```
std::atomic<int> seqlock;
int protected_data;

int reader() {
  while (true) {
    int s1 = seqlock;
    int ret = protected_data;  // ERROR
    int s2 = seqlock;
    if (s1 == s2 && is_even(s1))
      return ret;
  }
}
void writer(int newval) {
  while (true) {
    unsigned s = seqlock;
    if (is_even(s) && seqlock.compare_exchange_strong(s, s+1)  {
      protected_data = newval;
      seqlock = s + 2;
      return;
    }
  }
}
```

FIGURE 20.1 Incorrect use of a sequence lock: The lock release is not guaranteed to be ordered after the data access.