523   In the reaction controller in Subsection 9.1.6, it is supposed that the synchronizer receives digital data from the digitizer faster than requests from the controller. Now suppose that the controller is sometimes faster than the digitizer. Modify the synchronizer so that if two or more requests arrive in a row (before new digital data arrives), the same digital data will be sent in reply to each request.

After trying the question, scroll down to the solution.

§   This is really just a monitor for a shared variable. The synchronizer is specified as follows.

$synchronizer \ =$

$($ $(\sqrt{digitaldata} \lor \mathcal{T}digitaldata_{\textit{ndigitaldata}} \leq \mathcal{T}request_{\textit{nrequest}}) \land digitaldata?$

$\lor (\sqrt{request} \lor \mathcal{T}request_{\textit{nrequest}} \leq \mathcal{T}digitaldata_{\textit{ndigitaldata}}) \land (request?.\ reply!digitaldata\ ).$

$synchronizer$

The synchronizer can be implemented as follows.

$synchronizer \quad \Longleftarrow \quad$ **if** $\sqrt{digitaldata}$ **then** $digitaldata?$ **else** ok **fi**.

$\qquad\qquad\qquad\qquad$ **if** $\sqrt{request}$ **then** $request?.\ reply!\ digitaldata$ **else** ok **fi**.

$\qquad\qquad\qquad\qquad$ $synchronizer$

To avoid the startup problem of replying to a request for data before any has been read, we might define

$startup \quad = \quad$ **if** $\sqrt{digitaldata}$ **then** $synchronizer$

$\qquad\qquad\qquad$ **else** $\quad$ **if** $\sqrt{request}$ **then** $request?.\ reply!$ "no data yet" **else** ok **fi**.

$\qquad\qquad\qquad\qquad\quad$ $startup$ **fi**