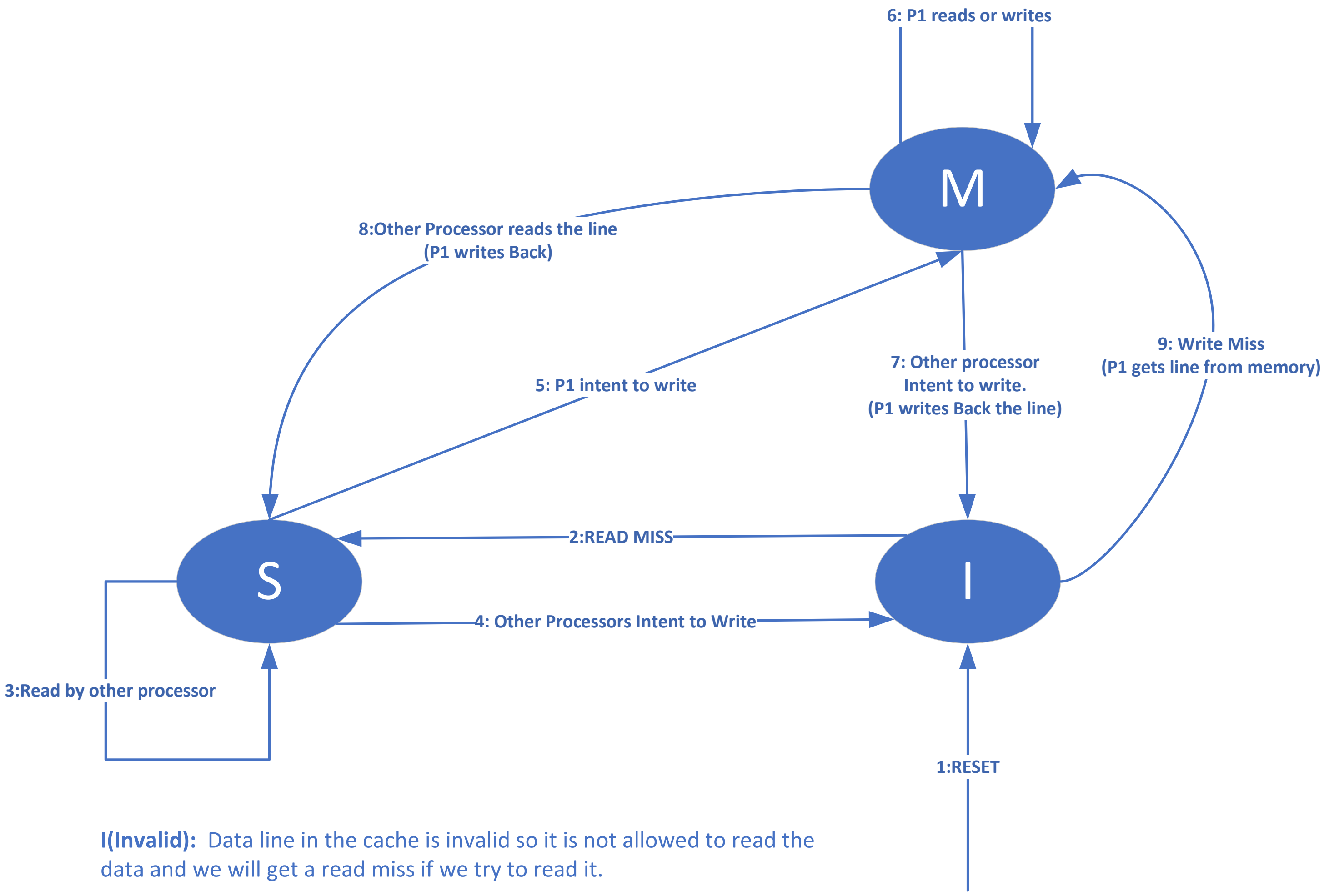
6: P1 reads or writes
M
8:Other Processor reads the line
(P1 writes Back)
5: P1 intent to write
7: Other processor
Intent to write.
(P1 writes Back the line)
9: Write Miss
(P1 gets line from memory)
S
2:READ MISS
I
4: Other Processors Intent to Write
3:Read by other processor
1:RESET

I(Invalid): Data line in the cache is invalid so it is not allowed to read the data and we will get a read miss if we try to read it.

M(Modified): Data is dirty. The data line in the cahce is modified after it was read from the main memory. It is the only copy. If a line is in the M state, no other cache can have the copy of the line. Before the line is moved to the Modified state, all the other copies are invalidated.

S(Shared): The line is a ready only copy. Some other processor may have a copy of the cache line, too.

2: P1 wants to read the data line and it is not in the cache. So P1 gets the line from the cache.
3: P1 has the cache line in the S state and some other processor also wants to read the cache line. No change is needed.
4: Another processor needs to write to a cache line that P1 has in shared state. P1 changes the state of the cache line to Invalid and doesn't need to write it back to cache(since it is not modified). It doesn't need to let anybody know that it had the line either. So if P1 wants to read that cache line again, it needs to go and read it again because the current copy is Invalid.
5: First P1 broadcast the intent to write to the cache so other processors know that P1 is about to write to that cache line. Some implementation strategies wait some time to let other processor invalidate their data, write data back to memory if they need to and so on. Some other implementations might wait for an ACK signal from all the other caches after the write intent is broadcasted. Or, if everyone has the read only copy of the data, they need to transition from Shared state to Invalid state. After the broadcast, all other caches need to invalidate this line in their cache line.
6: No other processor has the copy of the cache line so P1 can read and write to the cache line without communication with the other processors on the bus.
7: Some other processor broadcasts on the bus that it is going to write to a cache line and P1 has the same cache line in the Modified state(Most updated copy of the line).
8: P1 has the most up-to-date copy of the line so the other processors should get this copy. So P1 write the line back to the main memory first and then transition to the shared state. Then other processors can read the line from the main memory and get the most up-to-date copy of the line. So the cache in both processors will be in the shared state.
9: The cache line is in the Invalid state. First, P1 broadcasts the write intent and all the other processors