──────────── MODULE *StateCounter* ────────────
EXTENDS *Naturals*, *Sequences*, *Bags*, *TLC*, *SEC*
──

VARIABLES
    state,
    update,
    seq,
    incoming,   network variable
    msg,   network variable
    messageSet   network variable

$vars \triangleq \langle state, update, seq, incoming, msg, messageSet, SECvars \rangle$

$Vector \triangleq [Replica \to Nat]$
$InitVector \triangleq [r \in Replica \mapsto 0]$
$InitState \triangleq state = [r \in Replica \mapsto InitVector]$
$EmptyUpdate \triangleq update = [r \in Replica \mapsto 0]$
$Msg \triangleq [r : Replica, vc : Vector, seq : Nat, update : \text{SUBSET } Update]$
──

$Network \triangleq \text{INSTANCE } Network$
──

$TypeOK \triangleq$
    $\land \quad state \in [Replica \to Vector]$
    $\land \quad update \in [Replica \to \{0, 1\}]$
──

$Init \triangleq$
    $\land InitState$
    $\land EmptyUpdate$
    $\land seq = [r \in Replica \mapsto 0]$
    $\land Network!NInit$
    $\land SECInit$
──

$Inc(r) \triangleq$
    $\land state' = [state \text{ EXCEPT } ![r][r] = @ + 1]$
    $\land update' = [update \text{ EXCEPT } ![r] = 1]$
    $\land seq' = [seq \text{ EXCEPT } ![r] = @ + 1]$
    $\land SECUpdate(r, seq[r])$
    $\land \text{UNCHANGED } \langle incoming, msg, messageSet \rangle$

$Send(r) \triangleq$
    $\land update' = [update \text{ EXCEPT } ![r] = 0]$
    $\land Network!NBroadcast(r, [r \mapsto r, vc \mapsto state[r], seq \mapsto seq[r], update \mapsto StateUpdate(r)])$
    $\land SECSend(r)$
    $\land \text{UNCHANGED } \langle state, seq \rangle$

$SetMax(r, s) \triangleq \text{IF } r > s \text{ THEN } r \text{ ELSE } s$

$Receive(r) \triangleq$
  $\wedge \;\; \forall\, s \in Replica : state' = [state \;\text{EXCEPT}\; ![r][s] = SetMax(@,\; msg'[r].vc[s])]$
  $\wedge \;\; Network!NDeliver(r)$
  $\wedge \;\; SECDeliver(r,\; msg'[r])$
  $\wedge \;\; \text{UNCHANGED} \;\langle update,\; seq \rangle$

$Next \triangleq \;\; \wedge\, \exists\, r \in Replica : Inc(r) \vee Send(r) \vee Receive(r)$

$Spec \triangleq \;\; Init \wedge \Box[Next]_{vars}$

$SEC \triangleq \;\; \exists\, r1,\, r2 \in Replica :$
  $Sameupdate(r1,\, r2) \Rightarrow state[r1] = state[r2]$