MODULE *OpCounter*

EXTENDS
Naturals, *Sequences*, *SEC*

VARIABLES

counter,	*counter*[*r*]: current value of the counter at replica $r \in Replica$
buffer,	*buffer*[*r*]: number of increments performed since the last broadcast at replica $r \in Replica$
seq,	*seq*[*r*]: local sequence number at replica $r \in Replica$
incoming,	*incoming*[*r*]: incoming messages at replica $r \in Replica$
msg,	*incoming*[*r*]: current message at replica $r \in Replica$
messageSet	network variable

$vars \triangleq \langle counter, buffer, seq, incoming, msg, messageSet, SECvars \rangle$

$Msg \triangleq [r : Replica, buf : Nat, seq : Nat, update : \text{SUBSET } Update]$

$Network \triangleq \text{INSTANCE } ReliableNetwork$

$TypeOK \triangleq$
- $\land \quad counter \in [Replica \to Nat]$
- $\land \quad buffer \in [Replica \to Nat]$

$Init \triangleq$
- $\land\ counter = [r \in Replica \mapsto 0]$
- $\land\ buffer = [r \in Replica \mapsto 0]$
- $\land\ seq = [r \in Replica \mapsto 0]$
- $\land\ Network!RInit$
- $\land\ SECInit$

$Read(r) \triangleq counter[r]$

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

$Send(r) \triangleq$
- $\land\ buffer[r] \neq 0$
- $\land\ buffer' = [buffer \text{ EXCEPT } ![r] = 0]$
- $\land\ Network!RBroadcast(r, [r \mapsto r, seq \mapsto seq[r], update \mapsto OpUpdate(r), buf \mapsto buffer[r]])$
- $\land\ SECSend(r)$
- $\land\ \text{UNCHANGED } \langle counter, seq \rangle$

$Receive(r) \triangleq$
- $\land \quad counter' = [counter \text{ EXCEPT } ![r] = @ + msg'[r].buf]$

$\land$ $Network!RDeliver(r)$
$\land$ $SECDeliver(r,\ msg'[r])$
$\land$ UNCHANGED $\langle buffer,\ seq \rangle$

$Next \triangleq \exists\, r \in Replica : Inc(r) \lor Send(r) \lor Receive(r)$

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

$EmptyBuffer \triangleq buffer = [r \in Replica \mapsto 0]$
$EC \triangleq Network!EmptyChannel \land EmptyBuffer$
$\qquad \Rightarrow \forall\, r1,\ r2 \in Replica : counter[r1] = counter[r2]$

$SEC \triangleq \forall\, r1,\ r2 \in Replica : Sameupdate(r1,\ r2) \Rightarrow counter[r1] = counter[r2]$

---

* Modification History
* Last modified *Wed* May 15 16:52:35 *CST* 2019 by *zfwang*
* Last modified *Tue* May 07 00:57:30 *CST* 2019 by *xhdn*
* Last modified *Mon* May 06 15:51:30 *CST* 2019 by *jywellin*
* Created *Fri Mar* 22 20:43:27 *CST* 2019 by *jywellin*