```
MODULE cache invalidation v3
Cache Handle Invalidation Message \stackrel{\Delta}{=}
    \land \exists message \in invalidationQueue :
        \land \lor \land cache[message.key] \in CacheHit
               Message needs to be newer than the cache
               \land cache[message.key].version < message.version
            Or not in the cache, but with a pending fill request
            \lor \land cache[message.key] \in CacheMiss
              \land cacheFillStates[message.key].state \neq "inactive"
         Update item in cache
        \wedge cache' = [cache \ EXCEPT]
                          ![message.key] = [
                              type \mapsto "hit",
                               Update to version in invalidation message
                              version \mapsto message.version
         Remove message from queue because handled
        \land invalidationQueue' = invalidationQueue \setminus \{message\}
    ∧ UNCHANGED ⟨cacheFillStates, database, counter⟩
CacheIgnoreInvalidationMessage \stackrel{\Delta}{=}
    \land \exists message \in invalidationQueue : Dequeue invalidation queue in any order
         Ignore invalidation messages for messages not in cache
        \land \lor \land cache[message.key] \in CacheMiss
               and a fill is not occurring
               \land cacheFillStates[message.key].state = "inactive"
            Or when the cache already has the same or larger version
            \lor \land cache[message.key] \notin CacheMiss
              \land cache[message.key].version \ge message.version
         Remove message from queue to ignore
        \land invalidationQueue' = invalidationQueue \setminus \{message\}
    \wedge counter' = counter + 1
    \land UNCHANGED \langle cacheFillStates, database, cache \rangle
CacheEvict(k) \triangleq
    \land cache[k] \in CacheHit
     A key with a pending request will not be evicted
    \land cacheFillStates[k].state = "inactive"
```

 $\land cache' = [cache \ EXCEPT \ ![k] = [type \mapsto "miss"]]$

 $invalidationQueue\rangle$

 \land UNCHANGED $\langle database, cacheFillStates,$

 $\wedge counter' = counter + 1$