6.4.3 Equivalence

Given the above reassurance that the implementation is apparently behaving in the right way, the equivalence proof between behaviours for the two versions of the cell may be completed. The proof in the HOL system devolves into a case analysis on time, and results in a theorem of the usual form:

$$
\begin{array}{l}
\vdash\ \forall clck\ outp\ ci\ co\ now.\\
\quad \lambda t.\ \left(\begin{array}{l}
\texttt{if}\ \big((t \geq now+1) \wedge clck \wedge ci\big)\ \texttt{then}\\
\quad \big\{(\mathrm{clck}, clck), (\mathrm{outp}, \neg outp), (\mathrm{ci}, ci), \big(\mathrm{co}, (ci \wedge \neg outp)\big)\big\}\\
\texttt{elseif}\ (t \geq now)\ \texttt{then}\\
\quad \big\{(\mathrm{clck}, clck), (\mathrm{outp}, outp), (\mathrm{ci}, ci), (\mathrm{co}, co)\big\}\\
else\ \{\ \}
\end{array}\right)\\
\qquad\qquad\qquad\qquad \overset{\tau}{=}\\
\qquad\qquad\qquad\qquad \underset{\{\mathrm{clck,outp,ci,o}\}}{}\\
\quad \lambda t.\ \texttt{if}\ \big((t \geq now+1) \wedge clck \wedge ci\big)\ \texttt{then}\\
\qquad \left\{\begin{array}{l}
(\mathrm{clck}, clck), \big(\mathrm{outp}, \big((ci \vee outp) \wedge (\neg ci \vee \neg outp)\big)\big),\\
(\mathrm{ci}, ci), \big(\mathrm{co}, (ci \wedge \neg outp)\big), (\mathrm{new}, outp)
\end{array}\right\}\\
\qquad \texttt{elseif}\ (t \geq now)\ \texttt{then}\\
\qquad \left\{\begin{array}{l}
(\mathrm{clck}, clck), (\mathrm{outp}, outp), (\mathrm{ci}, ci), (\mathrm{co}, co),\\
\big(\mathrm{new}, \big((ci \vee outp) \wedge (\neg ci \vee \neg outp)\big)\big)
\end{array}\right\}\\
\qquad \texttt{else}\ \{\ \}
\end{array}
$$

$$
\begin{array}{l}
\vdash \forall now\ clck\ outp\ ci\ co. \\
\quad \texttt{let}\ \Pi = \ldots\ \texttt{in}\ \mu \vdash \langle \Pi, \lambda t.\{\} \rangle \xrightarrow{sim} beh = \\
\quad beh = \lambda t.\ \texttt{if}\ \big((t \geq now + 1) \wedge clck \wedge ci\big)\ \texttt{then} \\
\qquad \left\{ \begin{array}{l} (\text{clck}, clck), \big(\text{outp}, \big((ci \vee outp) \wedge (\neg ci \vee \neg outp)\big)\big), \\ (\text{ci}, ci), \big(\text{co}, (ci \wedge \neg outp)\big), (\text{new}, outp) \end{array} \right\} \\
\qquad \texttt{elseif}\ (t \geq now)\ \texttt{then} \\
\qquad \left\{ \begin{array}{l} (\text{clck}, clck), (\text{outp}, outp), (\text{ci}, ci), (\text{co}, co), \\ \big(\text{new}, \big((ci \vee outp) \wedge (\neg ci \vee \neg outp)\big)\big) \end{array} \right\} \\
\qquad \texttt{else}\ \{\}
\end{array}
$$

Since the behaviour of the device contains longer Boolean expressions than those of earlier examples, it is not intuitively clear that the right result has been obtained. One can gain some confidence in the derived behaviour by specialising the variables which represent the values of the signals. The interesting behaviour only occurs when the clock is high. So by specialising the variable *clck* to high, the conditional is reduced to:

$$
\begin{array}{l}
\lambda t.\ \texttt{if}\ \big((t \geq now + 1) \wedge ci\big)\ \texttt{then} \\
\qquad \left\{ \begin{array}{l} (\text{clck}, \mathsf{T}), \big(\text{outp}, \big((ci \vee outp) \wedge (\neg ci \vee \neg outp)\big)\big), \\ (\text{ci}, ci), \big(\text{co}, (ci \wedge \neg outp)\big), (\text{new}, outp) \end{array} \right\} \\
\quad \texttt{elseif}\ (t \geq now)\ \texttt{then} \\
\qquad \left\{ \begin{array}{l} (\text{clck}, \mathsf{T}), (\text{outp}, outp), (\text{ci}, ci), (\text{co}, co), \\ \big(\text{new}, \big((ci \vee outp) \wedge (\neg ci \vee \neg outp)\big)\big) \end{array} \right\}
\end{array}
$$