Problem 4: Cache coherence: MESI (20 pts)

Assume you are designing a MESI snoopy bus cache coherence protocol for write-back private caches in a multi-core processor. Each private cache is connected to its own processor core as well as a common bus that is shared among other private caches.

There are 4 input commands a private cache may get for a cacheline. Assume that bus commands and core commands will not arrive at the same time:

```
CoreRd:  Read request from the cache’s core
CoreWr:  Write request from the cache’s core
BusGet:  Read request from the bus
BusGetI: Read and Invalidate request from the bus (invalidates shared data in
the cache that receives the request)
```

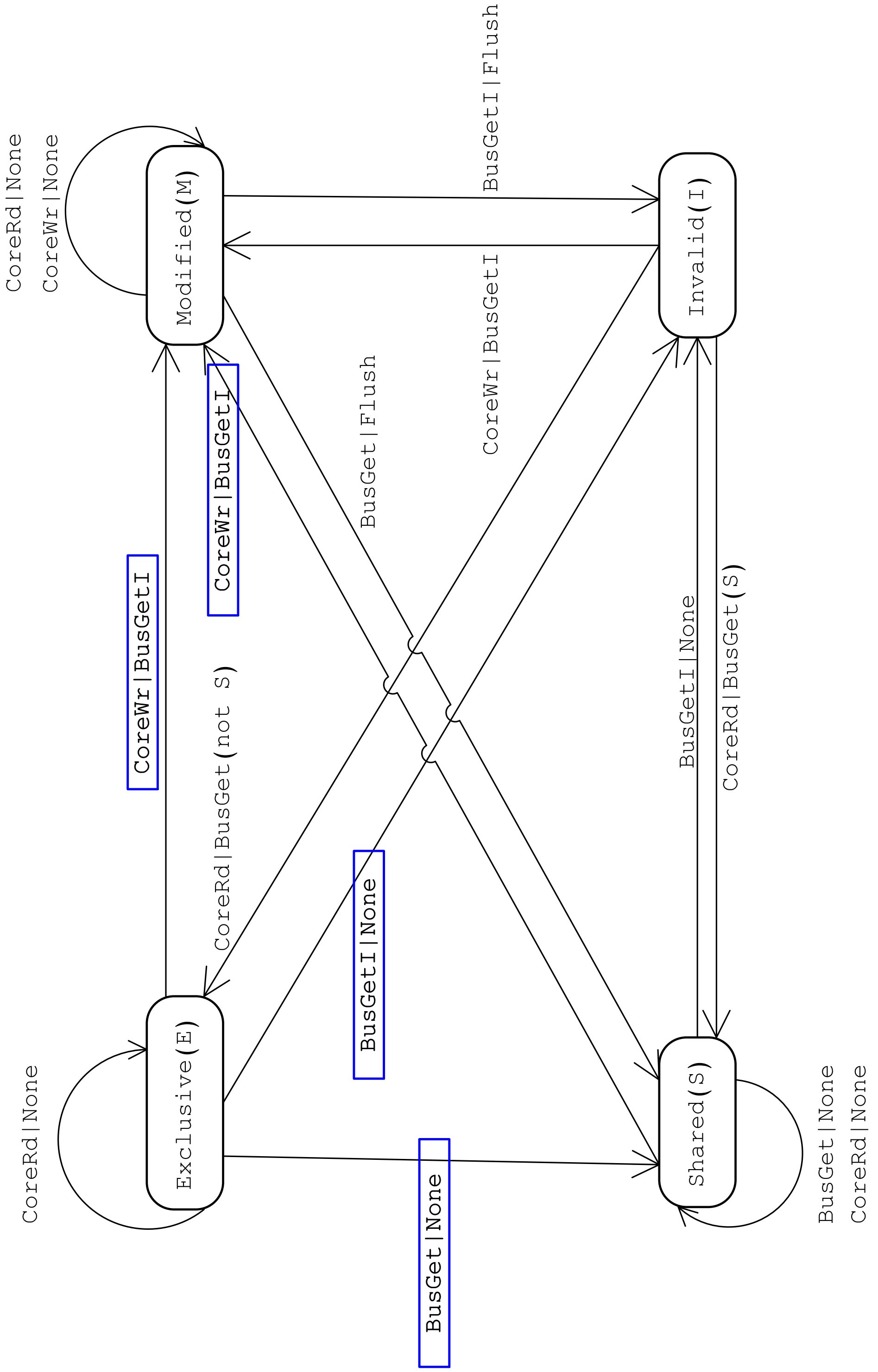
There are 4 actions a cache can take while transferring to another state:

```
Flush:  Write back the cacheline to lower-level cache
BusGet:  Issue a BusGet request to the bus
BusGetI: Issue a BusGetI request to the bus
None:    Take no action
```

There is also an “is_shared (S)” signal on the bus. S is asserted upon a BusGet request when at least one of the private caches shares a copy of the data (BusGet (S)). Otherwise S is deasserted (BusGet (not S)).

Assume upon a BusGet or BusGetI request, the inclusive lower-level cache will eventually supply the data and there are no private cache to private cache transfers.

On the next page, Hongyi drew a MESI state diagram. There are 4 mistakes in his diagram. **Please show the mistakes and correct them.** You may want to practice on the scratch paper first before finalizing your answer. If you made a mess, clearly write down the mistakes and the changes below.


Modified(M)
Invalid(I)
Exclusive(E)
Shared(S)
CoreRd|None
CoreWr|None
BusGetI|Flush
CoreWr|BusGetI
BusGet|Flush
CoreWr|BusGetI
CoreWr|BusGetI
CoreRd|BusGet(not S)
BusGetI|None
BusGetI|None
CoreRd|BusGet(S)
CoreRd|None
BusGet|None
BusGet|None
CoreRd|None