```
──────────────────────── MODULE Chain_SS ────────────────────────
Chain SimpleStore
EXTENDS Naturals, Sequences, Util, TLC
CONSTANT Val,
           NoVal,
           MaxReq

VARIABLE pending_wrreq,       list of pending update requests
         store,               data value of the store
           pending_rdreq,       list of pending read requests
           last_read_val        last read value

├────────────────────────────────────────────────────────────────┤
SS_TypeInvariant ≜
        ∧ pending_wrreq ∈ [Val → Nat]
        ∧ store ∈ Val ∪ {NoVal}
        ∧ pending_rdreq ∈ Nat
        ∧ last_read_val  ∈ Val ∪ {NoVal}

SS_Init ≜
        ∧ pending_wrreq = [v ∈ Val ↦ 0]
        ∧ store = NoVal
        ∧ pending_rdreq = 0
        ∧ last_read_val  = NoVal
├────────────────────────────────────────────────────────────────┤

Client operations:
SS_CliWrite(w) ≜   write request
    ∧ pending_wrreq' = [pending_wrreq EXCEPT ![w] = @ + 1]
    ∧ UNCHANGED ⟨store, last_read_val, pending_rdreq⟩

SS_CliRead ≜   read request
    ∧ pending_rdreq' = pending_rdreq + 1
    ∧ UNCHANGED ⟨store, last_read_val, pending_wrreq⟩

SS_Client ≜
    ∨ ∃ v ∈ Val : SS_CliWrite(v)
    ∨ SS_CliRead

STORE operations:
SS_HdlRead ≜   handle one read requests
    ∧ pending_rdreq > 0
    ∧ pending_rdreq' = pending_rdreq − 1
    ∧ last_read_val'  = store  get the most current value of the store

_CommitWrite(w) ≜   commit write w
    ∧ pending_wrreq[w] > 0  there's a write to commit
```

```
  $\land\ pending\_wrreq' = [pending\_wrreq \text{ EXCEPT } ![w] = @ - 1]$
  $\land\ store' = w$
$SS\_CommitWrite \triangleq$
  $\exists\, v \in Val : \_CommitWrite(v)$

$SS\_RunStore \triangleq$
  $\lor\ \land\ SS\_HdlRead$
     $\land\ \text{UNCHANGED } \langle pending\_wrreq,\ store \rangle$
  $\lor\ \land\ SS\_CommitWrite$
     $\land\ \text{UNCHANGED } \langle pending\_rdreq,\ last\_read\_val \rangle$
```

Channel operations:

```
$MapToSeq(map) \triangleq$
  $\text{LET } F[set \in \text{SUBSET } (Val)] \triangleq$
      $\text{IF } set\ = \{\} \text{ THEN } \langle\rangle$
       $\text{ELSE LET } v \triangleq \text{CHOOSE } v \in set : \text{TRUE}$
            $\text{IN } [i \in 1 \mathrel{..} map[v] \mapsto v] \circ F[set \setminus \{v\}]$
  $\text{IN}$
      $F[Val]$

$\_Drop(map,\ seq,\ drop\_idxs) \triangleq$
  $\text{LET } F[set \in \text{SUBSET } (1 \mathrel{..} Len(seq)),\ ret \in [Val \rightarrow Nat]] \triangleq$
        $\text{IF } set = \{\} \text{ THEN } ret$
         $\text{ELSE LET } i \triangleq \text{CHOOSE } i \in set : \text{TRUEIN}$
                $F[set \setminus \{i\},\ [ret \text{ EXCEPT } ![seq[i]] = @ - 1]]$
  $\text{IN } F[drop\_idxs,\ map]$
$Drop(q) \triangleq$
  $\text{LET } seq \triangleq MapToSeq(q)$
  $\text{IN}$
      $\exists\, drop\_idxs \in \text{SUBSET } (1 \mathrel{..} Len(seq)) :$
        $pending\_wrreq' = \_Drop(q,\ seq,\ drop\_idxs)$

$FailPendingWrReq \triangleq$    fail one or more requests at once
     $Drop(pending\_wrreq)$

$FailPendingRdReq \triangleq$
  $\exists\, n \in 1 \mathrel{..} pending\_rdreq : pending\_rdreq' = pending\_rdreq - n$

$SS\_ChannelActions \triangleq$
  $\lor\ FailPendingWrReq \land \text{UNCHANGED } \langle store,\ pending\_rdreq,\ last\_read\_val \rangle$
  $\lor\ FailPendingRdReq \land \text{UNCHANGED } \langle store,\ pending\_wrreq,\ last\_read\_val \rangle$
  $\lor\ FailPendingWrReq \land FailPendingRdReq \land \text{UNCHANGED } \langle store,\ last\_read\_val \rangle$

$SS\_Combined \triangleq$
  $\land\ \exists\, v \in Val :$
     $\land\ pending\_wrreq[v] > 0$
```

$$\begin{array}{l}
\qquad \land\ store' = v \\
\qquad \land\ Drop([pending\_wrreq \text{ EXCEPT } ![v] = @ - 1]) \\
\quad \land\ \text{UNCHANGED } \langle pending\_rdreq,\ last\_read\_val \rangle
\end{array}$$

$$\begin{array}{l}
SS\_Next\ \triangleq \\
\quad \land\ Print(\text{“}Pending\_wrreq\text{”},\ pending\_wrreq)\ \#\ \langle\rangle \\
\quad \land\ Print(\text{“}Pending\_wrreq'\text{”},\ pending\_wrreq')\ \#\ \langle\rangle \\
\quad \land\ Print(\text{“}Store\text{”},\ store)\ \#\ 8 \\
\quad \land\ Print(\text{“}Store'\text{”},\ store')\ \#\ 8 \\
\quad \land\ Print(\text{“}last\_read\_val\text{”},\ last\_read\_val)\ \#\ 9 \\
\quad \land\ Print(\text{“}last\_read\_val'\text{”},\ last\_read\_val')\ \#\ 9
\end{array}$$

$$\begin{array}{ll}
\land\ \lor\ SS\_Client & \text{a client submits a request (query or update)} \\
\quad\ \lor\ SS\_RunStore & \text{the store deals } w/ \text{ the updates} \\
\quad\ \lor\ SS\_ChannelActions & \text{the incoming channel for the store drops requests suddenly} \\
\quad\ \lor\ SS\_Combined & \text{this is a commit combined with some droppings in the same state}
\end{array}$$

Full spec:

$$\begin{array}{l}
ssvars\ \triangleq\ \langle store,\ pending\_wrreq,\ pending\_rdreq,\ last\_read\_val \rangle \\
SS\_Spec\ \triangleq\ SS\_Init \land \Box[SS\_Next]_{ssvars}
\end{array}$$

$$\begin{array}{l}
MaxRequests\ \triangleq \\
\quad \land\ pending\_rdreq \leq MaxReq \\
\quad \land\ \forall\, v \in Val : pending\_wrreq[v] \leq MaxReq
\end{array}$$

Invariants

$$\begin{array}{l}
SS\_AllInvariants\ \triangleq \\
\qquad \land\ SS\_TypeInvariant
\end{array}$$

Theorem

$$\text{THEOREM } SS\_Spec \Rightarrow \Box SS\_AllInvariants$$