combine-fifo+[11,40]

$\forall es$:ES, C, T:Type, S_1, S_2, Ack_1, Ack_2:($C\to C\to \mathrm{E}\to \mathbb{P}$), R_1, Req_1, R_2, Req_2:($C\to \mathrm{E}\to \mathbb{P}$),
$codes_1$:(j,i:$C\to e$:{x:E| $S_1(j,i,x)$} →state@loc(e)→T),
$codes_2$:(j,i:$C\to e$:{x:E| $S_2(j,i,x)$} →state@loc(e)→T),
$decodes_1$:(i:$C\to e$:{x:E| $R_1(i,x)$} →state@loc(e)→T),
$decodes_2$:(i:$C\to e$:{x:E| $R_2(i,x)$} →state@loc(e)→T), dec_R_1:(i:$C\to e$:E→Dec($R_1(i,e)$)).
($\forall i$:C, e:E. ¬($R_1(i,e)$ & $R_2(i,e)$))
⇒ ($\forall dec\_S_1$:(j,i:$C\to e$:E→Dec($S_1(j,i,e)$)).
($\forall j$, j', i:C, e:E. ¬($S_1(j,i,e)$ & $S_2(j',i,e)$))
⇒ ($\forall i$:C, e:E. Dec($\exists j$:C. ($S_1(j,i,e)$)))
⇒ ($\forall i$:C, e:E. Dec($\exists j$:C. ($S_2(j,i,e)$)))
⇒ ($\forall i$:C, e:E. Dec($R_2(i,e)$))
⇒ forclients C sends FIFO
from j to i via (S_1[j,i],$codes_1$)
receives at i via (R_1[i],$decodes_1$)
requests Req_1[j] are acknowledged by Ack_1[j,i]
⇒ forclients C sends FIFO
from j to i via (S_2[j,i],$codes_2$)
receives at i via (R_2[i],$decodes_2$)
requests Req_2[j] are acknowledged by Ack_2[j,i]
⇒ switch between fifo+ send $S_1(j,i,e)$
request $Req_1(i,e)$
acknowledge $Ack_1(j,i,e)$ and
send $S_2(j,i,e)$ request $Req_2(i,e)$ acknowledge $Ack_2(j,i,e)$
⇒ forclients C sends FIFO
from j to i via ($\lambda j,i,e.\ (S_1(j,i,e)) \vee (S_2(j,i,e))$[j,i],[$S_1$? $codes_1$: $codes_2$])
receives at i via ($\lambda i,e.\ (R_1(i,e)) \vee (R_2(i,e))$[i],[$R_1$? $decodes_1$: $decodes_2$]))

---

http://www.nuprl.org/FDLcontent/p0_963683_/p31_788864_{combine-fifo!}.html