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.

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

\[
\text{synchronizer} = ( (\sqrt{\text{digitaldata}} \lor \text{digitaldata} \preceq \text{request}) \land \text{digitaldata}
\lor (\sqrt{\text{request}} \lor \text{request} \preceq \text{digitaldata}) \land \text{request} \Rightarrow \text{digitaldata}. \\
\text{synchronizer}
\]

The synchronizer can be implemented as follows.

\[
\text{synchronizer} \leftarrow \begin{cases} \\
\text{if } \sqrt{\text{digitaldata}} \text{ then digitaldata} \Rightarrow \text{digitaldata} \text{ else ok fi.} \\
\text{if } \sqrt{\text{request}} \text{ then request}. \Rightarrow \text{reply! digitaldata} \text{ else ok fi.} \\
\text{synchronizer}
\end{cases}
\]

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

\[
\text{startup} = \begin{cases} \\
\text{if } \sqrt{\text{digitaldata}} \text{ then synchronizer} \\
\text{else if } \sqrt{\text{request}} \text{ then request}. \Rightarrow \text{reply! “no data yet” else ok fi.}
\end{cases}
\text{startup fi}
\]