

This is a specification for the *Gemini* Protocol. It describes the behavior of the submodules that make up *Gemini*'s two Top of Rack (*TOR*) solution for reliability. The purpose of the specification is to find all possible failure scenarios in the algorithm by exploring reachable states simulating transitions in the submodules' state machines.

EXTENDS *FiniteSets*

VARIABLES

torA,

torB,

mux Which *ToR* the *MUX* cable itself is pointing to

LinkProber knows whether the *TOR* it's hosted in should be active or standby by listening to the active *ToR*'s heartbeat that is sent to both *ToR*'s. It knows this because this *ICMP* "heartbeat" has the name or *MAC* address of the active *ToR*.

In the specification, this is represented by the *heartbeatSender* variable since the standby *ToR*'s heartbeat will be dropped and never listened to.

vars \triangleq $\langle \textit{torA}, \textit{torB}, \textit{mux} \rangle$

T \triangleq $\{ \text{"torA"}, \text{"torB"} \}$

Link *Prober* (page 9)

LPStates \triangleq $\{ \text{"LPActive"}, \text{"LPStandby"}, \text{"LPWait"}, \text{"LPUnknown"} \}$

Link *Stat* (page 10)

LinkStates \triangleq $\{ \text{"LinkUp"}, \text{"LinkDown"} \}$

Mux *State* (page 12)

MuxStates \triangleq $\{ \text{"MuxActive"}, \text{"MuxStandby"}, \text{"MuxWait"}, \text{"MuxUnknown"} \}$

MUX_XCVRD_ (page 11)

XCVRDStates \triangleq $\{ \text{"switch"}, \text{"check"}, \text{"-"} \}$

ToR \triangleq

[*dead* : BOOLEAN ,
name : *T*,
xcvrd : *XCVRDStates*,
heartbeat : $\{ \text{"on"}, \text{"off"} \}$,
heartbeatIn : SUBSET (*T* \cup $\{ \text{"noResponse"} \}$),
linkProber : *LPStates*,
linkState : *LinkStates*,
muxState : *MuxStates*]

"Goal" state for a *ToR*.

ActiveTor \triangleq

[*dead* : $\{ \text{FALSE} \}$,
name : *T*,
xcvrd : $\{ \text{"-"} \}$,

$heartbeat : \{ \text{"on"} \},$
 $heartbeatIn : \text{SUBSET } (T \cup \{ \text{"noResponse"} \}),$
 $linkProber : \{ \text{"LPActive"} \},$
 $linkState : \{ \text{"LinkUp"} \},$
 $muxState : \{ \text{"MuxActive"} \}$

$TypeOK \triangleq$
 $\wedge \quad torA \in ToR$
 $\wedge \quad torB \in ToR$
 $\wedge \quad mux \in [active : T, next : T]$

$Init \triangleq$
 $LET \quad InitialTor(name) \triangleq$
 $\quad [dead \quad \mapsto FALSE,$
 $\quad \quad name \quad \mapsto name,$
 $\quad \quad xcvr \quad \mapsto \text{"check"},$
 $\quad \quad heartbeat \mapsto \text{"on"},$
 $\quad \quad heartbeatIn \mapsto \{ \},$
 $\quad \quad linkProber \mapsto \text{"LPUnknown"},$
 $\quad \quad linkState \mapsto \text{"LinkDown"},$
 $\quad \quad muxState \mapsto \text{"MuxWait"}]$
 $IN \quad \wedge \quad mux \in \{ f \in [active : T, next : T] : f.active = f.next \}$
 $\quad \wedge \quad torA = InitialTor(\text{"torA"})$
 $\quad \wedge \quad torB = InitialTor(\text{"torB"})$

State machine and transition table pages 12 & 13 of the *Powerpoint* presentation as of 08/25/2022

XCVRD daemon described on page 11 of the *Powerpoint* presentation as of 08/25/2022

<https://microsoft-my.sharepoint.com/:u:/p/t-gegranados/ERThXZdF5MVFusk2rP-PF0cBGguDR3Rt9yJ3WxxwAt0hpg?e>

Merged *LinkWait* and *MuxWait* on *Powerpoint* slide 13 into *Wait*

MuxState State Transitions depend on *LinkManager*'s decisions and *XCVRD* responses when checking or switching the *MuxCable*'s direction

$TRIGGER_LINKMANAGER_CHECK(t) \triangleq$

Sends check request to *MUX* via *XCVRD*. Transitions *muxState* to *MuxWait*. Defined to use in subsequent *MuxState* actions.

$t' = [t \text{ EXCEPT } !.muxState = \text{"MuxWait"}, !.xcvr = \text{"check"}]$

$TRIGGER_LINKMANAGER_SWITCH(t, target) \triangleq$

Sends write request to *MUX* via *XCVRD*. Transitions *muxState* to *MuxWait* and *linkProber* to *LPWait* target refers to *ToR* the *Mux* should point to. Defined to use in subsequent *MuxState* actions.

$\wedge \quad t' = [t \text{ EXCEPT } !.muxState = \text{"MuxWait"}, !.xcvr = \text{"switch"}, !.linkProber = \text{"LPWait"}]$
 $\wedge \quad mux' = [mux \text{ EXCEPT } !.next = target.name]$

$$\begin{aligned}
& EXEC_LINKMANAGER_CHECK(t, otherTor) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle mux, otherTor \rangle \\
& \quad \wedge t.xcvrd = \text{"check"} \\
& \quad \wedge \vee \wedge mux.active = t.name \\
& \quad \quad \wedge t' = [t \text{ EXCEPT } !.muxState = \text{"MuxActive"}, !.heartbeat = \text{"on"}, !.xcvrd = \text{"-"}] \\
& \quad \vee \wedge mux.active \neq t.name \\
& \quad \quad \wedge t' = [t \text{ EXCEPT } !.muxState = \text{"MuxStandby"}, !.heartbeat = \text{"on"}, !.xcvrd = \text{"-"}] \\
& \quad \vee \wedge t' = [t \text{ EXCEPT } !.muxState = \text{"MuxUnknown"}, !.heartbeat = \text{"on"}, !.xcvrd = \text{"-"}] \\
& \quad \vee \wedge t.muxState = \text{"MuxWait"} \\
& \quad \quad \wedge t' = [t \text{ EXCEPT } !.muxState = \text{"MuxStandby"}, !.xcvrd = \text{"-"}] MUX_XCVRD_FAIL
\end{aligned}$$

$$\begin{aligned}
& EXEC_LINKMANAGER_SWITCH(t, otherTor) \triangleq \\
& \quad \text{Writing Switch direction} \\
& \quad \wedge \text{UNCHANGED } otherTor \\
& \quad \wedge t.xcvrd = \text{"switch"} \\
& \quad \wedge t.muxState = \text{"MuxWait"} \\
& \quad \wedge t.linkProber = \text{"LPWait"} \\
& \quad \wedge mux' = [mux \text{ EXCEPT } !.active = mux.next] \\
& \quad \wedge \vee \wedge mux.next = t.name \\
& \quad \quad \wedge t' = [t \text{ EXCEPT } !.muxState = \text{"MuxActive"}, !.heartbeat = \text{"on"}, !.xcvrd = \text{"-"}] \\
& \quad \vee \wedge mux.next \neq t.name \\
& \quad \quad \wedge t' = [t \text{ EXCEPT } !.muxState = \text{"MuxStandby"}, !.heartbeat = \text{"on"}, !.xcvrd = \text{"-"}] \\
& \quad \vee \wedge t' = [t \text{ EXCEPT } !.muxState = \text{"MuxUnknown"}, !.heartbeat = \text{"on"}, !.xcvrd = \text{"-"}] \\
& \quad \vee \wedge t.muxState = \text{"MuxWait"} \\
& \quad \quad \wedge t' = [t \text{ EXCEPT } !.muxState = \text{"MuxStandby"}, !.xcvrd = \text{"-"}] MUX_XCVRD_FAIL
\end{aligned}$$

$$\begin{aligned}
& MuxStateActive(t, otherTor) \triangleq \\
& \quad \wedge t.muxState = \text{"MuxActive"} \\
& \quad \wedge \vee \wedge t.linkState = \text{"LinkUp"} \\
& \quad \quad \text{LinkUp MuxStateActive Row} \\
& \quad \quad \wedge \vee \wedge t.linkProber \in \{\text{"LPStandby"}, \text{"LPUnknown"}\} \\
& \quad \quad \quad \wedge TRIGGER_LINKMANAGER_CHECK(t) \\
& \quad \quad \vee \wedge t.linkProber = \text{"LPWait"} \\
& \quad \quad \quad \text{Check and suspend heartbeat} \\
& \quad \quad \quad \wedge t' = [t \text{ EXCEPT } !.muxState = \text{"MuxWait"}, !.xcvrd = \text{"check"}, !.heartbeat = \text{"off"}] \\
& \quad \wedge \text{UNCHANGED } \langle mux, otherTor \rangle \\
& \quad \vee \wedge t.linkState = \text{"LinkDown"} \\
& \quad \quad \text{Switch to Standby} \\
& \quad \quad \wedge TRIGGER_LINKMANAGER_SWITCH(t, otherTor) \\
& \quad \quad \wedge \text{UNCHANGED } otherTor
\end{aligned}$$

$$\begin{aligned}
& MuxStateStandby(t, otherTor) \triangleq \\
& \quad \wedge t.muxState = \text{"MuxStandby"}
\end{aligned}$$

$$\begin{aligned}
& \wedge \vee \wedge t.linkState = \text{"LinkUp"} \\
& \quad LinkUp \ MuxStateStandby \ Row \\
& \quad \wedge \vee \wedge t.linkProber \in \{ \text{"LPActive"}, \text{"LPWait"} \} \\
& \quad \quad \wedge TRIGGER_LINKMANAGER_CHECK(t) \\
& \quad \quad \wedge UNCHANGED \langle mux, otherTor \rangle \\
& \quad \vee \wedge t.linkProber = \text{"LPUnknown"} \\
& \quad \quad \text{Switch to Active} \\
& \quad \quad \wedge TRIGGER_LINKMANAGER_SWITCH(t, t) \\
& \quad \quad \wedge UNCHANGED \ otherTor \\
& \vee \wedge t.linkState = \text{"LinkDown"} \\
& \quad LinkDown \ MuxStateStandby \ Row \\
& \quad \wedge t.linkProber \in \{ \text{"LPUnknown"}, \text{"LPWait"} \} \\
& \quad \wedge TRIGGER_LINKMANAGER_CHECK(t) \\
& \quad \wedge UNCHANGED \langle mux, otherTor \rangle \\
\\
MuxStateUnknown(t, otherTor) & \triangleq \\
& \wedge t.muxState = \text{"MuxUnknown"} \\
& \wedge \vee \wedge t.linkState = \text{"LinkUp"} \\
& \quad LinkUp \ MuxStateStandby \ Row \\
& \quad \wedge TRIGGER_LINKMANAGER_CHECK(t) \\
& \vee \wedge t.linkState = \text{"LinkDown"} \\
& \quad LinkDown \ MuxStateStandby \ Row \\
& \quad \wedge t.linkProber \in \{ \text{"LPUnknown"}, \text{"LPWait"} \} \\
& \quad \wedge TRIGGER_LINKMANAGER_CHECK(t) \\
& \wedge UNCHANGED \langle mux, otherTor \rangle \\
\\
MuxStateWait(t, otherTor) & \triangleq \\
& \quad TODO \text{ Specify receiving } XCVRD \text{ Response} \\
& \wedge t.muxState = \text{"MuxWait"} \\
& \wedge \vee \wedge EXEC_LINKMANAGER_CHECK(t, otherTor) \\
& \quad \vee \wedge EXEC_LINKMANAGER_SWITCH(t, otherTor) \\
\\
MuxState(t, otherTor) & \triangleq \\
& \wedge \neg t.dead \\
& \wedge \vee MuxStateActive(t, otherTor) \\
& \quad \vee MuxStateStandby(t, otherTor) \\
& \quad \vee MuxStateUnknown(t, otherTor) \\
& \quad \vee MuxStateWait(t, otherTor)
\end{aligned}$$

$$\begin{aligned}
MuxXCVRD & \triangleq \\
& \setminus * \\
& \wedge UNCHANGED \langle torA, torB \rangle \\
& \wedge mux' = [mux \text{ EXCEPT } !.active = mux.next]
\end{aligned}$$

State machine page 10 of the *Powerpoint* presentation as of 08/25/2022

$$\begin{aligned} \text{LinkState}(t, \text{otherTor}) &\triangleq \\ &\wedge \neg t.\text{dead} \\ &\wedge t.\text{linkState} = \text{"LinkDown"} \quad \text{unnecessary, because going from LinkUp to LinkUp is just (finite) stuttering. However} \\ &\wedge \text{UNCHANGED } \langle \text{otherTor}, \text{mux} \rangle \\ &\wedge t' = [t \text{ EXCEPT } !.\text{linkState} = \text{"LinkUp"}] \end{aligned}$$

State machine page 09 of the *Powerpoint* presentation as of 08/25/2022

https://microsoft-my.sharepoint.com/:u:/p/zhangjing/EclAzBSCq_5KuugbbpyUlMQB1RS_X9nibOrM1PjT8wM_uw?e=

$$\begin{aligned} \text{SendHeartbeat}(t) &\triangleq \\ &\wedge \neg t.\text{dead} \\ &\wedge t.\text{linkState} = \text{"LinkUp"} \\ &\wedge t.\text{heartbeat} = \text{"on"} \\ &\text{Active } ToR \text{ sends heartbeat to server. } MUX \text{ duplicates packet and sends it to both } ToR\text{'s} \\ &\wedge \text{mux.active} = t.\text{name} \quad \text{The } MUX \text{ will drop traffic from } ToR \text{ if it is not pointing to it} \\ &\wedge \text{torA}' = [\text{torA} \text{ EXCEPT } !.\text{heartbeatIn} = @ \cup \{t.\text{name}\}] \\ &\wedge \text{torB}' = [\text{torB} \text{ EXCEPT } !.\text{heartbeatIn} = @ \cup \{t.\text{name}\}] \\ &\wedge \text{UNCHANGED } \langle \text{mux} \rangle \\ \\ \text{LinkProberWait}(t, \text{otherTor}) &\triangleq \\ &\wedge \text{UNCHANGED } \langle \text{otherTor}, \text{mux} \rangle \\ &\wedge t.\text{linkState} = \text{"LinkUp"} \\ &\wedge t.\text{linkProber} = \text{"LPWait"} \\ &\wedge \exists \text{heartbeat} \in t.\text{heartbeatIn} : \\ &\quad \vee \wedge t.\text{name} = \text{heartbeat} \\ &\quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{linkProber} = \text{"LPActive"}, !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \\ &\quad \vee \wedge \text{otherTor.name} = \text{heartbeat} \\ &\quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{linkProber} = \text{"LPStandby"}, !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \\ &\quad \vee \wedge \text{"noResponse"} = \text{heartbeat} \\ &\quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \end{aligned}$$

$$\begin{aligned} \text{LinkProberUnknown}(t, \text{otherTor}) &\triangleq \\ &\wedge \text{UNCHANGED } \langle \text{otherTor}, \text{mux} \rangle \\ &\wedge t.\text{linkState} = \text{"LinkUp"} \\ &\wedge t.\text{linkProber} = \text{"LPUnknown"} \\ &\wedge \exists \text{heartbeat} \in t.\text{heartbeatIn} : \\ &\quad \vee \wedge t.\text{name} = \text{heartbeat} \\ &\quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{linkProber} = \text{"LPActive"}, !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \\ &\quad \vee \wedge \text{otherTor.name} = \text{heartbeat} \\ &\quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{linkProber} = \text{"LPStandby"}, !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \\ &\quad \vee \wedge \text{"noResponse"} = \text{heartbeat} \\ &\quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \end{aligned}$$

$$\begin{aligned}
& \text{LinkProberStandby}(t, \text{otherTor}) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle \text{otherTor}, \text{mux} \rangle \\
& \quad \wedge t.\text{linkState} = \text{"LinkUp"} \\
& \quad \wedge t.\text{linkProber} = \text{"LPStandby"} \\
& \quad \wedge \exists \text{heartbeat} \in t.\text{heartbeatIn} : \\
& \quad \quad \vee \wedge t.\text{name} = \text{heartbeat} \\
& \quad \quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{linkProber} = \text{"LPActive"}, !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \\
& \quad \quad \vee \wedge \text{otherTor.name} = \text{heartbeat} \\
& \quad \quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \\
& \quad \quad \vee \wedge \text{"noResponse"} = \text{heartbeat} \\
& \quad \quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{linkProber} = \text{"LPUnknown"}, !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \\
\\
& \text{LinkProberActive}(t, \text{otherTor}) \triangleq \\
& \quad \wedge \text{UNCHANGED } \langle \text{otherTor}, \text{mux} \rangle \\
& \quad \wedge t.\text{linkState} = \text{"LinkUp"} \\
& \quad \wedge t.\text{linkProber} = \text{"LPActive"} \\
& \quad \wedge \exists \text{heartbeat} \in t.\text{heartbeatIn} : \\
& \quad \quad \vee \wedge t.\text{name} = \text{heartbeat} \\
& \quad \quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \\
& \quad \quad \vee \wedge \text{otherTor.name} = \text{heartbeat} \\
& \quad \quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{linkProber} = \text{"LPStandby"}, !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \\
& \quad \quad \vee \wedge \text{"noResponse"} = \text{heartbeat} \\
& \quad \quad \quad \wedge t' = [t \text{ EXCEPT } !.\text{linkProber} = \text{"LPUnknown"}, !.\text{heartbeatIn} = @ \setminus \{\text{heartbeat}\}] \\
\\
& \text{LinkProber}(t, \text{otherTor}) \triangleq \\
& \quad \wedge \neg t.\text{dead} \\
& \quad \wedge \vee \text{LinkProberActive}(t, \text{otherTor}) \\
& \quad \quad \vee \text{LinkProberStandby}(t, \text{otherTor}) \\
& \quad \quad \vee \text{LinkProberWait}(t, \text{otherTor}) \\
& \quad \quad \vee \text{LinkProberUnknown}(t, \text{otherTor})
\end{aligned}$$

$$\begin{aligned}
& \text{System} \triangleq \\
& \quad \text{Mux handling a switch or check command.} \\
& \quad \vee \text{EXEC_LINKMANAGER_SWITCH}(\text{torA}, \text{torB}) \\
& \quad \vee \text{EXEC_LINKMANAGER_SWITCH}(\text{torB}, \text{torA}) \\
& \quad \vee \text{EXEC_LINKMANAGER_CHECK}(\text{torA}, \text{torB}) \\
& \quad \vee \text{EXEC_LINKMANAGER_CHECK}(\text{torB}, \text{torA}) \\
& \quad \text{XCVRD and LinkMgrd} \\
& \quad \vee \text{MuxState}(\text{torA}, \text{torB}) \\
& \quad \vee \text{MuxState}(\text{torB}, \text{torA}) \\
& \quad \text{ToR periodically send heartbeats via the mux to the server.} \\
& \quad \vee \text{SendHeartbeat}(\text{torA}) \\
& \quad \vee \text{SendHeartbeat}(\text{torB})
\end{aligned}$$

ToR receives heartbeat and triggers appropriate transition in *LinkProber*

$\vee \text{LinkProber}(torA, torB)$

$\vee \text{LinkProber}(torB, torA)$

Notification from the kernel that a physical link (*L1*) came up.

$\vee \text{LinkState}(torA, torB)$

$\vee \text{LinkState}(torB, torA)$

FailHeartbeat \triangleq

Sender fails to send heartbeat to *ToR*'s making them go into unknown state

$\wedge \vee \wedge \exists \text{heartbeat} \in \text{SUBSET } torA.\text{heartbeatIn} :$
 $\wedge torA' = [torA \text{ EXCEPT } !.\text{heartbeatIn} = \text{heartbeat}]$

$\wedge \text{UNCHANGED } torB$

$\vee \wedge \exists \text{heartbeat} \in \text{SUBSET } torB.\text{heartbeatIn} :$
 $\wedge torB' = [torB \text{ EXCEPT } !.\text{heartbeatIn} = \text{heartbeat}]$

$\wedge \text{UNCHANGED } torA$

$\wedge \text{UNCHANGED } mux$

FailMux \triangleq

Failure Action for inconsistent *MUX* States with *MuxCable* State

$\wedge mux' \in [active : T, next : T]$

$\wedge \text{UNCHANGED } \langle torA, torB \rangle$

FailTor(*t*, *otherTor*) \triangleq

$\wedge t' = [t \text{ EXCEPT } !.dead = \text{TRUE}]$

$\wedge \text{UNCHANGED } \langle otherTor, mux \rangle$

FailXCVRD(*t*, *otherTor*) \triangleq

$\wedge \text{UNCHANGED } \langle otherTor, mux \rangle$

According to *Vaibhav Dahiya*, the *mux* returns “Unknown” in case of failure, which, subsequently, causes the *ToR* to go to “Standby”.

$\wedge t' = [t \text{ EXCEPT } !.xcvrd = \text{“-”}]$

FailLinkState(*t*, *otherTor*) \triangleq

$\wedge \neg t.dead$

$\wedge \text{UNCHANGED } \langle otherTor, mux \rangle$

$\wedge t' = [t \text{ EXCEPT } !.linkState = \text{“LinkDown”}]$

Environment \triangleq

$\vee \text{FailMux}$

$\vee \text{FailHeartbeat}$

$\vee \text{FailTor}(torA, torB)$

$\vee \text{FailTor}(torB, torA)$

$\vee \text{FailXCVRD}(torA, torB)$

$\vee \text{FailXCVRD}(torB, torA)$

$$\begin{aligned} & \vee \text{FailLinkState}(\text{torA}, \text{torB}) \\ & \vee \text{FailLinkState}(\text{torB}, \text{torA}) \end{aligned}$$

$$\begin{aligned} \text{Next} & \triangleq \\ & \vee \text{Environment} \\ & \vee \text{System} \end{aligned}$$

$$\begin{aligned} \text{Spec} & \triangleq \\ & \text{Init} \wedge \Box[\text{Next}]_{\text{vars}} \wedge \text{WF}_{\text{vars}}(\text{System}) \end{aligned}$$

$$\begin{aligned} \text{AtMostOneActive} & \triangleq \\ & \Box \neg (\text{torA} \in \text{ActiveTor} \wedge \text{torB} \in \text{ActiveTor}) \end{aligned}$$

$$\begin{aligned} \text{RepeatedlyOneActive} & \triangleq \\ & \Box \Diamond (\exists t \in \{\text{torA}, \text{torB}\} : \neg t.\text{dead} \Rightarrow (\text{torA} \in \text{ActiveTor} \vee \text{torB} \in \text{ActiveTor})) \end{aligned}$$

$$\begin{aligned} \text{THEOREM } \text{Spec} & \Rightarrow \\ & \wedge \text{AtMostOneActive} \\ & \wedge \text{RepeatedlyOneActive} \end{aligned}$$

$$\begin{aligned} \text{Alias} & \triangleq \\ & [\\ & \quad \text{torA} \mapsto \text{torA}, \text{torB} \mapsto \text{torB}, \text{mux} \mapsto \text{mux}, \\ & \quad \text{active} \mapsto \{t.\text{name} : t \in \{t \in \{\text{torA}, \text{torB}\} : t \in \text{ActiveTor}\}\} \\ &] \end{aligned}$$
