

## Exercise 3

### 1 MRMW Atomic R/W Register

In the shared-memory model, implement a MRMW atomic R/W register using MRSW atomic R/W registers. Idea: every writer uses a separate pair of registers  $(\tau, x)$  to write, and  $\tau$  is implemented as a vector clock.

### 2 Fault-tolerant Register Implementation in a Model with Crashes and Recoveries

Consider a fault-tolerant implementation of a R/W register in a model where the servers may crash and recover later. Any server may crash and it may even be that all servers are crashed at the same time. A *correct* server is one that either never crashes or eventually recovers and never crashes again.

In order to log data persistently, every server has access to a local stable storage module through two operations *store* and *retrieve*, which atomically store and retrieve a set of values to and from stable storage, respectively.

Assume that messages between clients and servers are sent over reliable channels in the crash/recovery model, i.e., messages are retransmitted by the sender until an acknowledgement is received from the receiver.

Describe a protocol for implementing a MRSW regular register in this model, using the majority quorum system and assuming that a majority of the servers is correct.