MODULE *InnerFIFO*

Taken from “Specifying systems”

EXTENDS *Naturals*, *Sequences*

CONSTANT *Message*

VARIABLE *in*, *out*, *q*

$$\begin{aligned}
&InChan \triangleq \text{INSTANCE}\ Channel\ \text{WITH}\ Data \leftarrow Message,\ chan \leftarrow in \\
&OutChan \triangleq \text{INSTANCE}\ Channel\ \text{WITH}\ Data \leftarrow Message,\ chan \leftarrow out
\end{aligned}$$

$$\begin{aligned}
TypeInvariant \triangleq\ & \land\ InChan!TypeInvariant \\
& \land\ OutChan!TypeInvariant \\
& \land\ q \in Seq(Message)
\end{aligned}$$

Init both channels and make sure the message queue is empty.

$$\begin{aligned}
Init \triangleq\ & \land\ InChan!Init \\
& \land\ OutChan!Init \\
& \land\ q = \langle\rangle
\end{aligned}$$

Send *msg* to the in channel.

$$\begin{aligned}
InSend(msg) \triangleq\ & \land\ InChan!Send(msg) \\
& \land\ \text{UNCHANGED}\ \langle out,\ q\rangle
\end{aligned}$$

Append the received message to the queue.

$$\begin{aligned}
BufReceive \triangleq\ & \land\ InChan!Receive \\
& \land\ q' = Append(q,\ in.val) \\
& \land\ \text{UNCHANGED}\ out
\end{aligned}$$

Send the message to out channel and remove from queue.

$$\begin{aligned}
BufSend \triangleq\ & \land\ q \neq \langle\rangle \\
& \land\ OutChan!Send(Head(q))\quad \text{Send the first element out to the out channel.} \\
& \land\ q' = Tail(q)\quad \text{The queue after the send doesn't have that element.} \\
& \land\ \text{UNCHANGED}\ in
\end{aligned}$$

Receive message from the out channel.

$$\begin{aligned}
OutReceive \triangleq\ & \land\ OutChan!Receive \\
& \land\ \text{UNCHANGED}\ \langle in,\ q\rangle
\end{aligned}$$

$$\begin{aligned}
Next \triangleq\ & \lor\ \exists\, msg \in Message : InSend(msg) \\
& \lor\ BufReceive \\
& \lor\ BufSend \\
& \lor\ OutReceive
\end{aligned}$$

$$Spec \triangleq Init \land \Box[Next]_{\langle in,\ out,\ q\rangle}$$

THEOREM $Spec \Rightarrow \Box TypeInvariant$