MODULE *a_pluscal_abp*

EXTENDS *Naturals*, *Sequences*, *TLC*
CONSTANT *Msg*

*AltBitProtocol* from The *PlusCal* Algorithm Language paper by *Lamport*

$Remove(i,\ seq) \triangleq [j \in 1\ ..\ (Len(seq) - 1) \mapsto \text{IF } j < i \text{ THEN } seq[j] \text{ ELSE } seq[j+1]]$

*– wf*

```
--algorithm AltBitProtocol{
  variables
    input = ⟨⟩, output = ⟨⟩,
    msgChan = ⟨⟩, ackChan = ⟨⟩,
    newChan = ⟨⟩ ;
  macro Send( m, chan ) {
    chan := Append(chan, m) ;
   }
  macro Recv( v, chan ) {
    await chan ≠ ⟨⟩ ;     could also do Len(chan) > 0 ??
    v := Head(chan) ;
    chan := Tail(chan) ;
   }

  process ( Sender = "S" )
    variables next = 1, sbit = 0, ack ;
  {
  s:  while ( TRUE ) {
        either with ( m ∈ Msg ) {
          input := Append(input, m) ;
        } or {
          await next ≤ Len(input) ;
          Send(⟨input[next], sbit⟩, msgChan) ;
        } or {
          Recv(ack, ackChan) ;
          if ( ack = sbit ) {
            next := next + 1 ;
            sbit  := (sbit + 1)%2 ;
           } ;
        } ;
        print ⟨"Sender", input⟩ ;
      }
   } ;   end Sender process block
  process ( Receiver = "R" )
    variables rbit = 1, msg ;
  {
  r:  while ( TRUE ) {
```

```
      either {
        Send(rbit, ackChan) ;
      } or {
        Recv(msg, msgChan) ;
        if ( msg[2] ≠ rbit ) {
          rbit := (rbit + 1)%2 ;
          output := Append(output, msg[1]) ;
        } ;
      } ;
    }
  } ;  end Receiver process block

  process ( LoseMsg = "L" ) {
  l:  while ( TRUE ) {
      either with ( i ∈ 1 .. Len(msgChan) ) {
        msgChan := Remove(i, msgChan) ;
      } or with ( i ∈ 1 .. Len(ackChan) ) {
        ackChan := Remove(i, ackChan) ;
      } ;
    }
  } ;  end LoseMsg process block
}   \* end algorithm
```

BEGIN TRANSLATION ($chksum(pcal) =$ "99be29fe" $\land\ chksum(tla) =$ "54c15cec")

CONSTANT *defaultInitValue*

VARIABLES *input*, *output*, *msgChan*, *ackChan*, *newChan*, *next*, *sbit*, *ack*, *rbit*, *msg*

$vars \triangleq \langle input, output, msgChan, ackChan, newChan, next, sbit, ack, rbit, msg \rangle$

$ProcSet \triangleq \{\text{"S"}\} \cup \{\text{"R"}\} \cup \{\text{"L"}\}$

$Init \triangleq$ Global variables
$\land\ input = \langle\rangle$
$\land\ output = \langle\rangle$
$\land\ msgChan = \langle\rangle$
$\land\ ackChan = \langle\rangle$
$\land\ newChan = \langle\rangle$
Process *Sender*
$\land\ next = 1$
$\land\ sbit = 0$
$\land\ ack = defaultInitValue$
Process *Receiver*
$\land\ rbit = 1$
$\land\ msg = defaultInitValue$

$$
\begin{array}{l}
Sender \triangleq \land \lor \land \exists\, m \in Msg : \\
\qquad\qquad\qquad input' = Append(input,\ m) \\
\qquad\qquad \land \text{UNCHANGED } \langle msgChan,\ ackChan,\ next,\ sbit,\ ack\rangle \\
\qquad\quad \lor \land next \leq Len(input) \\
\qquad\qquad \land msgChan' = Append(msgChan,\ (\langle input[next],\ sbit\rangle)) \\
\qquad\qquad \land \text{UNCHANGED } \langle input,\ ackChan,\ next,\ sbit,\ ack\rangle \\
\qquad\quad \lor \land ackChan \neq \langle\rangle \\
\qquad\qquad \land ack' = Head(ackChan) \\
\qquad\qquad \land ackChan' = Tail(ackChan) \\
\qquad\qquad \land \text{IF } ack' = sbit \\
\qquad\qquad\qquad \text{THEN } \land next' = next + 1 \\
\qquad\qquad\qquad\qquad\quad \land sbit' \ = (sbit + 1)\%2 \\
\qquad\qquad\qquad \text{ELSE } \land \text{TRUE} \\
\qquad\qquad\qquad\qquad\quad \land \text{UNCHANGED } \langle next,\ sbit\rangle \\
\qquad\qquad \land \text{UNCHANGED } \langle input,\ msgChan\rangle \\
\qquad\quad \land PrintT(\langle \text{“Sender”},\ input'\rangle) \\
\qquad\quad \land \text{UNCHANGED } \langle output,\ newChan,\ rbit,\ msg\rangle
\end{array}
$$

$$
\begin{array}{l}
Receiver \triangleq \land \lor \land ackChan' = Append(ackChan,\ rbit) \\
\qquad\qquad\qquad\quad \land \text{UNCHANGED } \langle output,\ msgChan,\ rbit,\ msg\rangle \\
\qquad\qquad\qquad \lor \land msgChan \neq \langle\rangle \\
\qquad\qquad\qquad\quad \land msg' = Head(msgChan) \\
\qquad\qquad\qquad\quad \land msgChan' = Tail(msgChan) \\
\qquad\qquad\qquad\quad \land \text{IF } msg'[2] \neq rbit \\
\qquad\qquad\qquad\qquad\quad \text{THEN } \land rbit' = (rbit + 1)\%2 \\
\qquad\qquad\qquad\qquad\qquad\quad \land output' = Append(output,\ msg'[1]) \\
\qquad\qquad\qquad\qquad\quad \text{ELSE } \land \text{TRUE} \\
\qquad\qquad\qquad\qquad\qquad\quad \land \text{UNCHANGED } \langle output,\ rbit\rangle \\
\qquad\qquad\qquad\quad \land \text{UNCHANGED } ackChan \\
\qquad\qquad \land \text{UNCHANGED } \langle input,\ newChan,\ next,\ sbit,\ ack\rangle
\end{array}
$$

$$
\begin{array}{l}
LoseMsg \triangleq \land \lor \land \exists\, i \in 1\,..\,Len(msgChan) : \\
\qquad\qquad\qquad\quad msgChan' = Remove(i,\ msgChan) \\
\qquad\qquad\qquad\quad \land \text{UNCHANGED } ackChan \\
\qquad\qquad\qquad \lor \land \exists\, i \in 1\,..\,Len(ackChan) : \\
\qquad\qquad\qquad\quad ackChan' = Remove(i,\ ackChan) \\
\qquad\qquad\qquad\quad \land \text{UNCHANGED } msgChan \\
\qquad\qquad \land \text{UNCHANGED } \langle input,\ output,\ newChan,\ next,\ sbit,\ ack,\ rbit,\ msg\rangle
\end{array}
$$

$$Next \triangleq Sender \lor Receiver \lor LoseMsg$$

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

END TRANSLATION

$$Checkfalse \triangleq output = \langle\rangle$$

$$Invcommunication \triangleq output \neq \langle\rangle \Rightarrow \forall\, i \in 1\,..\,Len(output) : output[i] = input[i]$$

$test \triangleq Invcommunication$