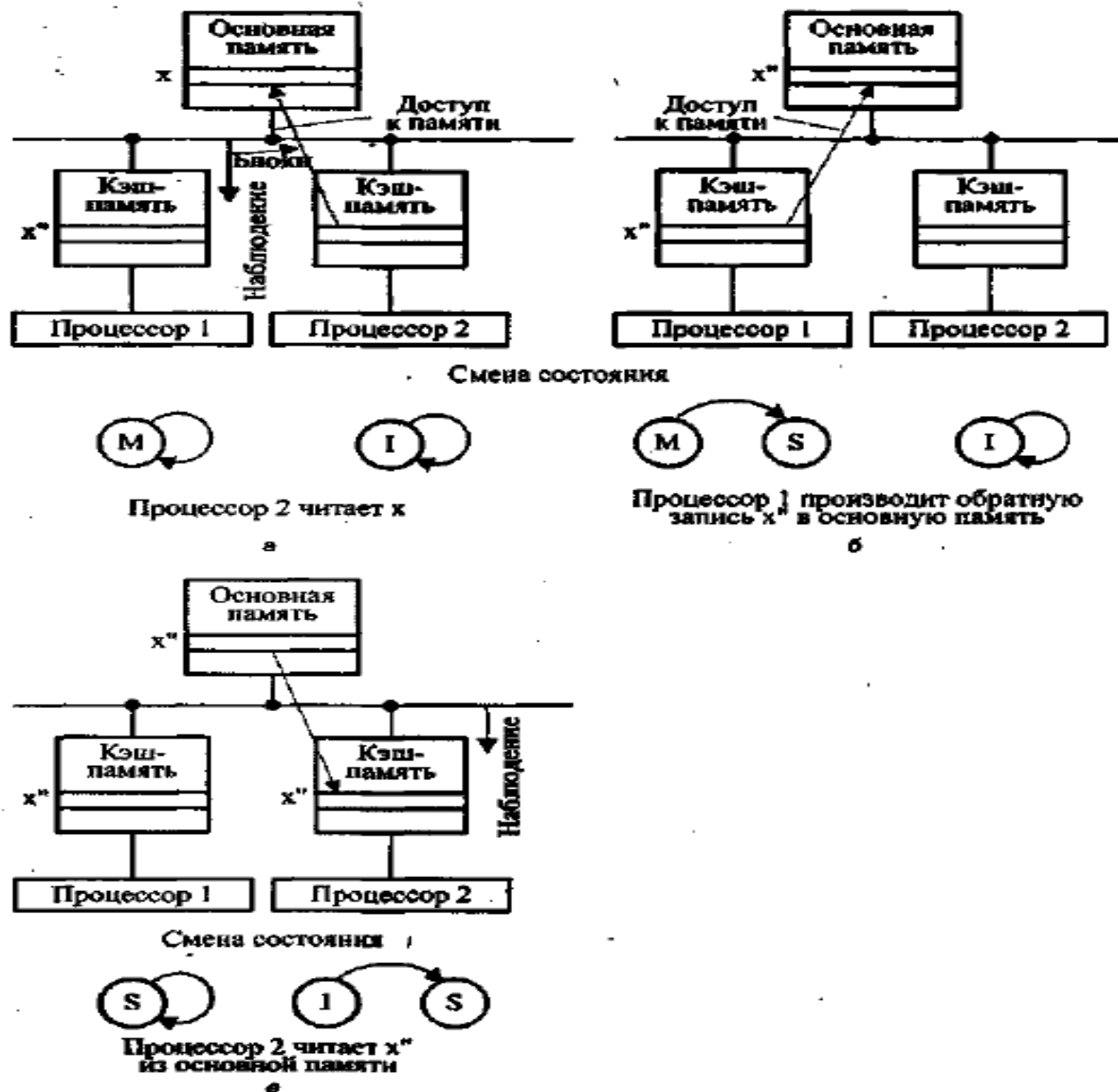


Рис. 11.16. Переход из состояния E в состояние S в протоколе MESI: а — процессор 2 читает x ; б — процессор 1 производит обратную запись x'' в основную память; в — процессор 2 читает x'' из основной памяти