Cache Coherency Protocols

Amirkabir University of Technology

Department of Computer Engineering

Dr. Farbeh

Department of
Computer Engineering

Amirkabir University of Technology
(Tehran Polytechnic)

01

Cache States

Cache states

M= *Modified* or **D**=*Dirty* or **DE**=*Dirty-Exclusive* or **EM**=*Exclusive Modified*

- modified in one cache only – write-back required at replacement.
- data is stored only in one cache but the data in memory is not updated (invalid, not clean).

O= *Owner* or **SD**=*Shared Dirty* or **SM**=*Shared Modified* or **T**=*Tagged*

- modified, potentially shared, owned, write-back required at replacement
- data may be stored in more than a cache but the data in memory is not updated (invalid, not clean). Only one cache is the "owner", other caches are set "Valid" (S/V/SC). On bus read request, the data is supplied by the "owner" instead of the memory.

E= *Exclusive* or **R**=*Reserved* or **VE**=*Valid-Exclusive* or **EC**=*Exclusive Clean* or **Me**=*Exclusive*

- clean, in one cache only
- Data is stored only in one cache and *clean* in memory

Cache states

S=*Shared* or **V**=*Valid* or **SC**=*Shared Clean*

- shared or valid
- Data potentially shared with other caches. The data can be clean or dirty. The term "clean" in SC is misleading because can be also dirty

I=*Invalid*

- Cache line invalid. If the cache line is not present (no tag matching) it is considered equivalent to invalid, therefore invalid data means data present but invalid or not present in cache

Cache states

F=*Forward* or **R**=*Recent*

- Additional states of MESI protocol
- Last data read. It is a special "Valid" state that is the "Owner" for *non modified shared data*, used in some extended MESI protocols (MERSI or R-MESI IBM, MESIF – Intel)
- The R/F state is used to allow "intervention" when the value is clean but shared among many caches. This cache is responsible for intervention (*shared intervention*).
- On bus read request, the data is supplied by this cache instead of the memory.
- The state **R** = Recent is used not only in MERSI = R-MESI protocol but in several other protocols. This state can be used in combination with other states. For instance RT-MESI, HR-MESI, HRT-MESI, HRT-ST-MESI. All protocols that use this state will be refereed as R-MESI type.

Cache states

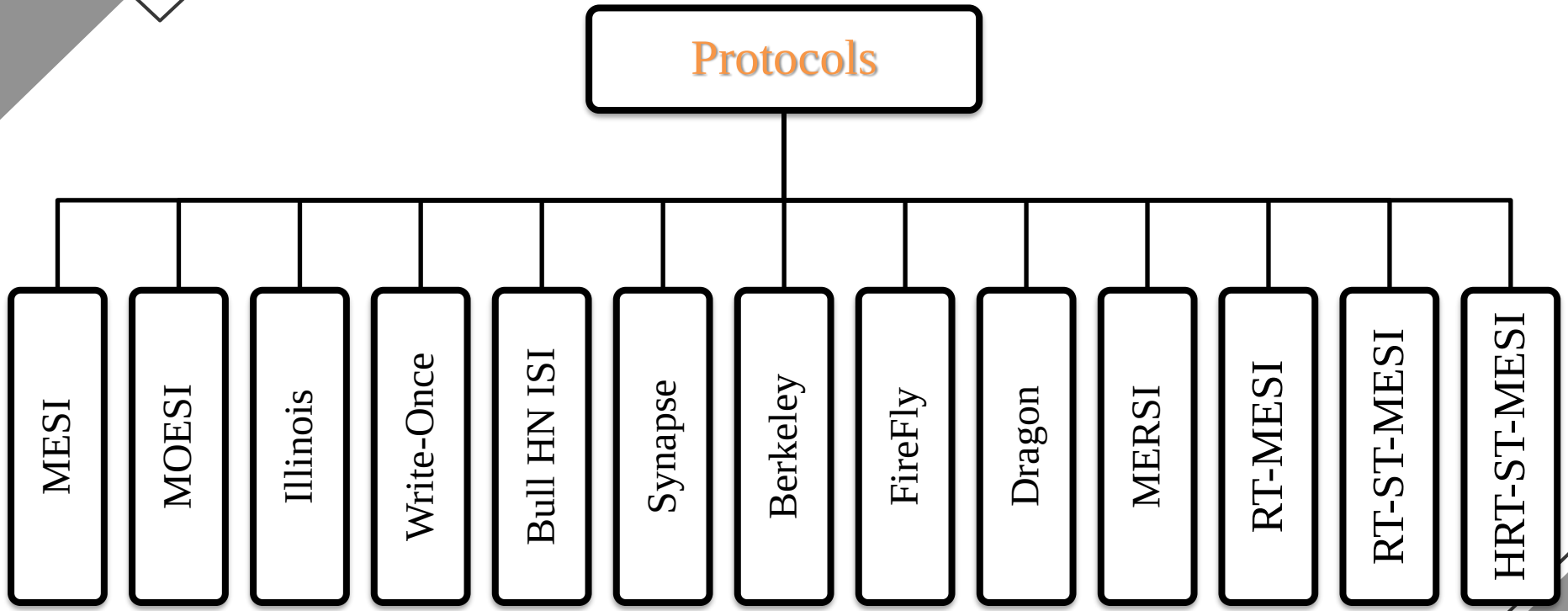
H= *Hover* – H-MESI (additional state of MESI protocol)

- The Hover (H) state allows a cache line to maintain an address Tag in the directory even though the corresponding value in the cache entry is an invalid copy.
- If the correspondent value happens on the bus (address Tag matching) due a valid "Read" or "Write" operation, the entry is updated with a valid copy and its state is changed in **S**
- This state can be used in combination with other states. For instance HR-MESI, HT-MESI, HRT-MESI, HRT-ST-MESI

02

Coherency Protocols

Coherency Protocols



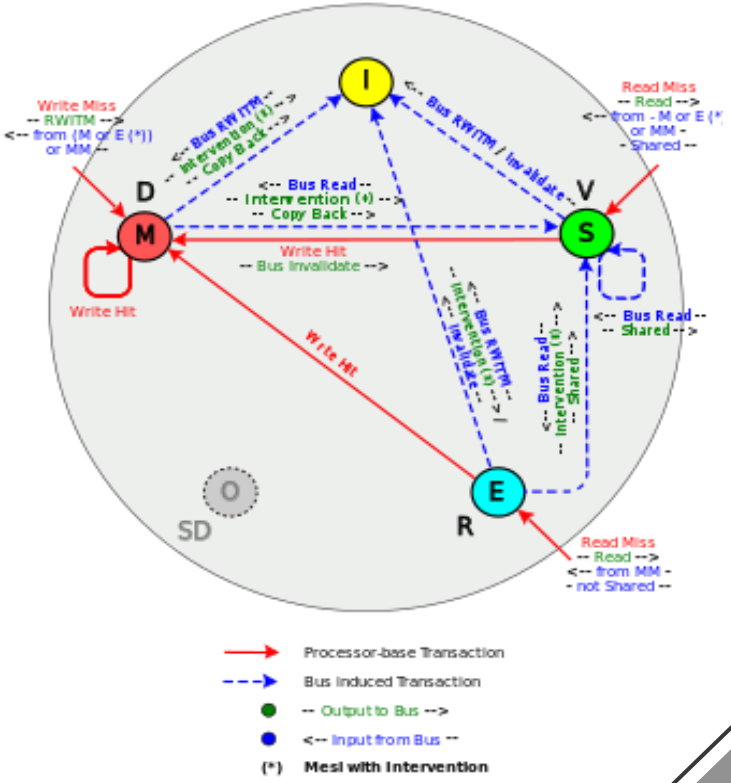
1. MESI Protocol

States MESI = D-R-V-I

Use of a bus "shared line" to

detect "shared" copy

in the other caches



1. MESI Protocol

Processor operations:

- Read Miss

There are two alternative implementations:

1. MESI "no Intervention" (e.g. PowerPC 604)

– If there is a **M** copy in a cache, the transaction is stopped and wait until the **M** cache updates the MM, then the transaction can continue and the data is read from MM. Both caches are set **S**

– else the data is read from MM. If the "shared line" is "on" the cache is set **S** else **E**

2. MESI "Intervention" from **M** and **E** (e.g. Pentium (R) II)

– If there is a **M** or **E** copy (exclusiveness) in a cache, the data is supplied to the requesting cache from **M** or from **E** (Intervention). If the sending cache is **M** the data is also written at the same time in MM (copy back). All caches are set **S**

– else the data is read from MM. If the "shared line" is "on" the cache is set **S** else **E**

1. MESI Protocol

Processor operations:

- Write Hit

– If the cache is **M** or **E** (exclusiveness), the write can take place locally without any other action

– else the data is written in cache and an invalidating transaction is sent to the bus to invalidate all the other caches

– The cache is set **M**

- Write Miss

– RWITM operation is sent to the bus. The operation is made in two step: a "Read Miss" with "invalidate" command to invalidate all the other caches, then like with a "Write Hit" with the state **M**

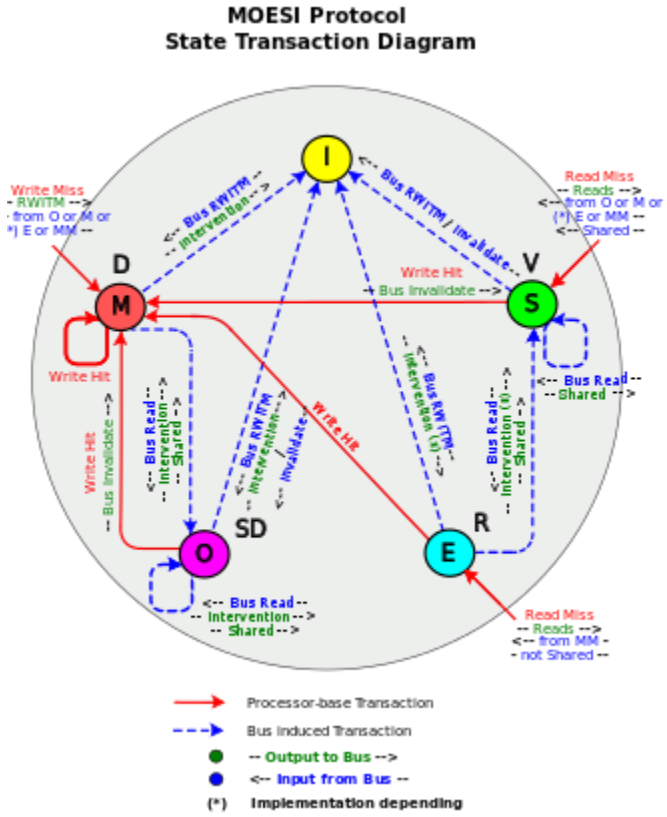
2. MOESI Protocol

States MEOSI = D-R-SD-V-I

Use of bus "shared line" to

detect "shared" copy

on the other caches



2. MOESI Protocol

Processor operations:

- Read Miss

– If there is a **M** or **O** or **E** copy in another cache the data is supplied by this cache (intervention).

The requesting cache is set **S** , **M** is changed to **O** and **E** to **S**

– else the data is read from MM.

– If "shared line" is "on" the requesting cache is set **S** else **E**

2. MOESI Protocol

Processor operations:

- Write Hit

– If the cache is **M** or **E** (exclusiveness), the write can take place locally without any other action

– else **O** or **S** (sharing) an "Invalidation" transaction is sent on the bus to invalidate all the other caches.

– The cache is set (or remains) **M**

- Write Miss

– A RWITM operation is sent to the bus

– Data is supplied from the "owner" or from MM as with Read Miss, then cache is written (updated)

– The cache is set **M** and all the other caches are set **I**

3. Illinois Protocol

States MESI = D-R-V-I

– Characteristics:

– It is an extension of MESI protocol

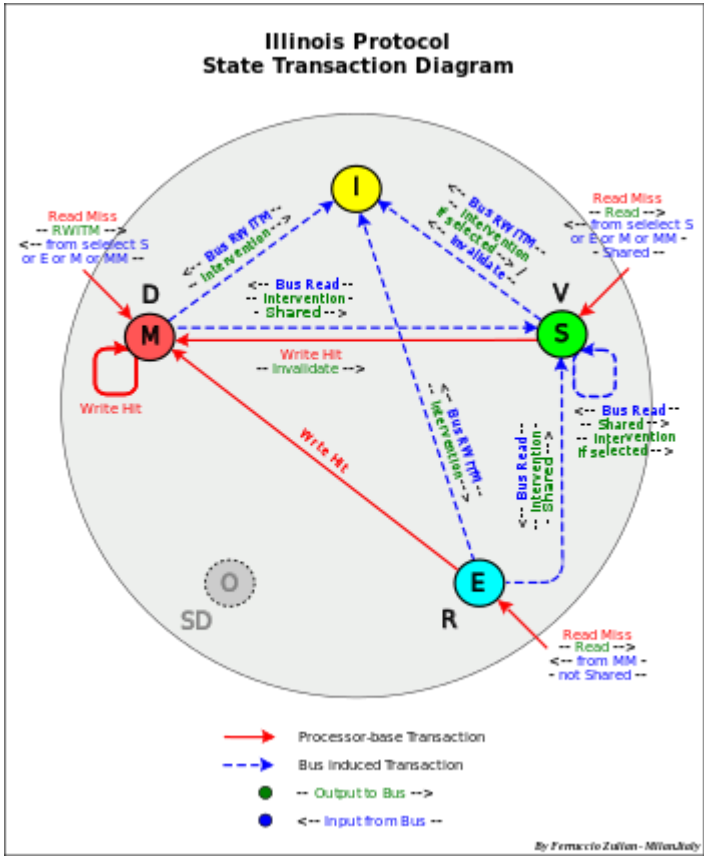
– Use of a network priority for **shared intervention** (intervention on shared data)

– Differences from MESI: in addition to **E** and **M**, intervention also from **S**

3. Illinois Protocol

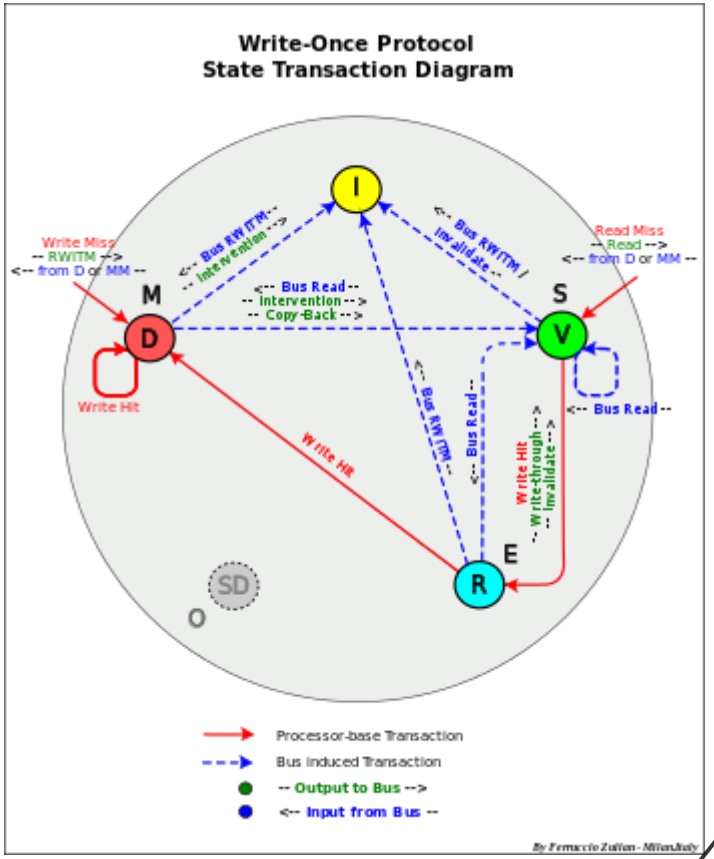
•**Operations:**

- Write Allocate
- Intervention: from M-E-S
- Write Invalidate
- Copy-Back: M replacement



4. Write-Once Protocol

States D-R-V-I (MESI)
– Characteristics:
– No use of "shared line" (protocol for standard or unmodifiable bus)
– Write Through on first Write Hit in state **V**, then Copy Back



4. Write-Once Protocol

Processor operations:

- Read Miss

– If there is a **D** copy in another cache, the data is supplied by this cache (intervention) and in the same time it is written also in MM (Copy-Back).

– else the data is read from MM

– all caches are set **V**

4. Write-Once Protocol

Processor operations:

- Write Hit

– If the cache is **D** or **R** (exclusiveness), the write can take place locally without any other action and the state is set (or remains) **D**

– else **V** (first Write Hit) the data is written in cache and in MM (Write Through) invalidating all the other caches (Write-Invalidate).

– The cache is set **R**

- Write Miss

– Like a Read Miss but with "invalidate" command (RWITM) plus a Write Hit in **D** state The cache is set **D** and all the other caches are set "Invalid"

– It is point out that in this case a bus transaction in any case is needed to invalidate the other caches and therefore it can be taken advantage of this fact to update also the MM. In "Write Hit" instead no more transaction is needed so a "Write Through" it would become a useless operation in case that the cache were updated again.

4. Write-Once Protocol

Bus transactions:

- Bus Read

– If the cache is **D** the data is sent to requesting cache (intervention) and to MM (copy-back). The cache is set **V**
– else the state is changed or remains in **V**
– If the cache is **D** the data is sent to the bus (Intervention)
– The cache is set "Invalid" (**I**)

- Bus Invalidate Transaction

– The cache is set "Invalid" (**I**)

5. Bull HN ISI Protocol

States D-SD-R-V-I (MOESI)

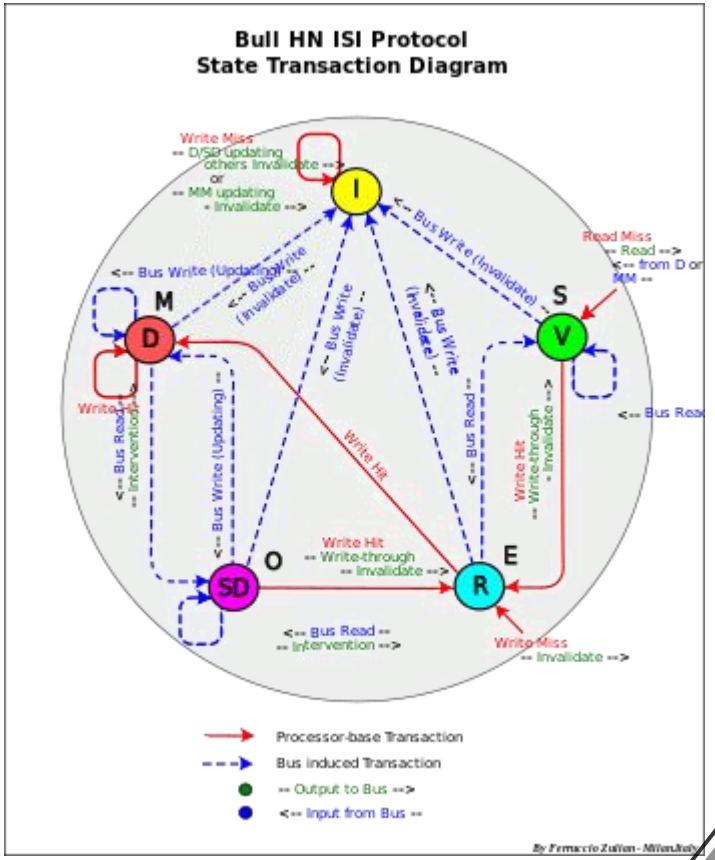
– Characteristics:

– MOESI extension of the Write-Once protocol

- Write-no-allocate on miss with **D** or **SD** updating

- No use of RWITM

- No use of "shared line"



5. Bull HN ISI Protocol

Processor operations:

- Read Miss

- Like with MOESI with "Shared Line" "on" and intervention only from the "owner" **D** or **SD** but not from **R**

- Write Hit

- If the cache is **D** or **R**, like with MOESI, the write can take place locally without any other action. The cache is set (or remains) **D**
- If **SD** or **V** (first write), like with Write-Once, the data is written in cache and in MM (Write Through) invalidating all the other caches (Write-Invalidate) – The cache is set **R**

- Write Miss

- The data is sent to the bus bypassing the cache (Write-no-allocate)
- If there is an "owner" copy **D** or **SD**, the "owner" is updated while the other caches are invalidated. The "owner" is set (or remains) **D**. The memory remains "dirty"
- else the data is sent to MM invalidating all the other caches (Write-Invalidate)

5. Bull HN ISI Protocol

Bus transactions:

- Bus Read

- Like with MOESI with intervention only from "owner" **D** or **SD**

- Bus Read (Write Update / Write Invalidate)

- If the cache is **D** or **SD**, the cache is updated, else is set "Invalid" (**I**)

5. Bull HN ISI Protocol

Operations:

- *Write-no-allocate*: on miss
- *Write update*: on miss
- *Write Through*: for the first write, then copy back
- *Write Update / Write Invalidate*
- *Intervention*: from SD-D
- *Copy-Back*: D replacement or SD replacement with invalidate

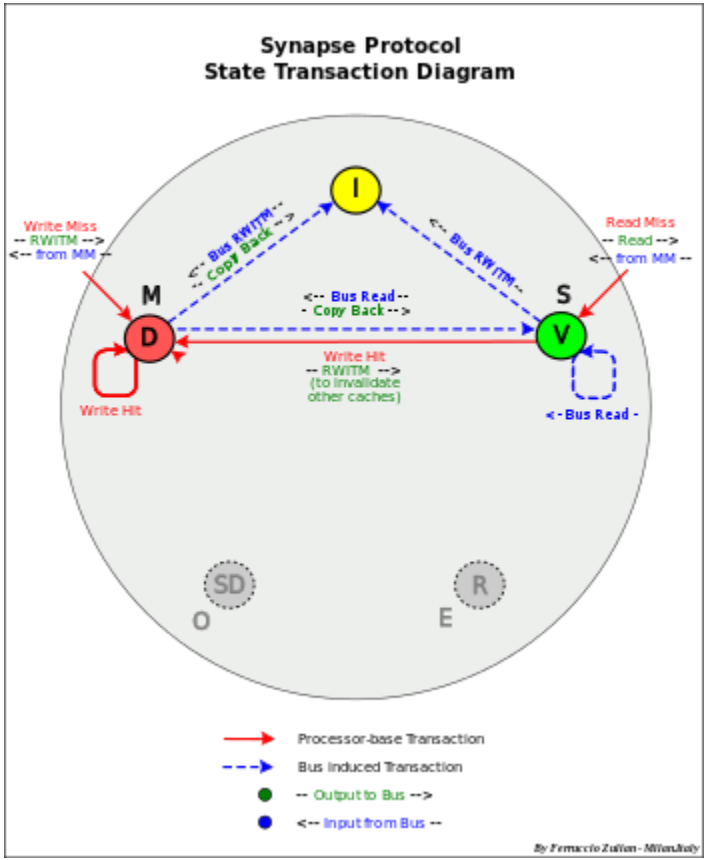
Obs. - This is the only protocol that has O-E (SD-R) transactions and it is also the only one that makes use of the Write-no-allocated on miss.

6. Synapse Protocol

States D-V-I (MSI)

Characteristics:

- The characteristic of this protocol is ti have a single-bit tag with each cache line in MM, indicating that a cache have the line in **D** state.
- This bit prevents a possible race condition if the **D** cache does not respond quickly enough to inhibit the MM from responding before being updating.
- The data comes always from MM
- No use of "shared line"



6. Synapse Protocol

Processor operations:

- Read Miss

- If there is a **D** copy in another cache, the read transaction is rejected (no acknowledgement).
- The **D** copy is written back to MM and changes its state in **V**, then the requesting cache resends a new read transaction and the data is read from MM.
- else the data is read from MM.
- The cache is set **V**

- Write Hit

- If the cache is **D** , the write can take place locally without any other action.
- else **V**, like with Read Miss does, including a data transfer from memory with in addition an invalidate command. This is done only to invalidate the other **V** copies
- The cache is set **D**. All the other caches copy are set "Invalid" (**I**)

- Write Miss

- Like with Read Miss, but with invalidate command. The cache line comes from MM, then the cache is written (updated). The cache is set **D**. All the other caches are set "Invalid" (**I**).

6. Synapse Protocol

Bus transactions:

- Bus Read

- If the cache is **D**, the data is sent to MM (Copy Back). The cache is set **V**

- else the state remains in **V**

- Bus Read

- If the cache is **D** the data is sent to MM (Copy Back)

- The cache (**D** or **V**) is set "Invalid" (**I**)

6. Synapse Protocol

Operations:

- Write Allocate
- Intervention: no intervention
- Write Invalidate
- No Invalidate transaction
- Copy-Back: D replacement

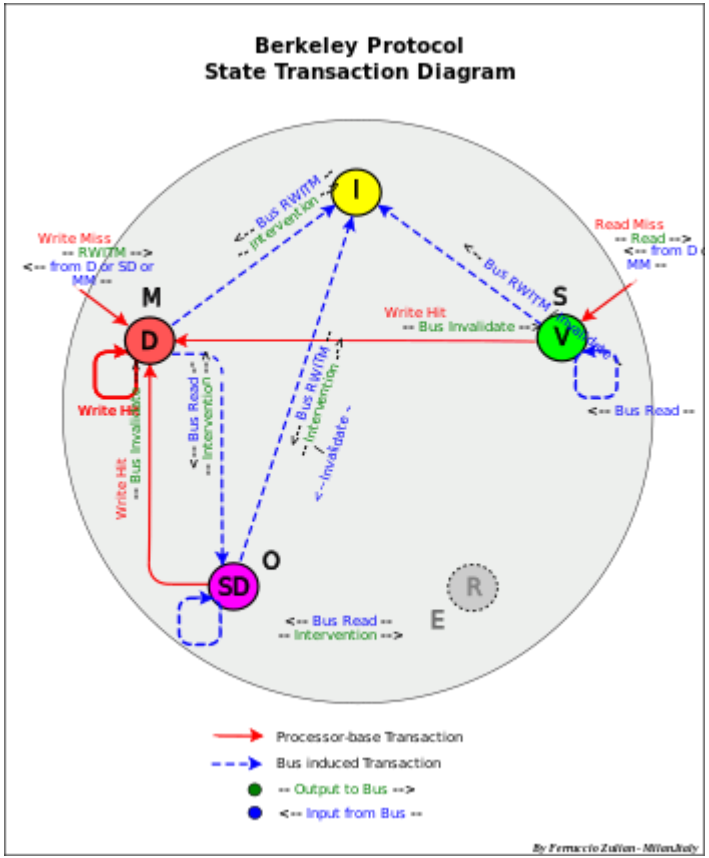
7. Berkeley Protocol

States D-SD-V-I (MOSI)

- Characteristics:

- As with MOESI without **E** state

- No use of "shared line"



7. Berkeley Protocol

Processor operations:

- Read Miss

- The data is supplied by the "owner", that is from **D** or from **SD** else from MM.
D is changed in **SD**
- The cache is set **V**

- Write Hit

- If the cache is **D** (exclusiveness), the write can take place locally without any other action
- else (**SD** or **V**), an "Invalidation" transaction is sent on the bus to invalidate the other caches.
- The cache is set (or remains) **D**

- Write Miss

- RWITM operation is sent to the bus
- Like with Read Miss, the data comes from the "owner", **D** or **SD** or from MM, then the cache is updated
- The cache is set **D**. all the other caches are set **I**

7. Berkeley Protocol

Bus transactions:

- Bus Read

- If the cache is **D** or **SD** the data is sent to requesting cache (intervention). The cache is set (or remains) in **SD**
- else the cache remains in **V**

- Bus Read

- If the cache is **D** or **SD** the data is sent to the bus (Intervention)
- The cache is set "Invalid" (**I**)

- Bus Invalidate Transaction

- The cache is set "Invalid" (**I**)

7. Berkeley Protocol

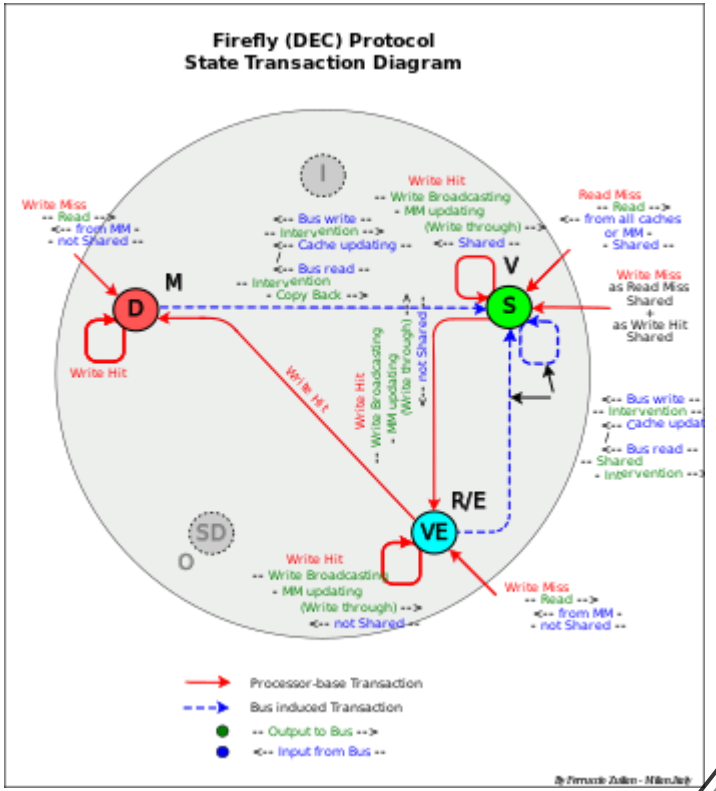
Operations:

- *Write Allocate*
- *Intervention*: from D-SD
- *Write Invalidate*
- *Copy-Back*: D-SD replacement

8. Firefly Protocol

States D-VE-S (MES)

- Characteristics:
- No "Invalid" state
- "Write-broadcasting” + "Write Through"
- Use of "shared line"
- "Write-broadcasting" avoid the necessity of "Invalid" state
- Simultaneous intervention from all caches (shared and dirty intervention – on not modified that modified data)
- This protocol requires a synchronous bus



8. Firefly Protocol

Processor operations:

- Read Miss

- Any other cache is the "owner", that is all the other caches with a copy supplied simultaneously the date on the bus (simultaneous intervention – the bus timing is fixed so that they all respond in the same cycle), otherwise the data is supplied from MM.
- If there is a cache **D**, the data is sent simultaneously also to MM (Copy Back)
- If there are copies in the other caches, the "Shared line" is set "on"
- If "Shared line" is "on" all the caches are set **S** else the requesting cache is set **VE**.

8. Firefly Protocol

Processor operations:

- Write Hit

- If the cache is **D** or **VE** (exclusiveness), the write can take place locally without any other action
and the cache is set **D**
- else **S**, a "Write-broadcasting" is sent to the bus to update all the other caches and the MM (Write Through)
- If there is a copy in another cache, the "Shared line" is set "on". If "Shared line“ is set "off" the cache is set **VE** else all caches are set **S**

- Write Miss

- The operation is made in two steps. Read Miss then Write Hit.
- If the data comes from a cache (Shared Line "on") a "Write-broadcasting" is sent to the bus to update all the other caches and the MM (Write Through).
All the caches are set **S**
- else the cache is set **D**

8. Firefly Protocol

Bus transactions:

- Bus Read

- If hit (**D** or **VE** or **S**) the data is sent to the bus (intervention) and in case of **D** the data is written also in MM. The cache is set **S**

- Bus Read

- If hit (**D** or **VE** or **S**) the data is sent to the bus (Intervention).
- All the caches are set **S**

- Write Broadcasting

- The cache is updated with new data. The state remains **S**

8. Firefly Protocol

Operations:

•Write Allocate

•Intervention: from D-VE-S (from all "valid" caches)

•Write-broadcasting – Write through

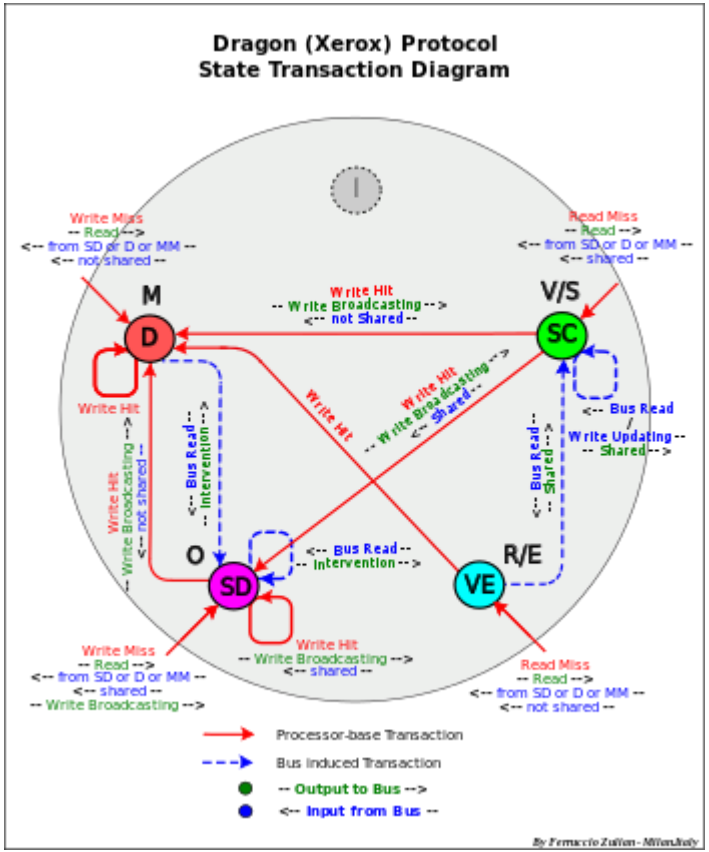
•Copy-Back: D replacement and on any transaction with a cache D

9. Dragon Protocol

States D-SD-VE-SC (MOES)

Note – the state **SC**, despite of the term "clean", can be "clean" or "dirty" as the **S** state of the other protocols. **SC** and **S** are equivalents

- Characteristics:
- No "Invalid" state
- "Write-broadcasting" (no "Write Through")
- Use of "shared line"
- "Write-broadcasting" avoid the necessity of "Invalid" state



9. Dragon Protocol

Processor operations:

- Read Miss

- The data is supplied by the "owner", that is from **D** or from **SD** else from MM.
D is changed in **SD**
- If "shared line" is "on" the cache is set **SC** else **VE**

- Write Miss

- Like with Read Miss, the data comes from the "owner", **D** or **SD** or from MM,
then the cache is updated
- If there is a copy in another cache, the "Shared line" is set "on".
- If the "Shared Line" is "on" the updated data is broadcast to the other caches and the state is set **SD**.
All the other caches are set **SC**
- else the cache is **D**

9. Dragon Protocol

Processor operations:

- Write Hit

- If the cache is **D** or **VE** (exclusiveness), the write can take place locally without any other action.
The cache is set (or remains) **D**
- else **SD** or **SC** (sharing) the data is written in cache and a "Write -broadcasting" is sent to the bus to update all the other caches – The MM is not updated (no Write through)
- If there is a copy in another cache, the "Shared line" is set "on"
- If the "Shared Line" is "on" the cache is set **SD**, else **D**. All the other caches possible copy are set **SC**

9. Dragon Protocol

Bus transactions:

- Bus Read
- If the cache is **D** or **SD** the data is sent to requesting cache (intervention). The cache is set (or remains) **SD**
- else the cache remains **SC**

- Bus Read

- If the cache is **D** or **SD** the data is sent to the bus (Intervention)
- The cache is set **SC**

- Write Broadcasting

- The cache is updated with new data. The cache remains **SC**

9. Dragon Protocol

Operations:

- Write Allocate
- Intervention: from D-SD (but not from VE)
- Write-broadcasting
- Copy-Back: D-SD replacement

10. MERSI Protocol

States MERSI or R-MESI

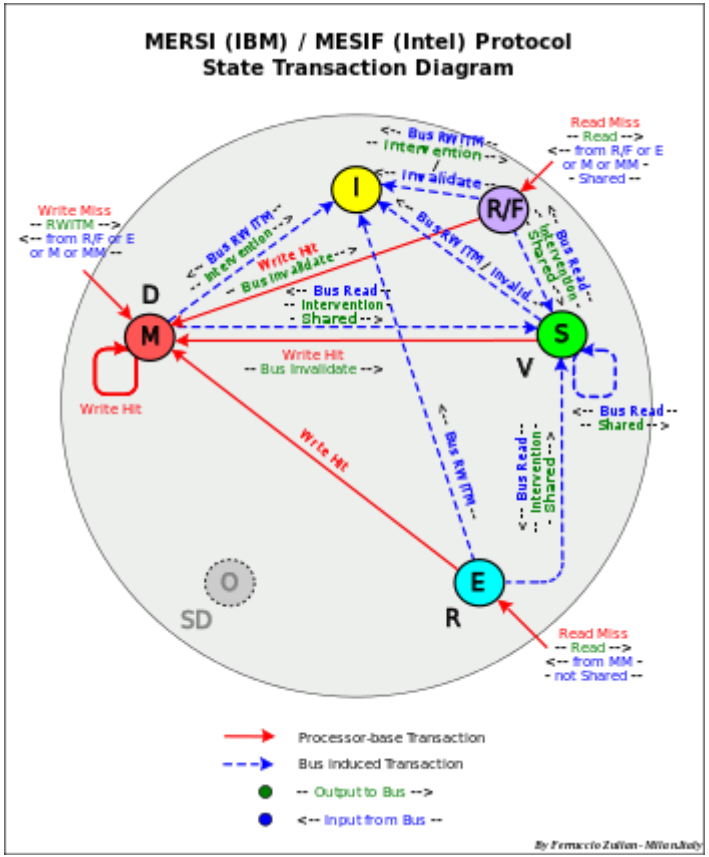
-Characteristics:

- The same functionality of Illinois protocol
- A new state **R** (Recent) / **F** (Forward) is the "*owner* " for "shared-clean" data (with MM updated).
- The "shared ownership" (on clean data) is not assigned by a network priority like with Illinois, but it is always assigned to the last cache with Read Miss, setting its state **R**/**F**
- The "ownership" is temporary loosed in case of **R**/**F** replacement. The "ownership" is reassigned again to the next Read Miss with caches "shared clean"
- Use of the "shared line"

10. MERSI Protocol

Operations

- Write Allocate
- Intervention: from M-E-R/F
- Write Invalidate
- Copy-Back: M replacement

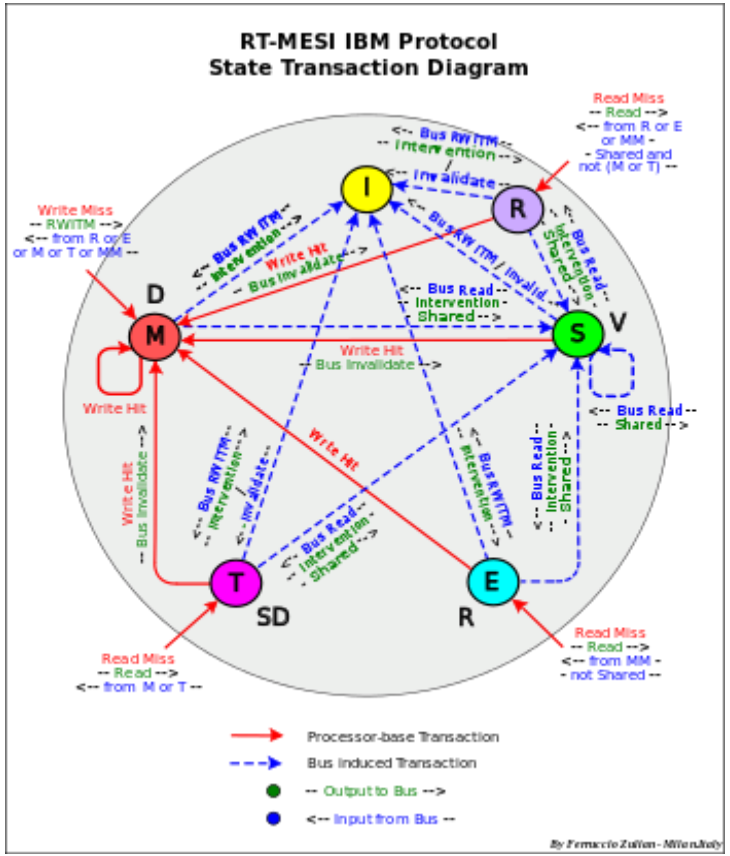


11. RT-MESI Protocol

States RT-MESI
IBM patented protocol

Characteristics:
- MESI and MOESI merging
- Shared Intervention + Dirty Intervention (both on clean and dirty data)
- Same functionality of R-MESI protocol with a new state **T**=Tagged, equivalent to **O** state
- "Dirty-Owner" migration
- The "owner" (both Shared or Dirty) is always the last requesting cache (the new "owner" (LRU) has less probability to be deallocated soon compared to the old one)
- The "owners" are **T**, **M**, **E**, **R** (all except **S**)
- Use of the "shared line"

11. RT-MESI Protocol



11. RT-MESI Protocol

Processor operations:

- Read Miss

- If there is a **M** or **T** (dirty-ownership) copy in another cache, the data is supplied by this cache (dirty intervention). The requesting cache is set **T** and the previous **M** or **T** are changed in **S**

- If there is a **E** or **R** (shared-ownership) copy in another cache, the data is supplied by this cache (shared intervention). The requesting data is set **R** and **E** or **R** are changed in **S**

- else the data is read from MM and the cache is set **R**.

11. RT-MESI Protocol

Processor operations:

- Write Hit

- If the cache is **M** or **E** (exclusiveness), the write can take place locally without any other action
- else **T** or **R** or **S** (sharing) an "Invalidation" transaction is sent on the bus to invalidate all the other caches.
- The cache is set (or remains) **M** and all the other caches are set **I**

- Write Miss

- RWITM operation is sent to the bus
- Data is supplied from the "owner" or from the MM as with Read Miss, then the data is written (updated) in cache.
- The cache is set **M** and all the other caches are set **I**

11. RT-MESI Protocol

Bus transactions:

- Bus Read

- If the cache is **T** or **M** or **R** or **E** the data is sent to requesting cache (intervention).
- The cache is set (or remains) in **S**

- Bus Read

- If the cache is **T** or **M** or **R** or **E** the data is sent to requesting cache (intervention)
- The cache is set "Invalid" (**I**)

- Bus Invalidate Transaction

- The cache is set "Invalid" (**I**)

12. RT-ST-MESI Protocol

It is an improvement of RT-MESI protocol and it is a subset of HRT-ST-MESI protocol

S_T = Shared-Tagged

- Use of the "Shared-Tagged" state allows to maintain intervention after deallocation of a Tagged cache line
- In case of **T** replacement (cache line deallocation), the data needs to be written back to MM and so to lose the "ownership".
- To avoid this, a new state **S_T** can be used.
- In Read Miss the previous **T** is set **S_T** instead of **S**.
- **S_T** is the candidate to replace the ownership in case of **T** deallocation.
- The **T** "Copy back" transaction is stopped (no MM updating) by the **S_T** cache that changes its state in **T**.
- In case of a new read from another cache, this last is set **T**, the previous **T** is changed in **S_T** and the previous **S_T** is changed in **S**

12. RT-ST-MESI Protocol

An additional improvement can be obtained using more than a $\mathbf{S_T}$ state, $\mathbf{S_{T1}}$, $\mathbf{S_{T2}}$, … $\mathbf{S_{Tn}}$

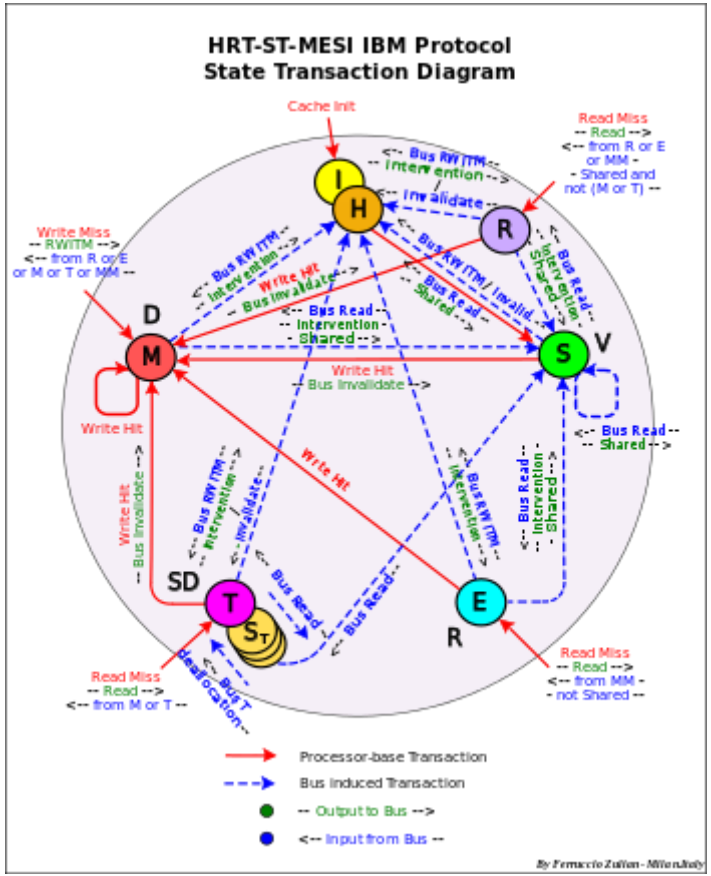
- In Read Miss, **T** is changed in $\mathbf{S_{T1}}$ and all the indices of the others $\mathbf{S_{Ti}}$ are increased by "1.
- In case of **T** deallocation, $\mathbf{S_{T1}}$ stops the "Copy Back" transaction, changes its state in **T** and all the indices of the others $\mathbf{S_{Ti}}$ are decrease by "1".
- In case of a deallocation, for instance $\mathbf{S_{Tk}}$, the chain will be interrupted and all the $\mathbf{S_{Ti}}$ with index greater of "k" are automatically loosen in term of $\mathbf{S_T}$ and will be considered *de facto* only as simple **S** states also if they are set as $\mathbf{S_T}$.
-All this because only $\mathbf{S_{T1}}$ intervenes to block and to replace itself with **T**.

13. HRT-ST-MESI Protocol

IBM patented full HRT-ST-MESI protocol
- **I** state = Invalid Tag – Invalid Data
- **H** state = Valid Tag – Invalid Data
- **I** state is set at the cache initialization and its state changes only after a processor Read or Write miss. After it will not return more in this state.
- **H** has the same functionality of **I** state but in addition with the ability to capture any bus transaction that match the Tag of the directory and to update the data cache.
- After the first utilization **I** is replaced by **H** in its functions
- The main features are :
- Write Back
- Intervention both in sharing-clean and dirty data – from T-M-R-E
- Reserve states of the Tagged (Shared-Tagged)
- Invalid **H** state (Hover) auto-updating

– Note: The Tag for definition is always valid, but until the first updating of the cache line it is considered invalid in order to avoid to update the cache also when this line has been not still required and used.

13. HRT-ST-MESI Protocol



03

YouTube Links

Cache Coherence Video Links

1) Cache Coherence

1) Cache Coherence problems & Protocols