1 Preliminary definitions

Definition 1 (General state similarity). *For a given* $sitpn \in SITPN$*, an* $\mathcal{H}$*-VHDL design* $d \in design$*, an elaborated design* $\Delta \in ElDesign$*, and a binder* $\gamma \in WM(sitpn, d)$*, an SITPN state* $s \in S(sitpn)$ *and a design state* $\sigma \in \Sigma$ *are similar, written* $\gamma \vdash s \sim \sigma$ *if*

1. $\forall p \in P, id_p \in Comps(\Delta)$ *s.t.* $\gamma(p) = id_p,\ \ s.M(p) = \sigma(id_p)(\texttt{s\_marking})$.
2. $\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,
   $\big(u(I_s(t)) = \infty \wedge s.I(t) \leq l(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(\texttt{s\_time\_counter})\big)$
   $\wedge\big(u(I_s(t)) = \infty \wedge s.I(t) > l(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)$
   $\wedge\big(u(I_s(t)) \neq \infty \wedge s.I(t) > u(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)$
   $\wedge\big(u(I_s(t)) \neq \infty \wedge s.I(t) \leq u(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(\texttt{s\_time\_counter})\big)$.
3. $\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t,\ s.reset_t(t) = \sigma(id_t)(\texttt{s\_reinit\_time\_counter})$.
4. $\forall c \in \mathcal{C}, id_c \in Ins(\Delta)$ *s.t.* $\gamma(c) = id_c,\ \ s.cond(c) = \sigma(id_c)$.
5. $\forall a \in \mathcal{A}, id_a \in Outs(\Delta)$ *s.t.* $\gamma(a) = id_a,\ \ s.ex(a) = \sigma(id_a)$.
6. $\forall f \in \mathcal{F}, id_f \in Outs(\Delta)$ *s.t.* $\gamma(f) = id_f,\ \ s.ex(f) = \sigma(id_f)$.

Definition 2 (Post rising edge state similarity). *For a given* $sitpn \in SITPN$*, an* $\mathcal{H}$*-VHDL design* $d \in design$*, an elaborated design* $\Delta \in ElDesign$*, and a binder* $\gamma \in WM(sitpn, d)$*, an SITPN state* $s \in S(sitpn)$ *and a design state* $\sigma \in \Sigma$ *are similar after a rising edge, written* $\gamma \vdash s \overset{\uparrow}{\sim} \sigma$ *iff*

1. $\forall p \in P, id_p \in Comps(\Delta)$ *s.t.* $\gamma(p) = id_p,\ \ s.M(p) = \sigma(id_p)(\texttt{s\_marking})$.
2. $\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,
   $\big(u(I_s(t)) = \infty \wedge s.I(t) \leq l(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(\texttt{s\_time\_counter})\big)$
   $\wedge\big(u(I_s(t)) = \infty \wedge s.I(t) > l(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)$
   $\wedge\big(u(I_s(t)) \neq \infty \wedge s.I(t) > u(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)$
   $\wedge\big(u(I_s(t)) \neq \infty \wedge s.I(t) \leq u(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(\texttt{s\_time\_counter})\big)$.
3. $\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t,\ s.reset_t(t) = \sigma(id_t)(\texttt{s\_reinit\_time\_counter})$.
4. $\forall a \in \mathcal{A}, id_a \in Outs(\Delta)$ *s.t.* $\gamma(a) = id_a,\ \ s.ex(a) = \sigma(id_a)$.
5. $\forall f \in \mathcal{F}, id_f \in Outs(\Delta)$ *s.t.* $\gamma(f) = id_f,\ \ s.ex(f) = \sigma(id_f)$.

Definition 3 (Full post rising edge state similarity). *For a given* $sitpn \in SITPN$*, an* $\mathcal{H}$*-VHDL design* $d \in design$*, an elaborated design* $\Delta \in ElDesign$*, and a binder* $\gamma \in WM(sitpn, d)$*, a clock cycle count* $\tau \in \mathbb{N}$*, and an SITPN execution environment* $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$*, an SITPN state* $s \in S(sitpn)$ *and a design state* $\sigma \in \Sigma$ *are fully similar after a rising edge happening at clock cycle count* τ*, written* $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$*, if* $\gamma \vdash s \overset{\uparrow}{\sim} \sigma$ *(Definition 2) and*

1. $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \in Sens(s.M) \Leftrightarrow \sigma(id_t)(\texttt{s\_enabled}) = \texttt{true}$.

2. $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \notin Sens(s.M) \Leftrightarrow \sigma(id_t)(\texttt{s\_enabled}) = \texttt{false}$.

3. $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,
$$\sigma(id_t)(\texttt{s\_condition\_combination}) = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}$$
where $conds(t) = \{c \in \mathcal{C} \mid \mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1\}$.

4. $\forall c \in \mathcal{C}, id_c \in Ins(\Delta)$ *s.t.* $\gamma(c) = id_c$, $\sigma(id_c) = E_c(\tau, c)$.

Definition 4 (Post falling edge state similarity)**.** *For a given* $sitpn \in SITPN$*, an* $\mathcal{H}$*-VHDL design* $d \in design$*, an elaborated design* $\Delta \in ElDesign$*, and a binder* $\gamma \in WM(sitpn, d)$*, an SITPN state* $s \in S(sitpn)$ *and a design state* $\sigma \in \Sigma$ *are similar after a falling edge, written* $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$*, if*

1. $\forall p \in P, id_p \in Comps(\Delta)$ *s.t.* $\gamma(p) = id_p$, $s.M(p) = \sigma(id_p)(\texttt{s\_marking})$.

2. $\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,
$\big(u(I_s(t)) = \infty \wedge s.I(t) \leq l(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(\texttt{s\_time\_counter})\big)$
$\wedge\big(u(I_s(t)) = \infty \wedge s.I(t) > l(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)$
$\wedge\big(u(I_s(t)) \neq \infty \wedge s.I(t) > u(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)$
$\wedge\big(u(I_s(t)) \neq \infty \wedge s.I(t) \leq u(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(\texttt{s\_time\_counter})\big)$.

3. $\forall c \in \mathcal{C}, id_c \in Ins(\Delta)$ *s.t.* $\gamma(c) = id_c$, $s.cond(c) = \sigma(id_c)$.

4. $\forall a \in \mathcal{A}, id_a \in Outs(\Delta)$ *s.t.* $\gamma(a) = id_a$, $s.ex(a) = \sigma(id_a)$.

5. $\forall f \in \mathcal{F}, id_f \in Outs(\Delta)$ *s.t.* $\gamma(f) = id_f$, $s.ex(f) = \sigma(id_f)$.

Definition 5 (Full post falling edge state similarity)**.** *For a given* $sitpn \in SITPN$*, an* $\mathcal{H}$*-VHDL design* $d \in design$*, an elaborated design* $\Delta \in ElDesign$*, and a binder* $\gamma \in WM(sitpn, d)$*, an SITPN state* $s \in S(sitpn)$ *and a design state* $\sigma \in \Sigma$ *are fully similar after a falling edge, written* $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$*, if* $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$ *(Definition 4) and*

1. $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \in Firable(s) \Leftrightarrow \sigma(id_t)(\texttt{s\_firable}) = \texttt{true}$.

2. $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \notin Firable(s) \Leftrightarrow \sigma(id_t)(\texttt{s\_firable}) = \texttt{false}$.

3. $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \in Fired(s) \Leftrightarrow \sigma(id_t)(\texttt{fired}) = \texttt{true}$.

4. $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \notin Fired(s) \Leftrightarrow \sigma(id_t)(\texttt{fired}) = \texttt{false}$.

5. $\forall p \in P, id_p \in Comps(\Delta)$ *s.t.* $\gamma(p) = id_p$, $\sum_{t \in Fired(s)} pre(p,t) = \sigma(id_p)(\texttt{s\_output\_token\_sum})$.

6. $\forall p \in P, id_p \in Comps(\Delta)$ *s.t.* $\gamma(p) = id_p$, $\sum_{t \in Fired(s)} post(t,p) = \sigma(id_p)(\texttt{s\_input\_token\_sum})$.

Definition 6 (Similar environments). *For a given sitpn $\in SITPN$, a $\mathcal{H}$-VHDL design $d \in design$, a design store $\mathcal{D} \in entity\text{-}id \nrightarrow design$, an elaborated version $\Delta \in ElDesign$ of design d, and a binder $\gamma \in WM(sitpn, d)$, the environment $E_p \in \mathbb{N} \to Ins(\Delta) \to value$, that yields the value of the primary input ports of Δ at a given simulation cycle, and the environment E_c, that yields the value of conditions of sitpn at a given execution cycle, are similar, written $\gamma \vdash E_p \overset{env}{=} E_c$, if for all $\tau \in \mathbb{N}$, $c \in \mathcal{C}$, $id_c \in Ins(\Delta)$ s.t. $\gamma(c) = id_c$, $E_p(\tau)(id_c) = E_c(\tau)(c)$.*

Definition 7 (Execution trace similarity). *For a given sitpn $\in SITPN$, a $\mathcal{H}$-VHDL design $d \in design$, an elaborated design $\Delta \in ElDesign$, and a binder $\gamma \in WM(sitpn, d)$, the execution trace $\theta_s \in \texttt{list}(S(sitpn))$ and the simulation trace $\theta_\sigma \in \texttt{list}(\Sigma)$ are similar if $\gamma \vdash \theta_s \overset{clk}{\sim} \theta_\sigma$ (where $clk \in \{\uparrow, \downarrow\}$) is derivable according to the following rules:*

$$\textsc{SimTraceNil}\ \frac{}{\gamma \vdash [\,] \overset{clk}{\sim} [\,]}\ clk \in \{\uparrow, \downarrow\} \qquad \textsc{SimTrace}\uparrow\ \frac{\gamma \vdash s \overset{\uparrow}{\sim} \sigma \qquad \gamma \vdash \theta_s \overset{\downarrow}{\sim} \theta_\sigma}{\gamma \vdash (s :: \theta_s) \overset{\uparrow}{\sim} (\sigma :: \theta_\sigma)} \qquad \textsc{SimTrace}\downarrow\ \frac{\gamma \vdash s \overset{\downarrow}{\sim} \sigma \qquad \gamma \vdash \theta_s \overset{\uparrow}{\sim} \theta_\sigma}{\gamma \vdash (s :: \theta_s) \overset{\downarrow}{\sim} (\sigma :: \theta_\sigma)}$$

Definition 8 (Full execution trace similarity). *For a given sitpn $\in SITPN$, a $\mathcal{H}$-VHDL design $d \in design$, an elaborated design $\Delta \in ElDesign(d, \mathcal{D}_\mathcal{H})$, and a binder $\gamma \in WM(sitpn, d)$, the execution trace $\theta_s \in \texttt{list}(S(sitpn))$ and the simulation trace $\theta_\sigma \in \texttt{list}(\Sigma)$ are fully similar, written $\gamma \vdash \theta_s \sim \theta_\sigma$, according to the following rules:*

$$\textsc{FullSimTraceNil}\ \frac{}{\gamma \vdash [\,] \sim [\,]} \qquad \textsc{FullSimTraceCons}\ \frac{\gamma \vdash s \sim \sigma \qquad \gamma \vdash \theta_s \overset{\uparrow}{\sim} \theta_\sigma}{\gamma \vdash (s :: \theta_s) \sim (\sigma :: \theta_\sigma)}$$

2 Correctness, behavior preservation, or semantic preservation theorem

Theorem 1 (Behavior preservation). *For all well-defined sitpn $\in SITPN$, $\mathcal{H}$-VHDL design $d \in design$, binder $\gamma \in WM(sitpn, d)$, clock cycle count $\tau \in \mathbb{N}$, execution environment $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$, execution trace $\theta_s \in \texttt{list}(S(sitpn))$ and maximal marking function $b \in P \to \mathbb{N}$ such that*

- *SITPN sitpn is transformed into the $\mathcal{H}$-VHDL design d and yields the binder γ: $\lfloor sitpn \rfloor_b = (d, \gamma)$*
- *SITPN sitpn is bounded through b: $\lceil sitpn \rceil^b$*
- *SITPN sitpn yields the execution trace θ_s after τ execution cycles in environment E_c: $E_c, \tau \vdash sitpn \xrightarrow{full} \theta_s$*

then there exist an elaborated design $\Delta \in ElDesign$ and a simulation trace $\theta_\sigma \in \texttt{list}(\Sigma)$ s.t. for all simulation environment $E_p \in \mathbb{N} \to Ins(\Delta) \to value$ verifying $\gamma \vdash E_p \overset{env}{=} E_c$ (simulation and execution environments are similar), we have:

- *In the context of the HILECOP design store $\mathcal{D}_{\mathcal{H}}$ and with an empty generic constant dimensioning function ($\emptyset$), design d elaborates into Δ and yields the simulation trace θ_σ after τ simulation cycles:* $\mathcal{D}_{\mathcal{H}}, \Delta, \emptyset, E_p, \tau \vdash \mathrm{d} \xrightarrow{full} \theta_\sigma$
- *Traces θ_s and θ_σ are fully similar:* $\theta_s \sim \theta_\sigma$

Proof.

Given a $sitpn \in SITPN$, a $d \in design$, a $\gamma \in WM(sitpn, d)$, a $\tau \in \mathbb{N}$, an $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$, a $\theta_s \in \texttt{list}(S(sitpn))$, and a $b \in P \to \mathbb{N}$, let us show that

$$\boxed{\exists \Delta, \theta_\sigma, \forall E_p,\ \gamma \vdash E_p \overset{env}{=} E_c \Rightarrow (\mathcal{D}_{\mathcal{H}}, \Delta, \emptyset, E_p, \tau \vdash \mathrm{d} \xrightarrow{full} \theta_\sigma) \wedge \theta_s \sim \theta_\sigma}$$

Appealing to Theorems 2 (p. 5), 3 (p. 5) and 4 (p. 5), let us take an elaborated design $\Delta \in ElDesign$, two design states $\sigma_e, \sigma_0 \in \Sigma$, and a simulation trace $\theta_\sigma \in \texttt{list}(\Sigma)$ such that:

– Δ is the elaborated version of design d, and σ_e is the default design state of Δ:
$\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$

– σ_0 is the initial simulation state: $\mathcal{D}_{\mathcal{H}}, \Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$

– Design d yields the simulation trace θ_σ after τ simulation cycles, starting from initial state σ_0:
$\mathcal{D}_{\mathcal{H}}, E_p, \Delta, \tau, \sigma_0 \vdash \mathrm{d.cs} \to \theta_\sigma$

Let us use this Δ and this θ_σ to prove the current goal. Given an E_p such that $\gamma \vdash E_p \overset{env}{=} E_c$, it remains to be proved that:

$$\boxed{(\mathcal{D}_{\mathcal{H}}, \Delta, \emptyset, E_p, \tau \vdash \mathrm{d} \xrightarrow{full} \theta_\sigma) \wedge \theta_s \sim \theta_\sigma}$$

First, we must prove that $\boxed{(\mathcal{D}_{\mathcal{H}}, \Delta, \emptyset, E_p, \tau \vdash \mathrm{d} \xrightarrow{full} \theta_\sigma)}$ holds. By definition of the $\mathcal{H}$-VHDL full simulation relation, we have:

$$\begin{aligned} \mathcal{D}_{\mathcal{H}}, \Delta, \emptyset, E_p, \tau \vdash \mathrm{d} \xrightarrow{full} \theta_\sigma \equiv \exists \sigma_e, \sigma_0 \in \Sigma(\Delta),\ & \mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e) \\ & \wedge \mathcal{D}_{\mathcal{H}}, \Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0 \\ & \wedge \mathcal{D}_{\mathcal{H}}, E_p, \Delta, \tau, \sigma_0 \vdash \mathrm{d.cs} \to \theta_\sigma \end{aligned} \tag{1}$$

Thus, it is equivalent to prove:

$$\boxed{\exists \sigma_e, \sigma_0 \text{ s.t. } \mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e) \wedge \mathcal{D}_{\mathcal{H}}, \Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0 \wedge \mathcal{D}_{\mathcal{H}}, E_p, \Delta, \tau, \sigma_0 \vdash \mathrm{d.cs} \to \theta_\sigma}$$

To prove the goal, let us use σ_e, $\sigma_0 \in \Sigma$ previously introduced by the invocation of Theorems 2, 3 and 4. Then, the three first points of the goal are previously assumed hypotheses.
Finally, appealing to Theorem 5, we can prove final point of the theorem, i.e. $\boxed{\theta_s \sim \theta_\sigma}$.

□

Theorem 2 (Elaboration). *For all well-defined* $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$ *and* $b \in P \to \mathbb{N}$ *such that*

- $\lfloor sitpn \rfloor_b = (d, \gamma)$

then there exists an elaborated design $\Delta \in ElDesign$ *and a design state* $\sigma_e \in \Sigma$ *s.t.* Δ *is the elaborated version of design* d*, and* σ_e *is the default design state of* Δ*:* $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$.

Theorem 3 (Initialization). *For all well-defined* $sitpn \in SITPN$, $d \in design$, $b \in P \to \mathbb{N}$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign$, $\sigma_e \in \Sigma(\Delta)$ *s.t.*

- $\lfloor sitpn \rfloor_b = (d, \gamma)$ *and* $\lceil sitpn \rceil^b$ *and* $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$

then there exists a design state $\sigma_0 \in \Sigma(\Delta)$ *s.t.* σ_0 *is the initial simulation state:* $\mathcal{D}_{\mathcal{H}}, \Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$.

Theorem 4 (Trace existence). *For all well-defined* $sitpn \in SITPN$, $d \in design$, $b \in P \to \mathbb{N}$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign$, $\sigma_e, \sigma_0 \in \Sigma$ *s.t.*

- $\lfloor sitpn \rfloor_b = (d, \gamma)$ *and* $\lceil sitpn \rceil^b$ *and* $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$ *and* $\mathcal{D}_{\mathcal{H}}, \Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$

then there exists a simulation trace $\theta_\sigma \in \texttt{list}(\Sigma)$ *such that for all simulation environment* $E_p \in \mathbb{N} \to Ins(\Delta) \to value$ *and simulation cycle count* $\tau \in \mathbb{N}$*, design* d *yields the simulation trace* θ_σ *after* τ *simulation cycles, starting from initial state* σ_0*:*
$\mathcal{D}_{\mathcal{H}}, E_p, \Delta, \tau, \sigma_0 \vdash \mathrm{d.cs} \to \theta_\sigma$

3 Trace similarity theorem

Definition 9 (HM2T hypotheses). *For all well-defined sitpn* $\in SITPN$*, bounding function* $b \in P \to \mathbb{N}$*,* $\mathcal{H}$*-VHDL design* $d \in design$*, binder* $\gamma \in WM(sitpn, d)$*, elaborated design* $\Delta \in ElDesign$*, default state* $\sigma_e \in \Sigma$*, simulation environment* $E_p \in \mathbb{N} \to (id \nrightarrow v)$*, and execution environment* $E_c \in \mathbb{N} \to (\mathcal{C} \to \mathbb{B})$*, assume that:*

1. *Taking the SITPN model sitpn and the bounding function b as inputs, the HM2T returns an output design d and a binder* γ*, written* $\texttt{sitpn2hvhdl}(sitpn, b) = \lfloor (d, \gamma) \rfloor$ *where* $\texttt{sitpn2hvhdl} \in SITPN \to (P \to \mathbb{N}) \nrightarrow (design \times WM(sitpn, d))$.
2. *sitpn is bounded through b, written* $\lceil sitpn \rceil^b$.
3. *In the context of the HILECOP design store* $\mathcal{D}_\mathcal{H}$ *and with an empty generic constant dimensioning function* ($\emptyset$)*, d is elaborated into* Δ *with a default state* σ_e*, written* $\mathcal{D}_\mathcal{H}, \emptyset \vdash d \xrightarrow{elab} \Delta, \sigma_e$.
4. *Simulation and execution environments are similar, written* $\gamma \vdash E_p \overset{env}{=} E_c$.

Theorem 5 (Full trace similarity). *For all well-defined sitpn* $\in SITPN$*, bounding function* $b \in P \to \mathbb{N}$*,* $\mathcal{H}$*-VHDL design* $d \in design$*, binder* $\gamma \in WM(sitpn, d)$*, default state* $\sigma_e \in \Sigma$*, simulation environment* $E_p \in \mathbb{N} \to (id \nrightarrow v)$*, execution environment* $E_c \in \mathbb{N} \to (\mathcal{C} \to \mathbb{B})$*,* $\tau \in \mathbb{N}$*, SITPN model trace* $\theta_s \in \texttt{list}(S(sitpn))$*, and* $\mathcal{H}$*-VHDL design trace* $\theta_\sigma \in \texttt{list}(\Sigma)$ *such that:*

1. $\texttt{sitpn2hvhdl}(sitpn, b) = \lfloor (d, \gamma) \rfloor$
2. $\lceil sitpn \rceil^b$
3. $\gamma \vdash E_p \overset{env}{=} E_c$
4. $E_c, \tau \vdash sitpn \xrightarrow{full} \theta_s$
5. $\mathcal{D}_\mathcal{H}, \emptyset, E_p, \tau \vdash d \xrightarrow{full} \theta_\sigma$

then $\gamma \vdash \theta_s \sim \theta_\sigma$.

Proof.

Proceeding by case analysis on the number of clock cycles τ, there are two cases. First $\tau = 0$, and then we must prove that the initial states are similar, which is true appealing to Lemma 1. Otherwise, $\tau > 0$ and then at least the first clock cycle is executed. Thanks to Lemmas 11 and 31, we can show that the states are similar during the first clock cycle. Then, we can reason by induction over τ to prove that the remnant of the execution traces are similar. We can appeal to Lemmas 21 and 31 to prove that states are similar during the induction step (corresponding to an arbitrary clock cycle step), and then use the induction hypothesis to complete the proof.

□

4 Similar initial states

Lemma 1 (Similar initial states)**.** *For all well-defined* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e \in \Sigma$, $E_p \in \mathbb{N} \to (id \nrightarrow v)$, *and* $E_c \in \mathbb{N} \to (\mathcal{C} \to \mathbb{B})$ *that verify the hypotheses of Definition 9, and for all* $\sigma_0, \sigma_i \in \Sigma$ *such that:*

- σ_0 *is the initial state of design* d*:*
  $\mathcal{D}, \Delta, \sigma_e \vdash d.beh \xrightarrow{cs_i} \sigma_i$ *and* $\mathcal{D}, \Delta, \sigma_i \vdash d.beh \overset{\rightsquigarrow}{\to} \sigma_0$

then $\gamma \vdash s_0 \approx \sigma_0$.

Proof.

By definition of the General state similarity relation, there are 6 points to prove.

1. $\forall p \in P, id_p \in Comps(\Delta)\ s.t.\ \gamma(p) = id_p,\ \ s_0.M(p) = \sigma_0(id_p)(\texttt{s\_marking})$.
2. $\forall t \in T_i, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
   $\big(u(I_s(t)) = \infty \wedge s_0.I(t) \leq l(I_s(t)) \Rightarrow s_0.I(t) = \sigma_0(id_t)(\texttt{s\_time\_counter})\big)$
   $\wedge\big(u(I_s(t)) = \infty \wedge s_0.I(t) > l(I_s(t)) \Rightarrow \sigma_0(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)$
   $\wedge\big(u(I_s(t)) \neq \infty \wedge s_0.I(t) > u(I_s(t)) \Rightarrow \sigma_0(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)$
   $\wedge\big(u(I_s(t)) \neq \infty \wedge s_0.I(t) \leq u(I_s(t)) \Rightarrow s_0.I(t) = \sigma_0(id_t)(\texttt{s\_time\_counter})\big)$.
3. $\forall t \in T_i, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,\ s_0.reset_t(t) = \sigma_0(id_t)(\texttt{s\_reinit\_time\_counter})$.
4. $\forall c \in \mathcal{C}, id_c \in Ins(\Delta)\ s.t.\ \gamma(c) = id_c,\ \ s_0.cond(c) = \sigma_0(id_c)$.
5. $\forall a \in \mathcal{A}, id_a \in Outs(\Delta)\ s.t.\ \gamma(a) = id_a,\ \ s_0.ex(a) = \sigma_0(id_a)$.
6. $\forall f \in \mathcal{F}, id_f \in Outs(\Delta)\ s.t.\ \gamma(f) = id_f,\ \ s_0.ex(f) = \sigma_0(id_f)$.

- Apply the Initial states equal marking lemma to solve 1.
- Apply the Initial states equal time counters lemma to solve 2.
- Apply the Initial states equal reset orders lemma to solve 3.
- Apply the Initial states equal condition values lemma to solve 4.
- Apply the Initial states equal action executions lemma to solve 5.
- Apply the Initial states equal function executions lemma to solve 6.

□

Definition 10 (Initial state hypotheses)**.** *Given an* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e, \sigma_0 \in \Sigma$, *assume that:*

- *SITPN sitpn is transformed into the design d and yields the binder γ:* $\lfloor sitpn \rfloor_b = (d, \gamma)$
- *Δ is the elaborated version of d, σ_e is the default state of Δ, i.e. the state of Δ where all signals are initialized to their default value:*
  $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$
- *σ_0 is the initial state of Δ:* $\Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$

4.1 Initial states and marking

Lemma 2 (Initial states equal marking)**.** *For all* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e, \sigma_0 \in \Sigma$ *that verify the hypotheses of Definition 10, then* $\forall p \in P, id_p \in Comps(\Delta)$ *s.t.* $\gamma(p) = id_p$, $s_0.M(p) = \sigma_0(id_p)(\texttt{s\_marking})$.

Proof.

Given a $p \in P$ and an $id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$, let us show that
$\boxed{s_0.M(p) = \sigma_0(id_p)(\texttt{s\_marking}).}$
By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$.
By property of the $\mathcal{H}$-VHDL initialization relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `marking` process defined in the place design architecture, we can deduce $\sigma_0(id_p)(\texttt{s\_marking}) = \sigma_0(id_p)(\texttt{initial\_marking})$.
Rewriting $\sigma_0(id_p)(\texttt{sm})$ as $\sigma_0(id_p)(\texttt{initial\_marking})$,
$\boxed{\sigma_0(id_p)(\texttt{initial\_marking}) = s_0.M(p).}$
By construction, $<\texttt{initial\_marking} \Rightarrow M_0(p)> \in i_p$.
By property of the $\mathcal{H}$-VHDL initialization relation, and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, then $\sigma_0(id_p)(\texttt{initial\_marking}) = M_0(p)$. Rewriting $\sigma_0(id_p)(\texttt{initial\_marking})$ as $M_0(p)$ in the current goal: $\boxed{M_0(p) = s_0.M(p).}$
By definition of s_0, we can rewrite $s_0.M(p)$ as $M_0(p)$ in the current goal, tautology.

□

Lemma 3 (Null input token sum at initial state)**.** *For all* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e, \sigma_0 \in \Sigma$ *that verify the hypotheses of Definition 10, then* $\forall p \in P, id_p \in Comps(\Delta)$ *s.t.* $\gamma(p) = id_p$, $\sigma_0(id_p)(\texttt{s\_input\_token\_sum}) = 0$.

Proof.

Given a p and an id_p s.t. $\gamma(p) = id_p$, let us show that $\boxed{\sigma_0(id_p)(\texttt{s\_input\_token\_sum}) = 0.}$
By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$.
By property of the initialization relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `input_tokens_sum` process defined in the place design architecture, we can deduce:

$$\sigma_0(id_p)(\texttt{sits}) = \sum_{i=0}^{\Delta(id_p)(\texttt{ian})-1} \begin{cases} \sigma_0(id_p)(\texttt{iaw})[i] \texttt{ if } \sigma_0(id_p)(\texttt{itf})[i] \\ 0 \ otherwise \end{cases} \tag{2}$$

Rewriting the goal with Equation (2):

$$\boxed{\sum_{i=0}^{\Delta(id_p)(\texttt{ian})-1} \begin{cases} \sigma_0(id_p)(\texttt{iaw})[i] \texttt{ if } \sigma_0(id_p)(\texttt{itf})[i] \\ 0 \ otherwise \end{cases} = 0.}$$

Let us perform case analysis on $input(p)$; there are two cases:

1. $input(p) = \emptyset$:

   By construction, we have $<\texttt{input\_arcs\_number} \Rightarrow 1> \in g_p$,
   $<\texttt{input\_transitions\_fired(0)} \Rightarrow \texttt{true}> \in i_p$,
   and $<\texttt{input\_arcs\_weights(0)} \Rightarrow 0> \in i_p$.

   By property of the elaboration relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and $<\texttt{input\_arcs\_number} \Rightarrow 1> \in g_p$, we can deduce $\Delta(id_p)(\texttt{ian}) = 1$.

   By property of the initialization relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, $<\texttt{input\_transitions\_fired(0)} \Rightarrow \texttt{true}> \in i_p$ and $<\texttt{input\_arcs\_weights(0)} \Rightarrow 0> \in i_p$, we can deduce $\sigma_0(id_p)(\texttt{itf})[0] = \texttt{true}$ and $\sigma_0(id_p)(\texttt{iaw})[0] = 0$.

   Rewriting the goal with $\Delta(id_p)(\texttt{ian}) = 1$, $\sigma_0(id_p)(\texttt{itf})[0] = \texttt{true}$, $\sigma_0(id_p)(\texttt{iaw})[0] = 0$ and simplifying the goal, tautology.

2. $input(p) \neq \emptyset$:

   By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(p)|> \in g_p$, and by property of the elaboration relation, and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\Delta(id_p)(\texttt{ian}) = |input(p)|$.

   Let us reason by induction on the sum term of the goal.

   - **BASE CASE**: The sum term equals 0, then tautology.
   - **INDUCTION CASE**:

     $$\sum_{i=1}^{\Delta(id_p)(\texttt{ian})-1} \begin{cases} \sigma_0(id_p)(\texttt{iaw})[i] \texttt{ if } \sigma_0(id_p)(\texttt{itf})[i] \\ 0 \ otherwise \end{cases} = 0$$

     $$\boxed{\begin{cases} \sigma_0(id_p)(\texttt{iaw})[0] \texttt{ if } \sigma_0(id_p)(\texttt{itf})[0] \\ 0 \ otherwise \end{cases} + \sum_{i=1}^{\Delta(id_p)(\texttt{ian})-1} \begin{cases} \sigma_0(id_p)(\texttt{iaw})[i] \texttt{ if } \sigma_0(id_p)(\texttt{itf})[i] \\ 0 \ otherwise \end{cases} = 0}$$

Using the induction hypothesis to rewrite the goal:

$$\boxed{\begin{cases}\sigma_0(id_p)(\texttt{iaw})[0] \texttt{ if } \sigma_0(id_p)(\texttt{itf})[0] \\ 0 \textit{ otherwise}\end{cases} = 0}$$

Since $input(p) \neq \emptyset$, by construction, there exist an $id_t \in Comps(\Delta), g_t, i_t, o_t$ s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, $id_{ft} \in Sigs(\Delta)$ s.t. $<\texttt{fired} \Rightarrow id_{ft}> \in o_t$ and $<\texttt{input\_transitions\_fired(0)} \Rightarrow \texttt{id}_{\texttt{ft}}> \in i_p$.

By property of the initialization relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, $<\texttt{fired} \Rightarrow id_{ft}> \in o_t$ and $<\texttt{input\_transitions\_fired(0)} \Rightarrow \texttt{id}_{\texttt{ft}}> \in i_p$, we can deduce $\sigma_0(id_p)(\texttt{itf})[0] = \sigma_0(id_t)(\texttt{fired})$.

Rewriting the goal with $\sigma_0(id_p)(\texttt{itf})[0] = \sigma_0(id_t)(\texttt{fired})$:

$$\boxed{\begin{cases}\sigma_0(id_p)(\texttt{iaw})[0] \texttt{ if } \sigma_0(id_t)(\texttt{fired}) \\ 0 \textit{ otherwise}\end{cases} = 0}$$

Appealing to Lemma 10, we can deduce $\sigma_0(id_t)(\texttt{fired}) = \texttt{false}$.

Rewriting the goal with $\sigma_0(id_t)(\texttt{fired}) = \texttt{false}$, and simplifying the goal, tautology.

□

Lemma 4 (Null output token sum at initial state)**.** *For all $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e, \sigma_0 \in \Sigma$ that verify the hypotheses of Definition 10, then $\forall p \in P, id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$, $\sigma_0(id_p)(\texttt{s\_output\_token\_sum}) = 0$.*

Proof.

The proof is similar to the proof of Lemma 3. □

4.2 Initial states and time counters

Lemma 5 (Initial states equal time counters)**.** *For all $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e, \sigma_0 \in \Sigma$ that verify the hypotheses of Definition 10, then $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,*

$u(I_s(t)) = \infty \wedge s_0.I(t) \leq l(I_s(t)) \Rightarrow s_0.I(t) = \sigma_0(id_t)(\texttt{s\_time\_counter}) \wedge$
$u(I_s(t)) = \infty \wedge s_0.I(t) > l(I_s(t)) \Rightarrow \sigma_0(id_t)(\texttt{s\_time\_counter}) = l(I_s(t)) \wedge$
$u(I_s(t)) \neq \infty \wedge s_0.I(t) > u(I_s(t)) \Rightarrow \sigma_0(id_t)(\texttt{s\_time\_counter}) = u(I_s(t)) \wedge$
$u(I_s(t)) \neq \infty \wedge s_0.I(t) \leq u(I_s(t)) \Rightarrow s_0.I(t) = \sigma_0(id_t)(\texttt{s\_time\_counter})$.

Proof.

Given a $t \in T_i$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show that:

1. $\boxed{u(I_s(t)) = \infty \wedge s_0.I(t) \leq l(I_s(t)) \Rightarrow s_0.I(t) = \sigma_0(id_t)(\texttt{s\_time\_counter})}$

2. $\boxed{u(I_s(t)) = \infty \wedge s_0.I(t) > l(I_s(t)) \Rightarrow \sigma_0(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))}$

3. $\boxed{u(I_s(t)) \neq \infty \wedge s_0.I(t) > u(I_s(t)) \Rightarrow \sigma_0(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))}$

4. $\boxed{u(I_s(t)) \neq \infty \wedge s_0.I(t) \leq u(I_s(t)) \Rightarrow s_0.I(t) = \sigma_0(id_t)(\texttt{s\_time\_counter})}$

By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$. Then, let us show the 4 previous points.

1. Assuming that $u(I_s(t)) = \infty \wedge s_0.I(t) \leq l(I_s(t))$, then let us show $\boxed{s_0.I(t) = \sigma_0(id_t)(\texttt{s\_time\_counter}).}$

   Rewriting $s_0.I(t)$ as 0, by definition of s_0, $\boxed{\sigma_0(id_t)(\texttt{s\_time\_counter}) = 0.}$

   By property of the $\mathcal{H}$-VHDL initialization relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `time_counter` process defined in the transition design architecture, we can deduce $\sigma_0(id_t)(\texttt{s\_time\_counter}) = 0.$

2. Assuming that $u(I_s(t)) = \infty$ and $s_0.I(t) > l(I_s(t))$, let us show $\boxed{\sigma_0(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))}$.

   By definition, $l(I_s(t)) \in \mathbb{N}^*$ and $s_0.I(t) = 0$. Then, $l(I_s(t)) < 0$ is a contradiction.

3. Assuming that $u(I_s(t)) \neq \infty$ and $s_0.I(t) > u(I_s(t))$, let us show $\boxed{\sigma_0(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))}$.

   By definition, $u(I_s(t)) \in \mathbb{N}^*$ and $s_0.I(t) = 0$. Then, $u(I_s(t)) < 0$ is a contradiction.

4. Assuming that $u(I_s(t)) \neq \infty$ and $s_0.I(t) \leq u(I_s(t))$, let us show $\boxed{s_0.I(t) = \sigma_0(id_t)(\texttt{s\_time\_counter})}$.

   Rewriting $s_0.I(t)$ as 0, by definition of s_0, $\boxed{\sigma_0(id_t)(\texttt{s\_time\_counter}) = 0.}$

   By property of the $\mathcal{H}$-VHDL initialization relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `time_counter` process defined in the transition design architecture, we can deduce $\sigma_0(id_t)(\texttt{s\_time\_counter}) = 0.$

□

4.3 Initial states and reset orders

Lemma 6 (Initial states equal reset orders)**.** *For all* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e, \sigma_0 \in \Sigma$ *that verify the hypotheses of Definition 10, then* $\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $s_0.reset_t(t) = \sigma_0(id_t)(\texttt{s\_reinit\_time\_counter})$.

Proof.

Given a $t \in T_i$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show that
$\boxed{s_0.reset_t(t) = \sigma_0(id_t)(\texttt{s\_reinit\_time\_counter}).}$

Rewriting $s_0.reset_t(t)$ as `false`, by definition of s_0, $\boxed{\sigma_0(id_t)(\texttt{s\_reinit\_time\_counter}) = \texttt{false}.}$

By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$.

By property of the $\mathcal{H}$-VHDL initialization relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `reinit_time_counter_evaluation` process defined in the `transition` design architecture

we can deduce $\sigma_0(id_t)(\texttt{s\_reinit\_time\_counter}) = \prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma_0(id_t)(\texttt{rt})[i]$.

Rewriting $\sigma_0(id_t)(\texttt{s\_reinit\_time\_counter})$ as $\prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma_0(id_t)(\texttt{rt})[i]$,

$\boxed{\prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma_0(id_t)(\texttt{rt})[i] = \texttt{false}.}$

For all $t \in T$ (resp. $p \in P$), let $input(t)$ (resp. $input(p)$) be the set of input places of t (resp. input transitions of p), and let $output(t)$ (resp. $output(p)$) be the set of output places of t (resp. output transitions of p).

Let us perform case analysis on $input(t)$; there are 2 cases:

- **CASE** $input(t) = \emptyset$.

  By construction, $<\texttt{input\_arcs\_number} \Rightarrow 1> \in g_t$, and by property of the elaboration relation, and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\Delta(id_t)(\texttt{ian}) = 1$.

  By construction, $< \texttt{reinit\_time(0)} \Rightarrow \texttt{false} > \in i_t$, and by property of the initialization relation and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\sigma_0(id_t)(\texttt{rt})[0] = \texttt{false}$.

  Rewriting $\Delta(id_t)(\texttt{ian})$ as 1 and $\sigma_0(id_t)(\texttt{rt})[0]$ as `false`, tautology.

- **CASE** $input(t) \neq \emptyset$.

  To prove the current goal, we can equivalently prove that
  $\boxed{\exists i \in [0, \Delta(id_t)(\texttt{ian}) - 1] \; s.t. \; \sigma_0(id_t)(\texttt{rt})[i] = \texttt{false}.}$

  Since $input(t) \neq \emptyset$, $\exists p \; s.t. \; p \in input(t)$. Let us take such a $p \in input(t)$.

  By construction, for all $p \in P$, there exist $id_p \; s.t. \; \gamma(p) = id_p$.

  By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$.

  By construction, there exist $i \in [0, |input(t)| - 1]$, $j \in [0, |output(p)| - 1]$, $id_{ji} \in Sigs(\Delta)$ s.t. $<\texttt{reinit\_transitions\_time(j)} \Rightarrow id_{ji}> \in o_p$ and $<\texttt{reinit\_time(i)} \Rightarrow id_{ji}> \in i_t$. Let us take such a i, j and id_{ji}.

  By construction and $input(t) \neq \emptyset$, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in g_t$.

  By property of the $\mathcal{H}$-VHDL elaboration relation and $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in g_t$, we can deduce $\Delta(id_t)(\texttt{ian}) = |input(t)|$.

  Since $\Delta(id_t)(\texttt{ian}) = |input(t)|$ and we have an $i \in [0, |input(t)| - 1]$, then, we have an $i \in [0, \Delta(id_t)(\texttt{ian}) - 1]$. Let us take that i to prove the goal.

Then, we must show $\boxed{\sigma_0(id_t)(\texttt{rt})[i] = \texttt{false}.}$

By property of the $\mathcal{H}$-VHDL initialization relation and $<\texttt{reinit\_time(i)} \Rightarrow id_{ji}> \in i_t$, we can deduce $\sigma_0(id_t)(\texttt{rt})[i] = \sigma_0(id_{ji})$.

Rewriting $\sigma_0(id_t)(\texttt{rt})[i]$ as $\sigma_0(id_{ji})$, $\boxed{\sigma_0(id_{ji}) = \texttt{false}.}$

By property of the $\mathcal{H}$-VHDL initialization relation and $<\texttt{reinit\_transitions\_time(j)} \Rightarrow id_{ji}> \in o_p$, we can deduce $\sigma_0(id_{ji}) = \sigma_0(id_p)(\texttt{rtt})[j]$.

Rewriting $\sigma_0(id_{ji})$ as $\sigma_0(id_p)(\texttt{rtt})[j]$, $\boxed{\sigma_0(id_p)(\texttt{rtt})[j] = \texttt{false}.}$

Since $t \in output(p)$, then we know that $output(p) \neq \emptyset$.

Then, by construction, $<\texttt{output\_arcs\_number} \Rightarrow |output(p)|> \in g_p$.

By property of the elaboration relation and $<\texttt{output\_arcs\_number} \Rightarrow |output(p)|> \in g_p$, we can deduce that $\Delta(id_p)(\texttt{oan}) = |output(p)|$.

Since $\Delta(id_p)(\texttt{oan}) = |output(p)|$ and $j \in [0, |output(p)| - 1]$, then $j \in [0, \Delta(id_p)(\texttt{oan}) - 1]$.

By property of the $\mathcal{H}$-VHDL initialization relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, through the examination of the `reinit_transitions_time_evaluation` process defined in the `place` design architecture, and since $j \in [0, \Delta(id_p)(\texttt{oan}) - 1]$, $\sigma_0(id_p)(\texttt{rtt})[j] = \texttt{false}.$

□

4.4 Initial states and condition values

Lemma 7 (Initial states equal condition values)**.** *For all $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e, \sigma_0 \in \Sigma$ that verify the hypotheses of Definition 10, then $\forall c \in \mathcal{C}, id_c \in Ins(\Delta)$ s.t. $\gamma(c) = id_c$, $s_0.cond(c) = \sigma_0(id_c)$.*

Proof.

Given a $c \in \mathcal{C}$ and an $id_c \in Ins(\Delta)$ *s.t.* $\gamma(c) = id_c$, let us show that $\boxed{s_0.cond(c) = \sigma_0(id_c).}$

Rewriting $s_0.cond(c)$ as $\texttt{false}$, by definition of s_0, $\boxed{\sigma_0(id_c) = \texttt{false}.}$
By construction, id_c is an input port identifier of Boolean type in the $\mathcal{H}$-VHDL design d, and thus, by property of the $\mathcal{H}$-VHDL elaboration relation, we can deduce $\sigma_e(id_c) = \texttt{false}$.
By property of the $\mathcal{H}$-VHDL initialization relation and $id_c \in Ins(\Delta)$, we can deduce $\sigma_e(id_c) = \sigma_0(id_c)$.
Rewriting $\sigma_0(id_c)$ as $\sigma_e(id_c)$ and $\sigma_e(id_c)$ as $\texttt{false}$, tautology.

□

4.5 Initial states and action executions

Lemma 8 (Initial states equal action executions)**.** *For all* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e, \sigma_0 \in \Sigma$ *that verify the hypotheses of Definition 10, then* $\forall a \in \mathcal{A}, id_a \in Outs(\Delta)$ *s.t.* $\gamma(a) = id_a,\ s_0.ex(a) = \sigma_0(id_a)$.

Proof.

Given a $a \in \mathcal{A}$ and an $id_a \in Outs(\Delta)$ *s.t.* $\gamma(a) = id_a$, let us show that $\boxed{s_0.ex(a) = \sigma_0(id_a).}$

Rewriting $s_0.ex(a)$ as `false`, by definition of s_0, $\boxed{\sigma_0(id_a) = \texttt{false}.}$
By construction, id_a is an output port identifier of Boolean type in the $\mathcal{H}$-VHDL design d. Moreover, we know that the output port identifier id_a is assigned to `false` in the generated `action` process during the initialization phase (i.e. the assignment is a part of a *reset* block). Thus, we can deduce that $\sigma_0(id_a) = \texttt{false}$.
Rewriting $\sigma_0(id_a)$ as `false`, tautology.

□

4.6 Initial states and function executions

Lemma 9 (Initial states equal function executions)**.** *For all* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e, \sigma_0 \in \Sigma$ *that verify the hypotheses of Definition 10, then* $\forall f \in \mathcal{F}, id_f \in Outs(\Delta)$ *s.t.* $\gamma(f) = id_f,\ s_0.ex(f) = \sigma_0(id_f)$.

Proof.

Given a $f \in \mathcal{F}$ and an $id_f \in Outs(\Delta)$ *s.t.* $\gamma(f) = id_f$, let us show that $\boxed{s_0.ex(f) = \sigma_0(id_f).}$

Rewriting $s_0.ex(f)$ as `false`, by definition of s_0, $\boxed{\sigma_0(id_f) = \texttt{false}.}$
By construction, id_f is an output port identifier of Boolean type in the $\mathcal{H}$-VHDL design d, and thus, by property of the $\mathcal{H}$-VHDL elaboration relation, we can deduce $\sigma_e(id_f) = \texttt{false}$.
By construction, and by property of the initialization relation, we know that the output port identifier id_f is assigned to `false` in the generated `function` process during the initialization phase (i.e. the assignment is a part of a *reset* block). Thus, we can deduce $\sigma_0(id_f) = \texttt{false}$.
Rewriting $\sigma_0(id_f)$ as `false`, tautology.

□

4.7 Initial states and fired transitions

Lemma 10 (No fired at initial state)**.** $\forall d \in design, \Delta \in ElDesign, \sigma_e, \sigma_0 \in \Sigma, id_t \in Comps(\Delta)$, g_t, i_t, o_t *s.t. :*

- $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d.cs \xrightarrow{elab} \sigma_0$
- $\Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$
- $\texttt{comp}(id_t, \textit{transition}, g_t, i_t, o_t) \in d.cs$

then $\sigma_0(id_t)(\textit{fired}) = \texttt{false}$.

Proof.

Assuming all the above hypotheses, let us show $\boxed{\sigma_0(id_t)(\texttt{fired}) = \texttt{false}.}$
By property of the initialization relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `fired_evaluation` process defined in the transition design architecture, we can deduce:

$$\sigma_0(id_t)(\texttt{fired}) = \sigma_0(id_t)(\texttt{s\_firable}) \;.\; \sigma_0(id_t)(\texttt{s\_priority\_combination}) \tag{3}$$

Rewriting the goal with Equation (3): $\boxed{\sigma_0(id_t)(\texttt{sfa}) \;.\; \sigma_0(id_t)(\texttt{spc}) = \texttt{false}.}$
By property of the initialization relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `firable` process defined in the `transition` design architecture, we can deduce $\sigma_0(id_t)(\texttt{sfa}) = \texttt{false}$.
Rewriting the goal with $\sigma_0(id_t)(\texttt{sfa}) = \texttt{false}$ and simplifying the goal, tautology. □

5 First rising edge lock-step simulation

Lemma 11 (First rising edge lock-step simulation)**.** *For all well-defined* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e \in \Sigma$, $E_p \in \mathbb{N} \to (id \nrightarrow v)$, *and* $E_c \in \mathbb{N} \to (\mathcal{C} \to \mathbb{B})$ *that verify the hypotheses of Definition 9, and for all clock count* $\tau \in \mathbb{N}$, $\sigma_0, \sigma_i, \sigma_\uparrow, \sigma'_0 \in \Sigma$ *such that:*

- σ_0 *is the initial state of design* d*:*
  $\mathcal{D}, \Delta, \sigma_e \vdash d.beh \xrightarrow{cs_i} \sigma_i$ *and* $\mathcal{D}, \Delta, \sigma_i \vdash d.beh \xrightsquigarrow{} \sigma_0$
- *a rising edge step leads from* σ_0 *to* σ'_0*:*
  $\mathcal{D}_{\mathcal{H}}, \Delta, \texttt{inj}(\sigma_0, E_p, \tau) \vdash d.beh \xrightarrow{cs_\uparrow} \sigma_\uparrow$ *and* $\mathcal{D}_{\mathcal{H}}, \Delta, \sigma_\uparrow \vdash d.beh \xrightsquigarrow{} \sigma'_0$

then $\gamma \vdash s_0 \overset{\uparrow}{\approx} \sigma'_0$.

Proof.

By definition of the Full post rising edge state similarity relation, there are 8 points to prove.

1. $\forall p \in P, id_p \in Comps(\Delta)\ s.t.\ \gamma(p) = id_p,\ s_0.M(p) = \sigma(id_p)(\texttt{s\_marking})$.
2. $\forall t \in T_i, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
   $\big(u(I_s(t)) = \infty \wedge s_0.I(t) \leq l(I_s(t)) \Rightarrow s_0.I(t) = \sigma(id_t)(\texttt{s\_time\_counter})\big)$
   $\wedge\big(u(I_s(t)) = \infty \wedge s_0.I(t) > l(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)$
   $\wedge\big(u(I_s(t)) \neq \infty \wedge s_0.I(t) > u(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)$
   $\wedge\big(u(I_s(t)) \neq \infty \wedge s_0.I(t) \leq u(I_s(t)) \Rightarrow s_0.I(t) = \sigma(id_t)(\texttt{s\_time\_counter})\big)$.
3. $\forall t \in T_i, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,\ s_0.reset_t(t) = \sigma(id_t)(\texttt{s\_reinit\_time\_counter})$.
4. $\forall a \in \mathcal{A}, id_a \in Outs(\Delta)\ s.t.\ \gamma(a) = id_a,\ s_0.ex(a) = \sigma(id_a)$.
5. $\forall f \in \mathcal{F}, id_f \in Outs(\Delta)\ s.t.\ \gamma(f) = id_f,\ s_0.ex(f) = \sigma(id_f)$.
6. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,\ t \in Sens(s_0.M) \Leftrightarrow \sigma(id_t)(\texttt{s\_enabled}) = \texttt{true}$.
7. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,\ t \notin Sens(s_0.M) \Leftrightarrow \sigma(id_t)(\texttt{s\_enabled}) = \texttt{false}$.
8. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
   $$\sigma(id_t)(\texttt{s\_condition\_combination}) = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}$$
   where $conds(t) = \{c \in \mathcal{C} \mid \mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1\}$.
9. $\forall c \in \mathcal{C}, id_c \in Ins(\Delta)$ s.t. $\gamma(c) = id_c,\ \sigma(id_c) = E_c(\tau, c)$.

- Apply the First rising edge equal marking lemma to solve 1.
- Apply the First rising edge equal time counters lemma to solve 2.
- Apply the First rising edge equal reset orders lemma to solve 3.
- Apply the First rising edge equal action executions lemma to solve 4.
- Apply the First rising edge equal function executions lemma to solve 5.
- Apply the First rising edge equal sensitized lemma to solve 6.
- Apply the First rising edge not equal sensitized lemma to solve 7.
- Apply the First rising edge equal condition combination lemma to solve 8.
- Apply the First rising edge equal conditions lemma to solve 9.

□

Definition 11 (First rising edge hypotheses)**.** *Given a* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign$, $\sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma \in \Sigma$, $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$, $E_p \in \mathbb{N} \times \{\uparrow, \downarrow\}) \to Ins(\Delta) \to value$, $\tau \in \mathbb{N}$, *assume that:*

- $\lfloor sitpn \rfloor_b = (d, \gamma)$ *and* $\mathcal{D}_\mathcal{H}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$ *and* $\gamma \vdash E_p \overset{env}{=} E_c$
- σ_0 *is the initial state of* Δ: $\Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$
- $E_c, \tau \vdash s_0 \xrightarrow{\uparrow_0} s_0$
- $\texttt{Inject}(\sigma_0, E_p, \tau, \sigma_i)$ *and* $\Delta, \sigma_i \vdash \text{d.cs} \xrightarrow{\uparrow} \sigma_\uparrow$ *and* $\Delta, \sigma_\uparrow \vdash \text{d.cs} \xrightarrow{\theta} \sigma$

5.1 First rising edge and marking

Lemma 12 (First rising edge equal marking)**.** *For all* $sitpn, b, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma, E_c, E_p, \tau$ *that verify the hypotheses of Definition 11, then* $\forall p \in P, id_p \in Comps(\Delta)$ *s.t.* $\gamma(p) = id_p$, $s_0.M(p) = \sigma(id_p)(\texttt{s\_marking})$.

Proof.

Given a p and an id_p s.t. $\gamma(p) = id_p$, let us show that $\boxed{s_0.M(p) = \sigma(id_p)(\texttt{s\_marking}).}$
By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$.
By property of the `Inject` relation, the $\mathcal{H}$-VHDL rising edge relation, the stabilize relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `marking` process defined in the place design architecture, we can deduce:

$$\sigma(id_p)(\texttt{sm}) = \sigma_0(id_p)(\texttt{sm}) + \sigma_0(id_p)(\texttt{sits}) - \sigma_0(id_p)(\texttt{sots}) \tag{4}$$

Rewriting the goal with Equation (4):
$\boxed{s_0.M(p) = \sigma_0(id_p)(\texttt{sm}) + \sigma_0(id_p)(\texttt{sits}) - \sigma_0(id_p)(\texttt{sots}).}$
Appealing to Lemmas 3 and 4, we can deduce $\sigma_0(id_p)(\texttt{sits}) = 0$ and $\sigma_0(id_p)(\texttt{sots}) = 0$. Rewriting the goal with $\sigma_0(id_p)(\texttt{sits}) = 0$ and $\sigma_0(id_p)(\texttt{sots}) = 0$, $\boxed{s_0.M(p) = \sigma_0(id_p)(\texttt{sm}).}$

Appealing to Lemma 2, $s_0.M(p) = \sigma_0(id_p)(\texttt{sm})$. □

5.2 First rising edge and time counters

Lemma 13 (First rising edge equal time counters)**.** *For all* $sitpn, b, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma, E_c, E_p, \tau$ *that verify the hypotheses of Definition 11, then* $\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,

$u(I_s(t)) = \infty \wedge s_0.I(t) \leq l(I_s(t)) \Rightarrow s_0.I(t) = \sigma(id_t)(\texttt{s\_time\_counter}) \wedge$
$u(I_s(t)) = \infty \wedge s_0.I(t) > l(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = l(I_s(t)) \wedge$
$u(I_s(t)) \neq \infty \wedge s_0.I(t) > u(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = u(I_s(t)) \wedge$
$u(I_s(t)) \neq \infty \wedge s_0.I(t) \leq u(I_s(t)) \Rightarrow s_0.I(t) = \sigma(id_t)(\texttt{s\_time\_counter}).$

Proof.

Given a $t \in T_i$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show that:

1. $\boxed{u(I_s(t)) = \infty \wedge s_0.I(t) \leq l(I_s(t)) \Rightarrow s_0.I(t) = \sigma(id_t)(\texttt{s\_time\_counter})}$
2. $\boxed{u(I_s(t)) = \infty \wedge s_0.I(t) > l(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))}$
3. $\boxed{u(I_s(t)) \neq \infty \wedge s_0.I(t) > u(I_s(t)) \Rightarrow \sigma(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))}$
4. $\boxed{u(I_s(t)) \neq \infty \wedge s_0.I(t) \leq u(I_s(t)) \Rightarrow s_0.I(t) = \sigma(id_t)(\texttt{s\_time\_counter})}$

By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$. Then, let us show the 4 previous points:

1. Assuming that $u(I_s(t)) = \infty$ and $s_0.I(t) \leq l(I_s(t))$, let us show $\boxed{s_0.I(t) = \sigma(id_t)(\texttt{stc}).}$

   By property of the `Inject` relation, the $\mathcal{H}$-VHDL rising edge and stabilize relations, and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\sigma(id_t)(\texttt{stc}) = \sigma_0(id_t)(\texttt{stc})$.

   Rewriting $\sigma(id_t)(\texttt{stc})$ as $\sigma_0(id_t)(\texttt{stc})$, $\boxed{s_0.I(t) = \sigma_0(id_t)(\texttt{stc}).}$

   Appealing to Lemma 5, $s_0.I(t) = \sigma_0(id_t)(\texttt{stc}).$

2. Assuming that $u(I_s(t)) = \infty$ and $s_0.I(t) > l(I_s(t))$, let us show $\boxed{\sigma(id_t)(\texttt{stc}) = l(I_s(t)).}$

   By definition, $l(I_s(t)) \in \mathbb{N}^*$ and $s_0.I(t) = 0$. Then, $l(I_s(t)) < 0$ is a contradiction.

3. Assuming that $u(I_s(t)) \neq \infty$ and $s_0.I(t) > u(I_s(t))$, let us show $\boxed{\sigma(id_t)(\texttt{stc}) = u(I_s(t))}$.

   By definition, $u(I_s(t)) \in \mathbb{N}^*$ and $s_0.I(t) = 0$. Then, $u(I_s(t)) < 0$ is a contradiction.

4. Assuming that $u(I_s(t)) \neq \infty$ and $s_0.I(t) \leq u(I_s(t))$, let us show $\boxed{s_0.I(t) = \sigma(id_t)(\texttt{stc}).}$

   By property of the `Inject` relation, the $\mathcal{H}$-VHDL rising edge and stabilize relations, and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\sigma(id_t)(\texttt{stc}) = \sigma_0(id_t)(\texttt{stc})$.

   Rewriting $\sigma(id_t)(\texttt{stc})$ as $\sigma_0(id_t)(\texttt{stc})$, $\boxed{s_0.I(t) = \sigma_0(id_t)(\texttt{stc}).}$

   Appealing to Lemma 5, $s_0.I(t) = \sigma_0(id_t)(\texttt{stc}).$

□

5.3 First rising edge and reset orders

Lemma 14 (First rising edge equal reset orders)**.** *For all* $sitpn, b, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma, E_c, E_p, \tau$ *that verify the hypotheses of Definition 11, then*
$\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $s_0.reset_t(t) = \sigma(id_t)(\texttt{s\_reinit\_time\_counter})$.

Proof.

Given a $t \in T$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show that
$\boxed{s_0.reset_t(t) = \sigma(id_t)(\texttt{srtc}).}$
By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$. By property of the stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `reinit_time_counter_evaluation` process defined in the transition design architecture, we can deduce:

$$\sigma(id_t)(\texttt{srtc}) = \sum_{i=0}^{\Delta(id_t)(\texttt{input\_arcs\_number})-1} \sigma(id_t)(\texttt{reinit\_time})[i] \tag{5}$$

Rewriting the goal with Equation (5): $\boxed{s_0.reset_t(t) = \sum_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma(id_t)(\texttt{rt})[i].}$

Let us perform case analysis on $input(t)$; there are two cases:

- **CASE** $input(t) = \emptyset$:

  By construction, $<\texttt{input\_arcs\_number} \Rightarrow 1> \in g_t$, and by property of the $\mathcal{H}$-VHDL elaboration relation, we can deduce $\Delta(id_t)(\texttt{ian}) = 1$.

  By construction, $<\texttt{reinit\_time}(0) \Rightarrow \texttt{false}> \in i_t$, and by property of the $\mathcal{H}$-VHDL stabilize relation, $\sigma(id_t)(\texttt{rt})[0] = \texttt{false}$.

  Rewriting the goal with $\Delta(id_t)(\texttt{ian}) = 1$ and $\sigma(id_t)(\texttt{rt})[0] = \texttt{false}$, $\boxed{s_0.reset_t(t) = \texttt{false}.}$

  By definition of s_0, $s_0.reset_t(t) = \texttt{false}$.

- **CASE** $input(t) \neq \emptyset$:

  By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in g_t$, and by property of the $\mathcal{H}$-VHDL elaboration relation, we can deduce $\Delta(id_t)(\texttt{ian}) = |input(t)|$.

  Rewriting $\Delta(id_t)(\texttt{ian})$ as $|input(t)|$, $\boxed{s_0.reset_t(t) = \sum_{i=0}^{|input(t)|-1} \sigma(id_t)(\texttt{rt})[i].}$

  By definition of s_0, $s_0.reset_t(t) = \texttt{false}$. Rewriting $s_0.reset_t(t)$ as $\texttt{false}$,
  $\boxed{\sum_{i=0}^{|input(t)|-1} \sigma(id_t)(\texttt{rt})[i] = \texttt{false}.}$

  Given a $i \in [0, |input(t)| - 1]$, let us show $\boxed{\sigma(id_t)(\texttt{rt})[i] = \texttt{false}.}$

By construction, and since $input(t) \neq \emptyset$, there exist a $p \in input(t)$, an $id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$, a g_p, an i_p, an o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and there exist a $j \in [0, |output(p)| - 1]$ and an $id_{ji} \in Sigs(\Delta)$ s.t. $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in o_p$ and $<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in i_t$.

By property of the stabilize relation, $<\texttt{reinit\_transition\_time(j)} \Rightarrow id_{ji}> \in o_p$ and $<\texttt{reinit\_time(i)} \Rightarrow id_{ji}> \in i_t$, we can deduce $\sigma(id_t)(\texttt{rt})[i] = \sigma(id_{ji}) = \sigma(id_p)(\texttt{rtt})[j]$.

Rewriting $\sigma(id_t)(\texttt{rt})[i]$ as $\sigma(id_{ji})$ and $\sigma(id_{ji})$ as $\sigma(id_p)(\texttt{rtt})[j]$, $\boxed{\sigma(id_p)(\texttt{rtt})[j] = \texttt{false}.}$

By property of the $\mathcal{H}$-VHDL rising edge and stabilize relations, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the process defined in the place design architecture, we can deduce:

$$\begin{aligned}\sigma(id_p)(\texttt{rtt})[j] =&((\sigma_0(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma_0(id_p)(\texttt{oat})[j] = \texttt{test})\\ &.(\sigma_0(id_p)(\texttt{sm}) - \sigma_0(id_p)(\texttt{sots}) < \sigma_0(id_p)(\texttt{oaw})[j])\\ &.(\sigma_0(id_p)(\texttt{sots}) > 0))\\ &+ (\sigma_0(id_p)(\texttt{otf})[j])\end{aligned} \tag{6}$$

Rewriting the goal with Equation (6),

$$\boxed{\begin{aligned}\texttt{false} =&((\sigma_0(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma_0(id_p)(\texttt{oat})[j] = \texttt{test})\\ &.(\sigma_0(id_p)(\texttt{sm}) - \sigma_0(id_p)(\texttt{sots}) < \sigma_0(id_p)(\texttt{oaw})[j])\\ &.(\sigma_0(id_p)(\texttt{sots}) > 0))\\ &+ (\sigma_0(id_p)(\texttt{otf})[j])\end{aligned}}$$

By construction, there exists an $id_{fj} \in Sigs(\Delta)$ s.t. $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{fj}}> \in o_t$ and $<\texttt{output\_transitions\_fired(j)} \Rightarrow \texttt{id}_{\texttt{fj}}> \in i_p$.

By property of the initialization relation, $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{fj}}> \in o_t$ and $<\texttt{output\_transitions\_fired(j)} \Rightarrow \texttt{id}_{\texttt{fj}}> \in i_p$, we can deduce $\sigma_0(id_p)(\texttt{otf})[j] = \sigma_0(id_{fj}) = \sigma_0(id_t)(\texttt{fired})$.

Appealing to Lemma 10, we can deduce $\sigma_0(id_t)(\texttt{fired}) = \texttt{false}$ and consequently $\sigma_0(id_p)(\texttt{otf})[j] = \texttt{false}$.

Rewriting $\sigma_0(id_p)(\texttt{otf})[j]$ as $\texttt{false}$ and simplifying the goal,

$$\boxed{\begin{aligned}false =&((\sigma_0(id_p)(\texttt{oat})[j] = \texttt{BASIC} + \sigma_0(id_p)(\texttt{oat})[j] = \texttt{TEST})\\ &.(\sigma_0(id_p)(\texttt{sm}) - \sigma_0(id_p)(\texttt{sots}) < \sigma_0(id_p)(\texttt{oaw})[j])\\ &.(\sigma_0(id_p)(\texttt{sots}) > 0))\end{aligned}}$$

Appealing to Lemma 4, we can deduce $\sigma_0(id_p)(\texttt{sots}) = 0$.

Rewriting $\sigma_0(id_p)(\texttt{sots})$ as 0 and simplifying the goal, tautology.

□

5.4 First rising edge and action executions

Lemma 15 (First rising edge equal action executions)**.** *For all* $sitpn, b, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma, E_c, E_p, \tau$ *that verify the hypotheses of Definition 11, then*
$\forall a \in \mathcal{A}, id_a \in Outs(\Delta)$ *s.t.* $\gamma(a) = id_a,\ s_0.ex(a) = \sigma(id_a)$.

Proof.

Given an $a \in \mathcal{A}$ and an $id_a \in Outs(\Delta)$ *s.t.* $\gamma(a) = id_a$, let us show that $\boxed{s_0.ex(a) = \sigma(id_a).}$
By construction, id_a is an output port identifier of Boolean type in the $\mathcal{H}$-VHDL design d. The generated `action` process assigns a value to the output port id_a only during the initialization phase or a falling edge phase.
By property of the `Inject`, $\mathcal{H}$-VHDL rising edge and stabilize relations, we can deduce $\sigma(id_a) = \sigma_0(id_a)$.
Rewriting $\sigma(id_a)$ as $\sigma_0(id_a)$, $\boxed{s_0.ex(a) = \sigma_0(id_a).}$ Appealing to Lemma 8, $s_0.ex(a) = \sigma_0(id_a).$

□

5.5 First rising edge and function executions

Lemma 16 (First rising edge equal function executions)**.** *For all* $sitpn, b, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma, E_c, E_p, \tau$ *that verify the hypotheses of Definition 11, then*
$\forall f \in \mathcal{F}, id_f \in Outs(\Delta)$ *s.t.* $\gamma(f) = id_f,\ s_0.ex(f) = \sigma(id_f)$.

Proof.

Given an $f \in \mathcal{F}$ and an $id_f \in Outs(\Delta)$ s.t. $\gamma(f) = id_f$, let us show that $\boxed{s_0.ex(f) = \sigma(id_f).}$
Rewriting $s_0.ex(f)$ as `false`, by definition of s_0, $\boxed{\sigma(id_f) = \texttt{false}.}$
By construction, id_f is an output port identifier of Boolean type in the $\mathcal{H}$-VHDL design d. The generated `function` process assigns a value to the output port id_f only during the initialization phase or during a rising edge phase.
By construction, the `function` process is defined in the behavior of design d, i.e.
$\texttt{ps}(\texttt{function}, \emptyset, sl, ss) \in d.cs$.
Let $trs(f)$ be the set of transitions associated to function f, i.e. $trs(f) = \{t \in T \mid \mathbb{F}(t, f) = true\}$.
Let us perform case analysis on $trs(f)$; there are two cases:

- **CASE** $trs(f) = \emptyset$:

  By construction, $\texttt{id}_\texttt{f} \Leftarrow \texttt{false} \in ss_\uparrow$ where $ss_\uparrow$ is the part of the "`function`" process body executed during a rising edge phase (i.e. a rising edge block statement).

  By property of the $\mathcal{H}$-VHDL rising edge and the stabilize relation, $\sigma(id_f) = \texttt{false}.$

- **CASE** $trs(f) \neq \emptyset$:

  By construction, $\texttt{id}_\texttt{f} \Leftarrow \texttt{id}_{\texttt{ft}_0} + \cdots + \texttt{id}_{\texttt{ft}_n} \in ss_\uparrow$ where $ss_\uparrow$ is the part of the "`function`" process

body executed during the rising edge phase, and $n = |trs(f)| - 1$, and for all $i \in [0, n-1]$, id_{ft_i} is an internal signal of design d.

By property of the `Inject`, the $\mathcal{H}$-VHDL rising edge and stabilize relations, we can deduce $\sigma(id_f) = \sigma_0(id_{ft_0}) + \cdots + \sigma_0(id_{ft_n})$.

Rewriting $\sigma(id_f)$ as $\sigma_0(id_{ft_0}) + \cdots + \sigma_0(id_{ft_n})$, $\boxed{\sigma_0(id_{ft_0}) + \cdots + \sigma_0(id_{ft_n}) = \texttt{false}.}$

By construction, for all id_{ft_i}, there exist a $t_i \in trs(f)$ and an id_{t_i} s.t. $\gamma(t_i) = id_{t_i}$.

By construction and by definition of id_{t_i}, there exist g_{t_i}, i_{t_i} and o_{t_i} s.t. $\texttt{comp}(id_{t_i}, \texttt{transition}, g_{t_i}, i_{t_i}, o_{t_i}) \in d.cs$.

By construction, we have $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}_\texttt{i}}> \in o_{t_i}$, and by property of the initialization relation, we have $\sigma_0(id_{ft_i}) = \sigma_0(id_{t_i})(\texttt{fired})$.

Rewriting $\sigma_0(id_{ft_i})$ as $\sigma_0(id_{t_i})(\texttt{fired})$, $\boxed{\sigma_0(id_{t_0})(\texttt{fired}) + \cdots + \sigma_0(id_{t_n})(\texttt{fired}) = \texttt{false}.}$

Appealing to Lemma 10, we can deduce $\sigma_0(id_{t_i})(\texttt{fired}) = \texttt{false}$.

Rewriting all $\sigma_0(id_{t_i})(\texttt{fired})$ as `false` and simplifying the goal, tautology.

□

5.6 First rising edge and sensitization

Lemma 17 (First rising edge equal sensitized)**.** *For all* $sitpn, b, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma, E_c, E_p, \tau$ *that verify the hypotheses of Definition 11, then*
$\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \in Sens(s_0.M) \Leftrightarrow \sigma(id_t)(\texttt{s\_enabled}) = \texttt{true}$.

Proof.

See the proof of Lemma 29. □

Lemma 18 (First rising edge not equal sensitized)**.** *For all* $sitpn, b, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma, E_c, E_p, \tau$ *that verify the hypotheses of Definition 11, then*
$\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \notin Sens(s_0.M) \Leftrightarrow \sigma(id_t)(\texttt{s\_enabled}) = \texttt{false}$.

Proof.

See the proof of Lemma 30. □

5.7 First rising edge and conditions

Lemma 19 (First rising edge equal condition combination)**.** *For all* $sitpn, b, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma, E_c, E_p, \tau$ *that verify the hypotheses of Definition 11, then*
$\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,

$$\sigma(id_t)(\texttt{s\_condition\_combination}) = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}$$

where $conds(t) = \{c \in \mathcal{C} \mid \mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1\}$.

Proof.

See the proof of Lemma 23. □

Lemma 20 (First rising edge equal conditions)**.** *For all* $sitpn, b, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma, E_c, E_p, \tau$ *that verify the hypotheses of Definition 11, then*
$\forall c \in \mathcal{C}, id_c \in Ins(\Delta)$ *s.t.* $\gamma(c) = id_c$, $\sigma(id_c) = E_c(\tau, c)$.

Proof.

See the proof of Lemma 24. □

6 Rising edge lock-step simulation

Lemma 21 (Rising edge lock-step simulation)**.** *For all well-defined* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e \in \Sigma$, $E_p \in \mathbb{N} \to (id \nrightarrow v)$, *and* $E_c \in \mathbb{N} \to (\mathcal{C} \to \mathbb{B})$ *that verify the hypotheses of Definition 9, and for all* $\tau \in \mathbb{N}$, $s, s' \in S(sitpn)$, $\sigma, \sigma_\uparrow, \sigma' \in \Sigma$, *such that*

- s *and* σ *are similar states as intended after a falling edge step:* $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$
- *a rising edge step leads from* s *to* s'*:* $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$
- *a rising edge step leads from* σ *to* σ'*:*
  $\mathcal{D}_\mathcal{H}, \Delta, \texttt{inj}(\sigma, E_p, \tau) \vdash d.beh \xrightarrow{cs\uparrow} \sigma_\uparrow$ *and* $\mathcal{D}_\mathcal{H}, \Delta, \sigma_\uparrow \vdash d.beh \overset{\rightsquigarrow}{\to} \sigma'$

then $\gamma \vdash s' \overset{\uparrow}{\approx} \sigma'$.

Proof.

By definition of the Full post rising edge state similarity relation, there are 9 points to prove:

1. $\forall p \in P, id_p \in Comps(\Delta)\ s'.t.\ \gamma(p) = id_p,\ \ s'.M(p) = \sigma'(id_p)(\texttt{s\_marking})$.

2. $\forall t \in T_i, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
$\big(u(I_s(t)) = \infty \wedge s'.I(t) \leq l(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)$
$\wedge\big(u(I_s(t)) = \infty \wedge s'.I(t) > l(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)$
$\wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) > u(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)$
$\wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) \leq u(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)$.

3. $\forall t \in T_i, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
$s'.reset_t(t) = \sigma'(id_t)(\texttt{s\_reinit\_time\_counter})$.

4. $\forall a \in \mathcal{A}, id_a \in Outs(\Delta)\ s.t.\ \gamma(a) = id_a,\ \ s'.ex(a) = \sigma'(id_a)$.

5. $\forall f \in \mathcal{F}, id_f \in Outs(\Delta)\ s.t.\ \gamma(f) = id_f,\ \ s'.ex(f) = \sigma'(id_f)$.

6. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
$t \in Sens(s'.M) \Leftrightarrow \sigma'(id_t)(\texttt{s\_enabled}) = \texttt{true}$.

7. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
$t \notin Sens(s'.M) \Leftrightarrow \sigma'(id_t)(\texttt{s\_enabled}) = \texttt{false}$.

8. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
$$\sigma'(id_t)(\texttt{s\_condition\_combination}) = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}$$
where $conds(t) = \{c \in \mathcal{C} \mid \mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1\}$.

9. $\forall c \in \mathcal{C}, id_c \in Ins(\Delta)$ s.t. $\gamma(c) = id_c$, $\sigma'(id_c) = E_c(\tau, c)$.

Each point is proved by a separate lemma:

– Apply the Rising edge equal marking lemma (p. 25) to solve Point 1.

– Apply the Rising edge equal time counters lemma (p. 28) to solve Point 2.

– Apply the Rising edge equal reset orders lemma (p. 29) to solve Point 3.

– Apply the Rising edge equal action executions lemma (p. 37) to solve Point 4.

– Apply the Rising edge equal function executions lemma (p. 37) to solve Point 5.

– Apply the Rising edge equal sensitized lemma (p. 39) to solve Point 6.

– Apply the Rising edge equal not sensitized lemma (p. 43) to solve Point 7.

– Apply the Rising edge equal condition combination lemma (p. 26) to solve Point 8.

– Apply the Rising edge equal conditions lemma (p. 28) to solve Point 9.

All the lemmas used above, and their corresponding proofs, are to be found in Appendix **??**, Section **??**. □

Definition 12 (Rising edge hypotheses)**.** *Given an* $sitpn \in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$, $\Delta \in ElDesign$, $E_p \in \mathbb{N} \to Ins(\Delta) \to value$, $\tau \in \mathbb{N}$, $s, s' \in S(sitpn)$, $\sigma_e, \sigma, \sigma_i, \sigma_\uparrow, \sigma' \in \Sigma$, *assume that:*

- $\lfloor sitpn \rfloor_b = (d, \gamma)$ *and* $\gamma \vdash E_p \overset{env}{=} E_c$ *and* $\mathcal{D}_\mathcal{H}, \emptyset \vdash \mathrm{d} \xrightarrow{elab} \Delta, \sigma_e$
- $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$
- $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$
- $\texttt{Inject}(\sigma, E_p, \tau, \sigma_i)$ *and* $\mathcal{D}_\mathcal{H}, \Delta, \sigma_i \vdash \mathrm{d.cs} \xrightarrow{\uparrow} \sigma_\uparrow$ *and* $\mathcal{D}_\mathcal{H}, \Delta, \sigma_\uparrow \vdash \mathrm{d.cs} \xrightarrow{\rightsquigarrow} \sigma'$
- *State* σ *is a stable design state:* $\mathcal{D}_\mathcal{H}, \Delta, \sigma \vdash \mathrm{d.cs} \xrightarrow{comb} \sigma$

6.1 Rising edge and Marking

Lemma 22 (Rising edge equal marking)**.** *For all sitpn, b, d, γ, E_c, E_p, τ, Δ, σ_e, s, s′, σ, σ_i, $\sigma_\uparrow$, σ′ that verify the hypotheses of Definition 12, then* $\forall p, id_p$ *s.t.* $\gamma(p) = id_p$, $s'.M(p) = \sigma'(id_p)(\texttt{s\_marking})$.

Proof.

Given a $p \in P$, let us show $\boxed{s'.M(p) = \sigma'(id_p)(\texttt{s\_marking}).}$
By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$.
By definition of the SITPN state transition relation on rising edge:

$$s'.M(p) = s.M(p) - \sum_{t \in Fired(s)} pre(p,t) + \sum_{t \in Fired(s)} post(t,p) \tag{7}$$

By property of the `Inject`, the $\mathcal{H}$-VHDL rising edge and the stabilize relations, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `marking` process defined in the place design architecture, we can deduce:

$$\begin{aligned}\sigma'(id_p)(\texttt{sm}) = \sigma(id_p)(\texttt{sm}) &- \sigma(id_p)(\texttt{s\_output\_token\_sum}) \\ &+ \sigma(id_p)(\texttt{s\_input\_token\_sum})\end{aligned} \tag{8}$$

Rewriting the goal with 7 and 8,

$$\boxed{\begin{aligned} s.M(p) - \sum_{t \in Fired(s)} pre(p,t) + \sum_{t \in Fired(s)} post(t,p) \\ = \\ \sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) + \sigma(id_p)(\texttt{sits}) \end{aligned}}$$

By definition of the Full post falling edge state similarity relation, we can deduce $s.M(p) = \sigma(id_p)(\texttt{sm})$, $\sum_{t \in Fired(s)} pre(p,t) = \sigma(id_p)(\texttt{sots})$ and $\sum_{t \in Fired(s)} post(t,p) = \sigma(id_p)(\texttt{sits})$, and thus,

$$s.M(p) - \sum_{t \in Fired(s)} pre(p,t) + \sum_{t \in Fired(s)} post(t,p) = \sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) + \sigma(id_p)(\texttt{sits})$$

□

6.2 Rising edge and conditions

Lemma 23 (Rising edge equal condition combination)**.** *For all sitpn, b, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Definition 12, then*
$\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,

$$\sigma'(id_t)(\texttt{s\_condition\_combination}) = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau,c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}$$

where $conds(t) = \{c \in \mathcal{C} \mid \mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1\}$.

Proof.

Given a t and an id_t s.t. $\gamma(t) = id_t$, let us show

$$\sigma'(id_t)(\texttt{s\_condition\_combination}) = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau,c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}.$$

By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$. By property of the $\mathcal{H}$-VHDL stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `condition_evaluation` process defined in the transition design architecture, we can deduce:

$$\sigma'(id_t)(\texttt{scc}) = \prod_{i=0}^{\Delta(id_t)(\texttt{conditions\_number})-1} \sigma'(id_t)(\texttt{input\_conditions})[i] \tag{9}$$

Rewriting the goal with 9,

$$\prod_{i=0}^{\Delta(id_t)(\texttt{cn})-1} \sigma'(id_t)(\texttt{ic})[i] = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau,c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}.$$

Let us perform case analysis on $conds(t)$; there are two cases:

- **CASE** $conds(t) = \emptyset$: $\prod_{i=0}^{\Delta(id_t)(\texttt{cn})-1} \sigma'(id_t)(\texttt{ic})[i] = \texttt{true}$.

  By construction, $<\texttt{cn} \Rightarrow 1> \in g_t$ and $<\texttt{ic(0)} \Rightarrow \texttt{true}> \in i_t$.

  By property of the stabilize relation, $<\texttt{cn} \Rightarrow 1> \in g_t$ and $<\texttt{ic(0)} \Rightarrow \texttt{true}> \in i_t$, we can deduce $\Delta(id_t)(\texttt{cn}) = 1$ and $\sigma'(id_t)(\texttt{ic})[0] = \texttt{true}$.

  Rewriting the goal with $\Delta(id_t)(\texttt{cn}) = 1$ and $\sigma'(id_t)(\texttt{ic})[0] = \texttt{true}$, tautology.

- **CASE** $conds(t) \neq \emptyset$:
  By construction, $<\texttt{cn} \Rightarrow |\texttt{conds(t)}|> \in g_t$, and by property of the stabilize relation, we can deduce $\Delta(id_t)(\texttt{cn}) = |conds(t)|$.

  Rewriting the goal with $\Delta(id_t)(\texttt{cn}) = |conds(t)|$:

  $$\boxed{\prod_{i=0}^{|conds(t)|-1} \sigma'(id_t)(\texttt{ic})[i] = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}}$$

  There exists a mapping, given by the transformation function, between the set $conds(t)$ and the indexes of $[0, |conds(t)| - 1]$.

  Let $\beta \in conds(t) \to [0, |conds(t)| - 1]$ be this mapping.

  To prove the current goal, it suffices to prove that for all condition $c \in conds(t)$, we have

  $$\boxed{\left(\begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} \right) = \sigma'(id_t)(\texttt{ic})[\beta(c)]}$$

  Given a $c \in conds(t)$, let us show the above goal.

  By construction, for all $c \in conds(t)$, there exists an $id_c \in Ins(\Delta)$ such that

  – $\gamma(c) = id_c$

  – $\mathbb{C}(t,c) = 1$ implies $<\texttt{ic}(\beta(c)) \Rightarrow \texttt{id}_\texttt{c}> \in i_t$

  – $\mathbb{C}(t,c) = -1$ implies $<\texttt{ic}(\beta(c)) \Rightarrow \texttt{not id}_\texttt{c}> \in i_t$

  Let us take such an id_c with the above properties.

  By definition of $c \in conds(t)$, we have $\mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1$. Let us perform case analysis on $\mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1$:

  – **CASE** $\mathbb{C}(t,c) = 1$:

    In that case, we must show: $\boxed{E_c(\tau, c) = \sigma'(id_t)(\texttt{ic})[\beta(c)]}$

    By assumption, we have $<\texttt{ic}(\beta(c)) \Rightarrow \texttt{id}_\texttt{c}> \in i_t$ and by property of the stabilize relation, we can deduce $\sigma(id_t)(\texttt{ic})[\beta(c)] = \sigma'(id_c)$.

    Rewriting the goal with $\sigma(id_t)(\texttt{ic})[\beta(c)] = \sigma'(id_c)$:

    $\boxed{E_c(\tau, c) = \sigma'(id_c)}$

    By property of the `Inject` relation and $id_c \in Ins(\Delta)$, we can deduce $\sigma'(id_c) = E_p(\tau)(id_c)$.

    By property of $\gamma \vdash E_p \overset{env}{=} E_c$, we can deduce $E_p(\tau)(id_c) = E_c(\tau, c)$.

    Rewriting the goal with $\sigma'(id_c) = E_p(\tau)(id_c)$ and $E_p(\tau)(id_c) = E_c(\tau, c)$: $\boxed{E_c(\tau, c) = E_c(\tau, c)}$, then tautology.

– **CASE** $\mathbb{C}(t,c) = -1$:

In that case, we must show: $\boxed{\texttt{not } E_c(\tau,c) = \sigma'(id_t)(\texttt{ic})[\beta(c)]}$

By assumption, we have $<\texttt{ic}(\beta(c)) \Rightarrow \texttt{not id}_\texttt{c}> \in i_t$ and by property of the stabilize relation, we can deduce $\sigma(id_t)(\texttt{ic})[\beta(c)] = \texttt{not } \sigma'(id_c)$.

Rewriting the goal with $\sigma(id_t)(\texttt{ic})[\beta(c)] = \texttt{not } \sigma'(id_c)$:

$\boxed{\texttt{not } E_c(\tau,c) = \texttt{not } \sigma'(id_c)}$

By property of the `Inject` relation and $id_c \in Ins(\Delta)$, we can deduce $\sigma'(id_c) = E_p(\tau)(id_c)$.

By property of $\gamma \vdash E_p \overset{env}{=} E_c$, we can deduce $E_p(\tau)(id_c) = E_c(\tau,c)$.

Rewriting the goal with $\sigma'(id_c) = E_p(\tau)(id_c)$ and $E_p(\tau)(id_c) = E_c(\tau,c)$:

$\boxed{\texttt{not } E_c(\tau,c) = \texttt{not } E_c(\tau,c)}$, then tautology.

□

Lemma 24 (Rising edge equal conditions)**.** *For all sitpn, b, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Definition 12, then*
$\forall c \in \mathcal{C}, id_c \in Ins(\Delta)$ *s.t.* $\gamma(c) = id_c$, $\sigma'(id_c) = E_c(\tau,c)$.

Proof.

Given a $c \in \mathcal{C}$ and an $id_c \in Ins(\Delta)$ such that $\gamma(c) = id_c$, let us show

$$\boxed{\sigma'(id_c) = E_c(\tau,c)}$$

By property of the `Inject` relation and $id_c \in Ins(\Delta)$, we can deduce $\sigma'(id_c) = E_p(\tau)(id_c)$.
By property of $\gamma \vdash E_p \overset{env}{=} E_c$, we can deduce $E_p(\tau)(id_c) = E_c(\tau,c)$.
Rewriting the goal with $\sigma'(id_c) = E_p(\tau)(id_c)$ and $E_p(\tau)(id_c) = E_c(\tau,c)$, tautology. □

6.3 Rising edge and time counters

Lemma 25 (Rising edge equal time counters)**.** *For all sitpn, b, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Definition 12, then*
$\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,
$\big(u(I_s(t)) = \infty \wedge s'.I(t) \leq l(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)$
$\wedge\big(u(I_s(t)) = \infty \wedge s'.I(t) > l(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)$
$\wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) > u(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)$
$\wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) \leq u(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)$.

Proof.

Given a $t \in T_i$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show

$$\boxed{\begin{array}{l}\big(u(I_s(t)) = \infty \wedge s'.I(t) \leq l(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)\\ \wedge\big(u(I_s(t)) = \infty \wedge s'.I(t) > l(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)\\ \wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) > u(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)\\ \wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) \leq u(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)\end{array}}$$

By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$. Then, there are 4 points to show:

1. $\boxed{u(I_s(t)) = \infty \wedge s'.I(t) \leq l(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})}$

   Assuming that $u(I_s(t)) = \infty$ and $s'.I(t) \leq l(I_s(t))$, let us show
   $\boxed{s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter}).}$

   By property of the `Inject`, $\mathcal{H}$-VHDL rising edge and stabilize relations, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `time_counter` process defined in the transition design architecture, we can deduce $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})$.

   By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, we can deduce $s.I(t) = \sigma(id_t)(\texttt{stc})$.

   Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})$ and $s.I(t) = \sigma(id_t)(\texttt{stc})$, tautology.

2. $\boxed{u(I_s(t)) = \infty \wedge s'.I(t) > l(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = l(I_s(t).}$

   Proved in the same fashion as 1.

3. $\boxed{u(I_s(t)) \neq \infty \wedge s'.I(t) > u(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = u(I_s(t).}$

   Proved in the same fashion as 1.

4. $\boxed{u(I_s(t)) \neq \infty \wedge s'.I(t) \leq u(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})}$

   Proved in the same fashion as 1.

□

6.4 Rising edge and reset orders

Lemma 26 (Rising edge equal reset orders)**.** *For all sitpn, b, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Definition 12, then*
$\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $s'.reset_t(t) = \sigma'(id_t)(\texttt{s\_reinit\_time\_counter})$

Proof.

Given a $t \in T_i$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show
$\boxed{s'.reset_t(t) = \sigma'(id_t)(\texttt{s\_reinit\_time\_counter}).}$

By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$. By property of the $\mathcal{H}$-VHDL stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `reinit_time_counter_evaluation` process defined in the transition design architecture, we can deduce:

$$\sigma'(id_t)(\texttt{srtc}) = \sum_{i=0}^{\Delta(id_t)(\texttt{input\_arcs\_number})-1} \sigma'(id_t)(\texttt{reinit\_time})[i] \tag{10}$$

Rewriting the goal with (10), $\boxed{s'.reset_t(t) = \sum_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{rt})[i].}$

Let us perform case analysis on $input(t)$; there are two cases:

- **CASE** $input(t) = \emptyset$:

  By construction, $<\texttt{input\_arcs\_number} \Rightarrow 1> \in g_t$, and by property of the elaboration relation, we can deduce $\Delta(id_t)(\texttt{ian}) = 1$.

  By construction, there exists an $id_{ft} \in Sigs(\Delta)$ s.t. $<\texttt{reinit\_time}(0) \Rightarrow \texttt{id}_{\texttt{ft}}> \in i_t$ and $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}}> \in o_t$, and by property of the stabilize relation and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\sigma'(id_t)(\texttt{rt})[0] = \sigma'(id_{ft}) = \sigma'(id_t)(\texttt{fired})$.

  Rewriting the goal with $\Delta(id_t)(\texttt{ian}) = 1$ and $\sigma'(id_t)(\texttt{rt})[0] = \sigma'(id_{ft}) = \sigma'(id_t)(\texttt{fired})$: $\boxed{s'.reset_t(t) = \sigma'(id_t)(\texttt{fired}).}$

  By property of the stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `fired_evaluation` process, we can deduce:

  $$\sigma'(id_t)(\texttt{fired}) = \sigma'(id_t)(\texttt{s\_firable}) \;.\; \sigma'(id_t)(\texttt{s\_priority\_combination}) \tag{11}$$

  Rewriting the goal with (11):
  $\boxed{s'.reset_t(t) = \sigma'(id_t)(\texttt{s\_firable}) \;.\; \sigma'(id_t)(\texttt{s\_priority\_combination}).}$

  By property of the stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `priority_authorization_evaluation` process defined in the transition design architecture, we can deduce:

  $$\sigma'(id_t)(\texttt{spc}) = \prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{priority\_authorizations})[i] \tag{12}$$

  As $\Delta(id_t)(\texttt{ian}) = 1$, we can deduce $\prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{pauths})[i] = \sigma'(id_t)(\texttt{pauths})[0]$.

  Rewriting the goal with (12) and $\prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{pauths})[i] = \sigma'(id_t)(\texttt{pauths})[0]$:
  $\boxed{s'.reset_t(t) = \sigma'(id_t)(\texttt{s\_firable}) \;.\; \sigma'(id_t)(\texttt{pauths})[0].}$

By construction, <$\texttt{priority\_authorizations}(0) \Rightarrow \texttt{true}$> $\in i_t$, and by property of the stabilize relation and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\sigma'(id_t)(\texttt{pauths})[0] = \texttt{true}$.

Rewriting the goal with $\sigma'(id_t)(\texttt{pauths})[0] = \texttt{true}$, and simplifying the equation: $\boxed{s'.reset_t(t) = \sigma'(id_t)(\texttt{s\_firable}).}$

Let us perform case analysis on $t \in Fired(s)$ or $t \notin Fired(s)$:

– **CASE** $t \in Fired(s)$:

By property of $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$ (Rule **??**), we can deduce $s'.reset_t(t) = \texttt{true}$.

Rewriting the goal with $s'.reset_t(t) = \texttt{true}$: $\boxed{\sigma'(id_t)(\texttt{s\_firable}) = \texttt{true}.}$

By property of the stabilize, the $\mathcal{H}$-VHDL rising edge and the `Inject` relations, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `firable` process defined in the transition design architecture, we can deduce $\sigma(id_t)(\texttt{s\_firable}) = \sigma'(id_t)(\texttt{s\_firable})$.

Rewriting the goal with $\sigma(id_t)(\texttt{s\_firable}) = \sigma'(id_t)(\texttt{s\_firable})$, we have $\boxed{\sigma(id_t)(\texttt{s\_firable}) = \texttt{true}.}$

By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, we can deduce $t \in Firable(s) \Leftrightarrow \sigma(id_t)(\texttt{sfa}) = \texttt{true}$.

Rewriting the goal with $t \in Firable(s) \Leftrightarrow \sigma(id_t)(\texttt{sfa}) = \texttt{true}$, $\boxed{t \in Firable(s).}$

By property of $t \in Fired(s)$, $t \in Firable(s)$.

– **CASE** $t \notin Fired(s)$:

By property of $input(t) = \emptyset$, there does not exist any input place connected to t by a `basic` or `test` arc. Thus, by property of $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$ (Rule **??**), we can deduce $s'.reset_t(t) = \texttt{false}$.

Rewriting the goal with $s'.reset_t(t) = \texttt{false}$: $\boxed{\sigma'(id_t)(\texttt{s\_firable}) = \texttt{false}.}$

By property of the stabilize, the $\mathcal{H}$-VHDL rising edge and the `Inject` relations, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `firable` process defined in the transition design architecture, we can deduce $\sigma(id_t)(\texttt{sfa}) = \sigma'(id_t)(\texttt{sfa})$.

Rewriting the goal with $\sigma(id_t)(\texttt{sfa}) = \sigma'(id_t)(\texttt{sfa})$, $\boxed{\sigma(id_t)(\texttt{sfa}) = \texttt{false}.}$

By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, we can deduce $t \notin Firable(s) \Leftrightarrow \sigma(id_t)(\texttt{sfa}) = \texttt{false}$.

By property of $t \notin Fired(s)$ and $input(t) = \emptyset$, $t \notin Firable(s)$.

- **CASE** $input(t) \neq \emptyset$:

By construction, <$\texttt{input\_arcs\_number} \Rightarrow |input(t)|$> $\in g_t$, and by property of the elaboration relation, we can deduce $\Delta(id_t)(\texttt{ian}) = |input(t)|$.

Rewriting the goal with $\Delta(id_t)(\texttt{ian}) = |input(t)|$, $\boxed{s'.reset_t(t) = \sum_{i=0}^{|input(t)|-1} \sigma'(id_t)(\texttt{rt})[i].}$

Let us perform case analysis on $t \in Fired(s)$ or $t \notin Fired(s)$:

– **CASE** $t \in Fired(s)$:

By property of $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$ (Rule **??**), we can deduce $s'.reset_t(t) = \texttt{true}$.

Rewriting the goal with $s'.reset_t(t) = \texttt{true}$, $\boxed{\sum_{i=0}^{|input(t)|-1} \sigma'(id_t)(\texttt{rt})[i] = \texttt{true}.}$

To prove the goal, let us show $\boxed{\exists i \in [0, |input(t)| - 1] \text{ s.t. } \sigma'(id_t)(\texttt{rt})[i] = \texttt{true}.}$

By construction, and $input(t) \neq \emptyset$, there exist $p \in input(t)$ and $id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$.

By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$.

By construction, there exist an $i \in [0, |input(t)| - 1]$, a $j \in [0, |output(p)| - 1]$ and $id_{ji} \in Sigs(\Delta)$ s.t. $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in o_p$ and $<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in i_t$. Let us take such an i, j and id_{ji}, and let us use i to prove the goal: $\boxed{\sigma'(id_t)(\texttt{rt})[i] = \texttt{true}.}$

By property of the stabilize relation, $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in o_p$ and $<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in i_t$, we can deduce $\sigma'(id_t)(\texttt{rt})[i] = \sigma'(id_{ji}) = \sigma'(id_p)(\texttt{rtt})[j]$.

Rewriting the goal with $\sigma'(id_t)(\texttt{rt})[i] = \sigma'(id_{ji}) = \sigma'(id_p)(\texttt{rtt})[j]$, $\boxed{\sigma'(id_p)(\texttt{rtt})[j] = \texttt{true}.}$

By property of the `Inject`, the $\mathcal{H}$-VHDL rising edge and the stabilize relations, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `reinit_transitions_time_evaluation` process defined in the place design architecture, we can deduce:

$$\begin{aligned}\sigma'(id_p)(\texttt{rtt})[j] = &\big((\sigma(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma(id_p)(\texttt{oat})[j] = \texttt{test}) \\ &.(\sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) < \sigma(id_p)(\texttt{oaw})[j]) \\ &.(\sigma(id_p)(\texttt{sots}) > 0)\big) \\ &+ \sigma(id_p)(\texttt{otf})[j]\end{aligned} \tag{13}$$

Rewriting the goal with (13),

$$\boxed{\begin{aligned}\texttt{true} = &((\sigma(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma(id_p)(\texttt{oat})[j] = \texttt{test}) \\ &.(\sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) < \sigma(id_p)(\texttt{oaw})[j]) \\ &.(\sigma(id_p)(\texttt{sots}) > 0)) \\ &+ (\sigma(id_p)(\texttt{otf})[j])\end{aligned}}$$

By construction, there exists $id_{ft} \in Sigs(\Delta)$ such that $<\texttt{output\_transitions\_fired(j)} \Rightarrow \texttt{id}_{\texttt{ft}}> \in i_p$ and $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}}> \in o_t$. By property of state σ, which is a stable state, we have $\sigma(id_t)(\texttt{fired}) = \sigma(id_{ft}) = \sigma(id_p)(\texttt{otf})[j]$.

Rewriting the goal with $\sigma(id_t)(\texttt{fired}) = \sigma(id_{ft}) = \sigma(id_p)(\texttt{otf})[j]$,

$$\boxed{\begin{aligned}\texttt{true} = &((\sigma(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma(id_p)(\texttt{oat})[j] = \texttt{test}) \\ &.(\sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) < \sigma(id_p)(\texttt{oaw})[j]) \\ &.(\sigma(id_p)(\texttt{sots}) > 0)) \\ &+ \sigma(id_t)(\texttt{fired})\end{aligned}}$$

By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, we can deduce $t \in Fired(s) \Leftrightarrow \sigma(id_t)(\texttt{fired}) = \texttt{true}$.
Rewriting the goal with $t \in Fired(s) \Leftrightarrow \sigma(id_t)(\texttt{fired}) = \texttt{true}$ and simplify the goal, then tautology.

– **CASE** $t \notin Fired(s)$: Then, there are two cases that will determine the value of $s'.reset_t(t)$. Either there exists a place p with an output token sum greater than zero, that is connected to t by an `basic` or `test` arc, and such that the transient marking of p disables t; or such a place does not exist (the predicate is decidable).

∗ **CASE** there exists such a place p as described above:
Then, let us take such a place p and $\omega \in \mathbb{N}^*$ s.t.:

1. $\sum_{t_i \in Fired(s)} pre(p, t_i) > 0$
2. $pre(p,t) = (\omega, \texttt{basic}) \vee pre(p,t) = (\omega, \texttt{test})$
3. $s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega$

We will only consider the case where $pre(p,t) = (\omega, \texttt{basic})$; the proof is the similar when $pre(p,t) = (\omega, \texttt{test})$.

Assuming that p exists, and by property of $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$ (Rule **??**), we can deduce $s'.reset_t(t) = \texttt{true}$.
Rewriting the goal with $s'.reset_t(t) = \texttt{true}$, $\boxed{\sum_{i=0}^{|input(t)|-1} \sigma'(id_t)(\texttt{rt})[i] = \texttt{true}.}$
To prove the goal, let us show $\boxed{\exists i \in [0, |input(t)| - 1] \text{ s.t. } \sigma'(id_t)(\texttt{rt})[i] = \texttt{true}.}$
By construction, there exists $id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$.
By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$.
By construction, there exist an $i \in [0, |input(t)| - 1]$, a $j \in [0, |output(p)| - 1]$ and $id_{ji} \in Sigs(\Delta)$ s.t. $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in o_p$ and
$<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in i_t$. Let us take such an i, j and id_{ji}, and let us use i to prove the goal: $\boxed{\sigma'(id_t)(\texttt{rt})[i] = \texttt{true}.}$
By property of the stabilize relation, $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in o_p$ and $<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in i_t$, we have $\sigma'(id_t)(\texttt{rt})[i] = \sigma'(id_{ji}) = \sigma'(id_p)(\texttt{rtt})[j]$.
Rewriting the goal with $\sigma'(id_t)(\texttt{rt})[i] = \sigma'(id_{ji}) = \sigma'(id_p)(\texttt{rtt})[j]$, we have
$\boxed{\sigma'(id_p)(\texttt{rtt})[j] = \texttt{true}.}$
By property of the `Inject`, the $\mathcal{H}$-VHDL rising edge and the stabilize relation, and through the examination of the `reinit_transitions_time_evaluation` process defined in the place design architecture, we can deduce:

$$\begin{aligned}\sigma'(id_p)(\texttt{rtt})[j] = &\big((\sigma(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma(id_p)(\texttt{oat})[j] = \texttt{test})\\ &.(\sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) < \sigma(id_p)(\texttt{oaw})[j])\\ &.(\sigma(id_p)(\texttt{sots}) > 0)\big)\\ &+ \sigma(id_p)(\texttt{otf})[j]\end{aligned} \tag{14}$$

Rewriting the goal with (14),

$$\boxed{\begin{array}{l}\texttt{true} = ((\sigma(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma(id_p)(\texttt{oat})[j] = \texttt{test}) \\ \quad .(\sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) < \sigma(id_p)(\texttt{oaw})[j]) \\ \quad .(\sigma(id_p)(\texttt{sots}) > 0)) \\ \quad + \sigma(id_p)(\texttt{otf})[j]\end{array}}$$

By construction, $<\texttt{output\_arcs\_types(j)} \Rightarrow \texttt{basic}> \in i_p$ and $<\texttt{output\_arcs\_weights(j)} \Rightarrow \omega> \in i_p$.
By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma'(id_p)(\texttt{oat})[j] = \texttt{basic}$ and $\sigma'(id_p)(\texttt{oaw})[j] = \omega$.
By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, we can deduce $\sigma(id_p)(\texttt{sm}) = s.M(p)$ and $\sigma(id_p)(\texttt{sots}) = \sum_{t_i \in Fired(s)} pre(p, t_i)$.
Rewriting the goal with $\sigma'(id_p)(\texttt{oat})[j] = \texttt{basic}$, $\sigma'(id_p)(\texttt{oaw})[j] = \omega$, $\sigma(id_p)(\texttt{sm}) = s.M(p)$ and $\sigma(id_p)(\texttt{sots}) = \sum_{t_i \in Fired(s)} pre(p, t_i)$, and simplifying the goal:

$$\boxed{\begin{array}{c}\left((s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega) \ . \ (\sum_{t_i \in Fired(s)} pre(p, t_i) > 0)\right) + \sigma(id_t)(\texttt{fired}) \\ = \\ \texttt{true}\end{array}}$$

We assumed that $s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega$ and $\sum_{t_i \in Fired(s)} pre(p, t_i) > 0$.
Thus, by assumption:

$$\begin{array}{c}\left((s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega) \ . \ (\sum_{t_i \in Fired(s)} pre(p, t_i) > 0)\right) + \sigma(id_t)(\texttt{fired}) \\ = \\ \texttt{true}\end{array}$$

∗ **CASE** such a place does not exist:
Then, let us assume that, for all place $p \in P$

1. $\sum_{t_i \in Fired(s)} pre(p, t_i) = 0$
2. or $\forall \omega \in \mathbb{N}^*, \ pre(p, t) = (\omega, \texttt{basic}) \vee pre(p, t) = (\omega, \texttt{test}) \Rightarrow s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) \geq \omega$.

In that case, by property of $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$ (Rule **??**), we can deduce $s'.reset_t(t) = \texttt{false}$.
Rewriting the goal with $s'.reset_t(t) = \texttt{false}$: $\boxed{\sum_{i=0}^{|input(t)|-1} \sigma'(id_t)(\texttt{rt})[i] = \texttt{false}.}$
To prove the goal, let us show $\boxed{\forall i \in [0, |input(t)| - 1], \ \sigma'(id_t)(\texttt{rt})[i] = \texttt{false}.}$

Given an $i \in [0, |input(t)| - 1]$, let us show $\boxed{\sigma'(id_t)(\texttt{rt})[i] = \texttt{false}.}$

By construction, there exist a $p \in input(t)$, an $id_p \in Comps(\Delta)$, g_p, i_p, o_p, a $j \in [0, |output(p)| - 1]$, an $id_{ji} \in Sigs(\Delta)$ s.t. $\gamma(p) = id_p$ and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$ and $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in o_p$ and $<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in i_t$. Let us take such a p, id_p, g_p, i_p, o_p, j and id_{ji}.

By property of the stabilize relation, $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in o_p$ and $<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in i_t$, we have $\sigma'(id_t)(\texttt{rt})[i] = \sigma'(id_{ji}) = \sigma'(id_p)(\texttt{rtt})[j]$.
Rewriting the goal with $\sigma'(id_t)(\texttt{rt})[i] = \sigma'(id_{ji}) = \sigma'(id_p)(\texttt{rtt})[j]$:
$\boxed{\sigma'(id_p)(\texttt{rtt})[j] = \texttt{false}.}$
By property of the `Inject`, the $\mathcal{H}$-VHDL rising edge and the stabilize relations, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `reinit_transitions_time_evaluation` process defined in the place design architecture, we can deduce:

$$\begin{aligned}\sigma'(id_p)(\texttt{rtt})[j] = &\big((\sigma(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma(id_p)(\texttt{oat})[j] = \texttt{test})\\ &.(\sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) < \sigma(id_p)(\texttt{oaw})[j])\\ &.(\sigma(id_p)(\texttt{sots}) > 0)\big)\\ &+ \sigma(id_p)(\texttt{otf})[j]\end{aligned} \tag{15}$$

Rewriting the goal with (15),

$$\boxed{\begin{aligned}\texttt{false} = &((\sigma(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma(id_p)(\texttt{oat})[j] = \texttt{test})\\ &.(\sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) < \sigma(id_p)(\texttt{oaw})[j])\\ &.(\sigma(id_p)(\texttt{sots}) > 0))\\ &+ \sigma(id_p)(\texttt{otf})[j])\end{aligned}}$$

By construction, there exists $id_{ft} \in Sigs(\Delta)$ such that $<\texttt{output\_transitions\_fired(j)} \Rightarrow \texttt{id}_{\texttt{ft}}> \in i_p$ and $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}}> \in o_t$. By property of state σ as being a stable state, we have $\sigma(id_t)(\texttt{fired}) = \sigma(id_{ft}) = \sigma(id_p)(\texttt{otf})[j]$.
Rewriting the goal with $\sigma(id_t)(\texttt{fired}) = \sigma(id_{ft}) = \sigma(id_p)(\texttt{otf})[j]$:

$$\boxed{\begin{aligned}\texttt{false} = &((\sigma(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma(id_p)(\texttt{oat})[j] = \texttt{test})\\ &.(\sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) < \sigma(id_p)(\texttt{oaw})[j])\\ &.(\sigma(id_p)(\texttt{sots}) > 0))\\ &+ \sigma(id_t)(\texttt{fired})\end{aligned}}$$

By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, we can deduce $t \notin Fired(s) \Leftrightarrow \sigma(id_t)(\texttt{fired}) = \texttt{false}$
Rewriting the goal with $t \notin Fired(s) \Leftrightarrow \sigma(id_t)(\texttt{fired}) = \texttt{false}$ and simplifying the goal:

$$\boxed{\begin{aligned}\texttt{false} = &((\sigma(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma(id_p)(\texttt{oat})[j] = \texttt{test})\\ &.(\sigma(id_p)(\texttt{sm}) - \sigma(id_p)(\texttt{sots}) < \sigma(id_p)(\texttt{oaw})[j])\\ &.(\sigma(id_p)(\texttt{sots}) > 0))\end{aligned}}$$

Then, based on the assumptions made at the beginning of case, there are two cases:

1. **CASE** $\sum_{t_i \in Fired(s)} pre(p, t_i) = 0$:

   By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, we can deduce $\sum_{t_i \in Fired(s)} pre(p, t_i) = \sigma(id_p)(\texttt{sots})$.

   Rewriting the goal with $\sum_{t_i \in Fired(s)} pre(p, t_i) = \sigma(id_p)(\texttt{sots})$ and $\sum_{t_i \in Fired(s)} pre(p, t_i) = 0$, and simplifying the goal: tautology.

2. **CASE** $\forall \omega \in \mathbb{N}^*,\ pre(p, t) = (\omega, \texttt{basic}) \vee pre(p, t) = (\omega, \texttt{test}) \Rightarrow s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) \geq \omega$:

   Let us perform case analysis on $pre(p, t)$; there are two cases:

   (a) **CASE** $pre(p, t) = (\omega, \texttt{basic})$ or $pre(p, t) = (\omega, \texttt{basic})$:

   By construction, $<\texttt{output\_arcs\_weights(j)} \Rightarrow \omega> \in i_p$.

   By property of stable state σ and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma(id_p)(\texttt{oaw})[j] = \omega$.

   By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, we can deduce $\sigma(id_p)(\texttt{sm}) = s.M(p)$ and $\sigma(id_p)(\texttt{sots}) = \sum_{t_i \in Fired(s)} pre(p, t_i)$.

   Rewriting the goal with $\sigma(id_p)(\texttt{oaw})[j] = \omega$, $\sigma(id_p)(\texttt{sm}) = s.M(p)$ and $\sigma(id_p)(\texttt{sots}) = \sum_{t_i \in Fired(s)} pre(p, t_i)$:

   $$\begin{aligned}\texttt{false} =& ((\sigma(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma(id_p)(\texttt{oat})[j] = \texttt{test}) \\ & .(s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega) \\ & .(\sum_{t_i \in Fired(s)} pre(p, t_i) > 0))\end{aligned}$$

   We assumed that $s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) \geq \omega$, and then we can deduce $s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega = \texttt{false}$.

   Rewriting the goal with $s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega = \texttt{false}$, and simplifying the goal, tautology.

   (b) **CASE** $pre(p, t) = (\omega, \texttt{inhib})$:

   By construction, $<\texttt{output\_arcs\_types(j)} \Rightarrow \texttt{inhib}> \in i_p$.

   By property of stable state σ and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma(id_p)(\texttt{oat})[j] = \texttt{inhib}$.

   Rewriting the goal with $\sigma(id_p)(\texttt{oat})[j] = \texttt{inhib}$, and simplifying the goal, we have a tautology.

□

6.5 Rising edge and action executions

Lemma 27 (Rising edge equal action executions)**.** *For all sitpn, b, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Definition 12, then*
$\forall a \in \mathcal{A}, id_a \in Outs(\Delta)$ *s.t.* $\gamma(a) = id_a,\ s'.ex(a) = \sigma'(id_a)$.

Proof.

Given an $a \in \mathcal{A}$ and an $id_a \in Outs(\Delta)$ *s.t.* $\gamma(a) = id_a$, let us show $\boxed{s'.ex(a) = \sigma'(id_a).}$

By property of $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$, we can deduce $s.ex(a) = s'.ex(a)$.
By construction, id_a is an output port identifier of Boolean type in the $\mathcal{H}$-VHDL design d. The generated ‘‘`action`’’ process is responsible for the assignment of the *id* a only during the initialization phase or during a falling edge phase.
By property of the $\mathcal{H}$-VHDL `Inject`, rising edge, stabilize relations, and the ‘‘`action`’’ process, we can deduce $\sigma(id_a) = \sigma'(id_a)$.
Rewriting the goal with $s.ex(a) = s'.ex(a)$ and $\sigma(id_a) = \sigma'(id_a)$, $\boxed{s.ex(a) = \sigma(id_a).}$

By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, $s.ex(a) = \sigma(id_a)$. □

6.6 Rising edge and function executions

Lemma 28 (Rising edge equal function executions)**.** *For all sitpn, b, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Definition 12, then*
$\forall f \in \mathcal{F}, id_f \in Outs(\Delta)$ *s.t.* $\gamma(f) = id_f,\ s'.ex(f) = \sigma'(id_f)$.

Proof.

Given an $f \in \mathcal{F}$ and an $id_f \in Outs(\Delta)$ s.t. $\gamma(f) = id_f$, let us show $\boxed{s'.ex(f) = \sigma'(id_f).}$

By property of $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$ (Rule **??**):

$$s'.ex(f) = \sum_{t \in Fired(s)} \mathbb{F}(t, f) \tag{16}$$

By construction, id_f is an output port identifier of Boolean type in the $\mathcal{H}$-VHDL design d. The generated `function` process assigns a value to the output port id_f only during the initialization phase or during a rising edge phase.
By construction, the `function` process is defined in the behavior of design d, i.e.
$\texttt{ps}(\texttt{function}, \emptyset, sl, ss) \in d.cs$.
Let $trs(f)$ be the set of transitions associated to function f, i.e. $trs(f) = \{t \in T \mid \mathbb{F}(t, f) = \texttt{true}\}$.
Let us perform case analysis on $trs(f)$; there are two cases:

- **CASE** $trs(f) = \emptyset$:

  By construction, $\texttt{id}_\texttt{f} \Leftarrow \texttt{false} \in ss_\uparrow$ where $ss_\uparrow$ is the part of the `function` process body executed during a rising edge phase.

By property of the $\mathcal{H}$-VHDL rising edge, the stabilize relations and $\texttt{ps}(\texttt{function}, \emptyset, sl, ss) \in d.cs$, we can deduce $\sigma'(id_f) = \texttt{false}$.

By property of $\sum_{t \in Fired(s)} \mathbb{F}(t, f)$ and $trs(f) = \emptyset$, we can deduce $\sum_{t \in Fired(s)} \mathbb{F}(t, f) = \texttt{false}$.

Rewriting the goal with (16), $\sigma'(id_f) = \texttt{false}$ and $\sum_{t \in Fired(s)} \mathbb{F}(t, f) = \texttt{false}$: tautology.

- **CASE** $trs(f) \neq \emptyset$:

  By construction, $\texttt{id}_\texttt{f} \Leftarrow \texttt{id}_{\texttt{ft}_0} + \cdots + \texttt{id}_{\texttt{ft}_n} \in ss_\uparrow$, where $id_{ft_i} \in Sigs(\Delta)$, $ss_\uparrow$ is the part of the `function` process body executed during a rising edge phase, and $n = |trs(f)| - 1$.

  By property of the `Inject`, the $\mathcal{H}$-VHDL rising edge, the stabilize relations, and $\texttt{ps}(\texttt{function}, \emptyset, sl, ss) \in d.cs$, we can deduce:

  $$\sigma'(id_f) = \sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}) \tag{17}$$

  Rewriting the goal with (16) and (17), $\boxed{\sum_{t \in Fired(s)} \mathbb{F}(t, f) = \sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}).}$

  Let us reason on the value of $\sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n})$; there are two cases:

  – **CASE** $\sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}) = \texttt{true}$:

    Then, we can rewrite the goal as follows: $\boxed{\sum_{t \in Fired(s)} \mathbb{F}(t, f) = \texttt{true}.}$

    To prove the above goal, let us show $\boxed{\exists t \in Fired(s)\ s.t.\ \mathbb{F}(t, f) = \texttt{true}.}$

    From $\sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}) = \texttt{true}$, we can deduce $\exists id_{ft_i}\ s.t.\ \sigma(id_{ft_i}) = \texttt{true}$. Let us take such an id_{ft_i}.

    By construction, there exist a $t \in trs(f)$, an $id_t \in Comps(\Delta)$, g_t, i_t, o_t such that:

    * $\gamma(t) = id_t$
    * $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$
    * $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}_i}> \in o_t$

    By property of σ as being a stable design state, and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\sigma(id_t)(\texttt{fired}) = \sigma(id_{ft_i})$, and thus that $\sigma(id_t)(\texttt{fired}) = \texttt{true}$.

    By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, we can deduce $t \in Fired(s)$.

    Let us use t to prove the goal: $\boxed{\mathbb{F}(t, f) = \texttt{true}.}$

    By definition of $t \in trs(f)$, $\mathbb{F}(t, f) = \texttt{true}$.

  – **CASE** $\sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}) = \texttt{false}$:

    Then, we can rewrite the goal as follows: $\boxed{\sum_{t \in Fired(s)} \mathbb{F}(t, f) = \texttt{false}.}$

    To prove the above goal, let us show $\boxed{\forall t \in Fired(s)\ s.t.\ \mathbb{F}(t, f) = \texttt{false}.}$

    Given a $t \in Fired(s)$, let us show $\boxed{\mathbb{F}(t, f) = \texttt{false}.}$

    Let us perform case analysis on $\mathbb{F}(t, f)$; there are 2 cases:

* **CASE** $\mathbb{F}(t, f) = \texttt{false}$.
* **CASE** $\mathbb{F}(t, f) = \texttt{true}$:

  By construction, there exist an $id_t \in Comps(\Delta)$, g_t, i_t, o_t and $id_{ft_i} \in Sigs(\Delta)$ such that:
  - $\gamma(t) = id_t$
  - $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$
  - $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}_\texttt{i}}> \in o_t$

  By property of stable design state σ and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\sigma(id_t)(\texttt{fired}) = \sigma(id_{ft_i})$.
  By property of $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$, we can deduce $t \in Fired(s) \Leftrightarrow \sigma(id_t)(\texttt{fired}) = \texttt{true}$.
  Since $t \in Fired(s)$, we can deduce $\sigma(id_t)(\texttt{fired}) = \texttt{true}$, and from $\sigma(id_t)(\texttt{fired}) = \sigma(id_{ft_i})$, we can deduce $\sigma(id_{ft_i}) = \texttt{true}$.
  Then, $\sigma(id_{ft_i}) = \texttt{true}$ contradicts $\sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}) = \texttt{false}$.

□

6.7 Rising edge and sensitization

Lemma 29 (Rising edge equal sensitized)**.** *For all sitpn, b, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Definition 12, then*
$\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \in Sens(s'.M) \Leftrightarrow \sigma'(id_t)(\texttt{s\_enabled}) = \texttt{true}$.

Proof.

Given a $t \in T$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show
$\boxed{t \in Sens(s'.M) \Leftrightarrow \sigma'(id_t)(\texttt{s\_enabled}) = \texttt{true}.}$

By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$.. Then, the proof is in two parts:

1. Assuming that $t \in Sens(s'.M)$, let us show $\boxed{\sigma'(id_t)(\texttt{s\_enabled}) = \texttt{true}.}$

   By property of the stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `enable_evaluation` process defined in the transition design architecture:

$$\sigma'(id_t)(\texttt{se}) = \prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{input\_arcs\_valid})[i] \tag{18}$$

   Rewriting the goal with (18), $\boxed{\prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{iav})[i] = \texttt{true}.}$

   To prove the goal, let us show that $\boxed{\forall i \in [0, \Delta(id_t)(\texttt{ian}) - 1],\ \sigma'(id_t)(\texttt{iav})[i] = \texttt{true}.}$

Given an $i \in [0, \Delta(id_t)(\texttt{ian}) - 1]$, let us show $\boxed{\sigma'(id_t)(\texttt{iav})[i] = \texttt{true}.}$

Let us perform case analysis on $input(t)$.

- **CASE** $input(t) = \emptyset$:
  By construction, $<\texttt{input\_arcs\_number} \Rightarrow \texttt{1}> \in g_t$ and
  $<\texttt{input\_arcs\_valid(0)} \Rightarrow \texttt{true}> \in i_t$.
  By property of the elaboration and stabilize relations and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\Delta(id_t)(\texttt{ian}) = 1$ and $\sigma'(id_t)(\texttt{iav})[0] = \texttt{true}$.
  Thanks to $\Delta(id_t)(\texttt{ian}) = 1$, we can deduce that $i = 0$.
  Rewriting the goal with $\sigma'(id_t)(\texttt{iav})[0] = \texttt{true}$, tautology.
- **CASE** $input(t) \neq \emptyset$:
  By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in g_t$.
  By property of the elaboration relation and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\Delta(id_t)(\texttt{ian}) = |input(t)|$.
  Thanks to $\Delta(id_t)(\texttt{ian}) = |input(t)|$, we know that $i \in [0, |input(t)| - 1]$.
  By construction, there exist a $p \in input(t)$, $id_p \in Comps(\Delta)$, g_p, i_p, o_p, $j \in [0, |output(p)| - 1]$ and $id_{ji} \in Sigs(\Delta)$ s.t. $\gamma(p) = id_p$ and
  $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$ and $<\texttt{output\_arcs\_valid(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in o_p$ and $<\texttt{input\_arcs\_valid(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in i_t$.
  By property of the stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$ and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma'(id_t)(\texttt{iav})[i] = \sigma'(id_{ji}) = \sigma'(id_p)(\texttt{oav})[j]$.
  Rewriting the goal with $\sigma'(id_t)(\texttt{iav})[i] = \sigma'(id_{ji}) = \sigma'(id_p)(\texttt{oav})[j]$:
  $\boxed{\sigma'(id_p)(\texttt{oav})[j] = \texttt{true}.}$
  By property of the stabilize relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `marking_validation_evaluation` process defined in the place design architecture, we can deduce:

$$\begin{aligned}\sigma'(id_p)(\texttt{oav})[j] = &\big((\sigma'(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma'(id_p)(\texttt{oat})[j] = \texttt{test}) \\ &\quad . \ \sigma'(id_p)(\texttt{sm}) \geq \sigma'(id_p)(\texttt{oaw})[j]\big) \\ &+ \big(\sigma'(id_p)(\texttt{oat})[j] = \texttt{inhib} \ . \ \sigma'(id_p)(\texttt{sm}) < \sigma'(id_p)(\texttt{oaw})[j]\big)\end{aligned} \tag{19}$$

  Rewriting the goal with (19),

$$\boxed{\begin{aligned}\texttt{true} = &\big((\sigma'(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma'(id_p)(\texttt{oat})[j] = \texttt{test}) \\ &\quad . \ \sigma'(id_p)(\texttt{sm}) \geq \sigma'(id_p)(\texttt{oaw})[j]\big) \\ &+ \big(\sigma'(id_p)(\texttt{oat})[j] = \texttt{inhib} \ . \ \sigma'(id_p)(\texttt{sm}) < \sigma'(id_p)(\texttt{oaw})[j]\big)\end{aligned}}$$

  Let us perform case analysis on $pre(p, t)$; there are 3 cases:
  – **CASE** $pre(p, t) = (\omega, \texttt{basic})$:

By construction, $<\texttt{output\_arcs\_types(j)} \Rightarrow \texttt{basic}> \in i_p$ and
$<\texttt{output\_arcs\_weights(j)} \Rightarrow \omega> \in i_p$.
By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$,
we can deduce $\sigma'(id_p)(\texttt{oat})[j] = \texttt{basic}$ and $\sigma'(id_p)(\texttt{oaw})[j] = \omega$.
Rewriting the goal with $\sigma'(id_p)(\texttt{oat})[j] = \texttt{basic}$ and $\sigma'(id_p)(\texttt{oaw})[j] = \omega$, and simplifying the goal:
$\boxed{\sigma'(id_p)(\texttt{sm}) \geq \omega = \texttt{true}.}$
Appealing to Lemma 22, we can deduce $s'.M(p) = \sigma'(id_p)(\texttt{sm})$.
Rewriting the goal with $s'.M(p) = \sigma'(id_p)(\texttt{sm})$: $\boxed{s'.M(p) \geq \omega = \texttt{true}.}$
By definition of $t \in Sens(s'.M)$, $s'.M(p) \geq \omega = \texttt{true}.$

- **CASE** $pre(p,t) = (\omega, \texttt{test})$: same as above.
- **CASE** $pre(p,t) = (\omega, \texttt{inhib})$:
  By construction, $<\texttt{output\_arcs\_types(j)} \Rightarrow \texttt{inhib}> \in i_p$ and
  $<\texttt{output\_arcs\_weights(j)} \Rightarrow \omega> \in i_p$.
  By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma'(id_p)(\texttt{oat})[j] = \texttt{inhib}$ and $\sigma'(id_p)(\texttt{oaw})[j] = \omega$.
  Rewriting the goal with $\sigma'(id_p)(\texttt{oat})[j] = \texttt{inhib}$ and $\sigma'(id_p)(\texttt{oaw})[j] = \omega$, and simplifying the goal: $\boxed{\sigma'(id_p)(\texttt{sm}) < \omega = \texttt{true}.}$
  Appealing to Lemma 22, we can deduce $s'.M(p) = \sigma'(id_p)(\texttt{sm})$.
  Rewriting the goal with $s'.M(p) = \sigma'(id_p)(\texttt{sm})$: $\boxed{s'.M(p) < \omega = \texttt{true}.}$
  By definition of $t \in Sens(s'.M)$, $s'.M(p) < \omega = \texttt{true}.$

2. Assuming that $\sigma'(id_t)(\texttt{s\_enabled}) = \texttt{true}$, let us show $\boxed{t \in Sens(s'.M).}$

   By definition of $t \in Sens(s'.M)$, let us show

   $$\boxed{\forall p \in P, \omega \in \mathbb{N}^*,\ \big(pre(p,t) = (\omega, \texttt{basic}) \vee pre(p,t) = (\omega, \texttt{test}) \Rightarrow s'.M(p) \geq \omega\big) \wedge \big(pre(p,t) = (\omega, \texttt{inhib}) \Rightarrow s'.M(p) < \omega\big)}$$

   Given a $p \in P$ and an $\omega \in \mathbb{N}^*$, let us show
   $\boxed{pre(p,t) = (\omega, \texttt{basic}) \vee pre(p,t) = (\omega, \texttt{test}) \Rightarrow s'.M(p) \geq \omega}$ and
   $\boxed{pre(p,t) = (\omega, \texttt{inhib}) \Rightarrow s'.M(p) < \omega.}$

   (a) Assuming $pre(p,t) = (\omega, \texttt{basic}) \vee pre(p,t) = (\omega, \texttt{test})$, let us show $\boxed{s'.M(p) \geq \omega.}$

   The proceeding is the same for $pre(p,t) = (\omega, \texttt{basic})$ and $pre(p,t) = (\omega, \texttt{test})$. Therefore, we will only cover the case where $pre(p,t) = (\omega, \texttt{basic})$.

   By property of the stabilize relation and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, equation (18) holds.

   Rewriting $\sigma'(id_t)(\texttt{se}) = \texttt{true}$ with (18), we can deduce:
   $\prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{iav})[i] = \texttt{true}.$

Then, we can deduce that $\forall i \in [0, \Delta(id_t)(\texttt{ian}) - 1],\ \sigma'(id_t)(\texttt{iav})[i] = \texttt{true}$.

By construction, there exist an $id_p \in Comps(\Delta)$, g_p, i_p, o_p, $i \in [0, |input(t)| - 1]$, $j \in [0, |output(p)| - 1]$ and $id_{ji} \in Sigs(\Delta)$ s.t. $\gamma(p) = id_p$ and
$\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$ and $<\texttt{output\_arcs\_valid(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in o_p$ and $<\texttt{input\_arcs\_valid(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in i_t$. Let us take such an $id_p \in Comps(\Delta)$, g_p, i_p, o_p, $i \in [0, |input(t)| - 1]$, $j \in [0, |output(p)| - 1]$ and $id_{ji} \in Sigs(\Delta)$.

By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in g_t$.

By property of the elaboration relation and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\Delta(id_t)(\texttt{ian}) = |input(t)|$.

Thanks to $\Delta(id_t)(\texttt{ian}) = |input(t)|$, we can deduce that $\forall i \in [0, |input(t)| - 1]$, $\sigma'(id_t)(\texttt{iav})[i] = \texttt{true}$.

Having such an $i \in [0, |input(t)| - 1]$, we can deduce that $\sigma'(id_t)(\texttt{iav})[i] = \texttt{true}$.

By property of the stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$ and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma'(id_t)(\texttt{iav})[i] = \sigma'(id_{ji}) = \sigma'(id_p)(\texttt{oav})[j]$.

Thanks to $\sigma'(id_t)(\texttt{iav})[i] = \sigma'(id_{ji}) = \sigma'(id_p)(\texttt{oav})[j]$, we have $\sigma'(id_p)(\texttt{oav})[j] = \texttt{true}$.

By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, equation (19) holds. Thanks to (19), we can deduce that:

$$\begin{aligned}\texttt{true} = &\big((\sigma'(id_p)(\texttt{oat})[j] = \texttt{basic} + \sigma'(id_p)(\texttt{oat})[j] = \texttt{test}) \\ &\quad . \ \sigma'(id_p)(\texttt{sm}) \geq \sigma'(id_p)(\texttt{oaw})[j]\big) \\ &+ \big(\sigma'(id_p)(\texttt{oat})[j] = \texttt{inhib} \ . \ \sigma'(id_p)(\texttt{sm}) < \sigma'(id_p)(\texttt{oaw})[j]\big)\end{aligned} \tag{20}$$

By construction, $<\texttt{output\_arcs\_types(j)} \Rightarrow \texttt{basic}> \in i_p$ and $<\texttt{output\_arcs\_weights(j)} \Rightarrow \omega> \in i_p$.

By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma'(id_p)(\texttt{oat})[j] = \texttt{basic}$ and $\sigma'(id_p)(\texttt{oaw})[j] = \omega$.

Thanks to $\sigma'(id_p)(\texttt{oat})[j] = \texttt{basic}$, $\sigma'(id_p)(\texttt{oaw})[j] = \omega$, and simplifying Equation (20), we can deduce $\sigma'(id_p)(\texttt{sm}) \geq \omega = \texttt{true}$.

Appealing to Lemma 22, $s'.M(p) \geq \omega$.

(b) Assuming $pre(p, t) = (\omega, \texttt{inhib})$, let us show $\boxed{s'.M(p) < \omega.}$

The proceeding is the same as in the preceding case. Here, we will start the proof where the two cases are diverging, i.e:

By construction, $<\texttt{output\_arcs\_types(j)} \Rightarrow \texttt{inhib}> \in i_p$ and $<\texttt{output\_arcs\_weights(j)} \Rightarrow \omega> \in i_p$.

By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma'(id_p)(\texttt{oat})[j] = \texttt{inhib}$ and $\sigma'(id_p)(\texttt{oaw})[j] = \omega$.

Thanks to $\sigma'(id_p)(\texttt{oat})[j] = \texttt{inhib}$ and $\sigma'(id_p)(\texttt{oaw})[j] = \omega$, and simplifying Equation (20), we can deduce $\sigma'(id_p)(\texttt{sm}) < \omega = \texttt{true}$.

Appealing to Lemma 22, $s'.M(p) < \omega$.

□

Lemma 30 (Rising edge equal not sensitized)**.** *For all sitpn, b, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Definition 12, then*
$\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \notin Sens(s'.M) \Leftrightarrow \sigma'(id_t)(\texttt{s\_enabled}) = \texttt{false}$.

Proof.

Proving the above lemma is trivial by appealing to Lemma 29 and by reasoning on contrapositives. □

7 Falling edge lock-step simulation

Lemma 31 (Falling edge lock-step simulation)**.** *For all well-defined sitpn $\in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign, \sigma_e \in \Sigma$, $E_p \in \mathbb{N} \to (id \nrightarrow v)$, and $E_c \in \mathbb{N} \to (\mathcal{C} \to \mathbb{B})$ that verify the hypotheses of Definition 9, and for all $\tau \in \mathbb{N}$, $s, s' \in S(sitpn)$, $\sigma, \sigma_\downarrow, \sigma' \in \Sigma$, such that*

- *s and σ are similar states as intended after a rising edge step: $\gamma \vdash s \overset{\downarrow}{\approx} \sigma$*
- *a falling edge step leads from s to s': $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$*
- *a falling edge step leads from σ to σ':*
  $\mathcal{D}_\mathcal{H}, \Delta, \sigma \vdash d.beh \xrightarrow{cs_\downarrow} \sigma_\downarrow$ and $\mathcal{D}_\mathcal{H}, \Delta, \sigma_\downarrow \vdash d.beh \overset{\rightsquigarrow}{\to} \sigma'$

then $\gamma \vdash s' \overset{\downarrow}{\approx} \sigma'$.

Proof.

By definition of the Post falling edge state similarity relation, there are 11 points to prove:

1. $\forall p \in P, id_p \in Comps(\Delta)\ s.t.\ \gamma(p) = id_p,\ \ s'.M(p) = \sigma'(id_p)(\texttt{s\_marking})$.

2. $\forall t \in T_i, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
$\big(u(I_s(t)) = \infty \wedge s'.I(t) \leq l(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)$
$\wedge\big(u(I_s(t)) = \infty \wedge s'.I(t) > l(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)$
$\wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) > u(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)$
$\wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) \leq u(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)$.

3. $\forall c \in \mathcal{C}, id_c \in Ins(\Delta)\ s.t.\ \gamma(c) = id_c,\ \ s'.cond(c) = \sigma'(id_c)$.

4. $\forall a \in \mathcal{A}, id_a \in Outs(\Delta)\ s.t.\ \gamma(a) = id_a,\ \ s'.ex(a) = \sigma'(id_a)$.

5. $\forall f \in \mathcal{F}, id_f \in Outs(\Delta)\ s.t.\ \gamma(f) = id_f,\ \ s'.ex(f) = \sigma'(id_f)$.

6. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
$t \in Firable(s') \Leftrightarrow \sigma'(id_t)(\texttt{s\_firable}) = \texttt{true}$.

7. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
$t \notin Firable(s') \Leftrightarrow \sigma'(id_t)(\texttt{s\_firable}) = \texttt{false}$.

8. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,\ t \in Fired(s') \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$.

9. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,\ t \notin Fired(s') \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{false}$.

10. $\forall p \in P, id_p \in Comps(\Delta)\ s.t.\ \gamma(p) = id_p,$
$\sum_{t \in Fired(s')} pre(p,t) = \sigma'(id_p)(\texttt{s\_output\_token\_sum})$.

11. $\forall p \in P, id_p \in Comps(\Delta)\ s.t.\ \gamma(p) = id_p,$
$\sum_{t \in Fired(s')} post(t,p) = \sigma'(id_p)(\texttt{s\_input\_token\_sum})$.

Each point is proved by a separate lemma:

– Apply the Falling edge equal marking lemma (p. 45) to solve Point 1.

– Apply the Falling edge equal time counters lemma (p. 51) to solve Point 2.

– Apply the Falling edge equal condition values lemma (p. 57) to solve Point 3.

– Apply the Falling edge equal action executions lemma (p. 57) to solve Point 4.

– Apply the Falling edge equal function executions lemma (p. 60) to solve Point 5.

– Apply the Falling edge equal firable lemma (p. 60) to solve Point 6.

– Apply the Falling edge equal not firable lemma (p. 71) to solve Point 7.

– Apply the ?? lemma (p. ??) to solve Point 8. The proof of the ?? lemma is detailled in Section ??.

– Apply the Falling edge equal not fired lemma (p. 86) to solve Point 9.

– Apply the Falling edge equal output token sum lemma (p. 46) to solve Point 10.

– Apply the Falling edge equal input token sum lemma (p. 49) to solve Point 11.

All the lemmas used above, and their corresponding proofs, are to be found in Appendix **??**, Section **??**. □

Definition 13 (Falling edge hypotheses)**.** *Given a sitpn* $\in SITPN$, $b \in P \to \mathbb{N}$, $d \in design$, $\gamma \in WM(sitpn, d)$, $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$, $\Delta \in ElDesign$, $E_p \in \mathbb{N} \to Ins(\Delta) \to value$, $\tau \in \mathbb{N}$, $s, s' \in S(sitpn)$, $\sigma_e, \sigma, \sigma_\downarrow, \sigma' \in \Sigma$, *assume that:*

- *SITPN sitpn is transformed into the* $\mathcal{H}$*-VHDL design* d *and yields the binder* γ*:* $\lfloor sitpn \rfloor_b = (d, \gamma)$
- *Simulation/Execution environments are similar:* $\gamma \vdash E_p \overset{env}{=} E_c$
- Δ *is the elaborated version of design* d*, and* σ_e *is the default design state of* Δ*:* $\mathcal{D}_\mathcal{H}, \emptyset \vdash \mathrm{d} \xrightarrow{elab} \Delta, \sigma_e$
- *Starting states are similar according to the full post rising edge similarity relation:* $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$
- *On the SITPN side, the execution of a falling edge phase starting from state* s *leads to state* s'*:* $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$
- *On the* $\mathcal{H}$*-VHDL side, the simulation of a falling edge phase starting from state* σ *leads to state* σ'*:* $\Delta, \sigma \vdash \mathrm{d.cs} \xrightarrow{\downarrow} \sigma_\downarrow$ *and* $\Delta, \sigma_\downarrow \vdash \mathrm{d.cs} \xrightarrow{\rightsquigarrow} \sigma'$
- *State* σ *is a stable design state:* $\mathcal{D}_\mathcal{H}, \Delta, \sigma \vdash \mathrm{d.cs} \xrightarrow{comb} \sigma$

Lemma 32 (Falling edge equal marking)**.** *For all sitpn,* b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' *that verify the hypotheses of Definition 13, then* $\forall p \in P, id_p \in Comps(\Delta)$ *s.t.* $\gamma(p) = id_p$, $s'.M(p) = \sigma'(id_p)(\texttt{s\_marking})$.

Proof.

Given a $p \in P$ and an $id \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$, let us show
$\boxed{s'.M(p) = \sigma'(id_p)(\texttt{s\_marking}).}$

By definition of $E_c, \tau \vdash sitpn, s \xrightarrow{\downarrow} s'$, we can deduce $s.M(p) = s'.M(p)$.
By property of the $\mathcal{H}$-VHDL falling edge relation, the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `marking` process defined in the `place` design architecture, we can deduce $\sigma'(id_p)(\texttt{s\_marking}) = \sigma(id_p)(\texttt{s\_marking})$.
Rewriting the goal with $s.M(p) = s'.M(p)$ and $\sigma'(id_p)(\texttt{sm}) = \sigma(id_p)(\texttt{sm})$:
$\boxed{s.M(p) = \sigma(id_p)(\texttt{sm}).}$

By definition of $\gamma, E_c, \tau \vdash s \overset{\downarrow}{\approx} \sigma$: $s.M(p) = \sigma(id_p)(\texttt{sm})$.

□

Lemma 33 (Falling edge equal output token sum)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall p, id_p$ s.t. $\gamma(p) = id_p$, $\sum_{t \in Fired(s')} pre(p,t) = \sigma'(id_p)(s\_output\_token\_sum)$.*

Proof.

Given a $p \in P$ and an $id_p \in Comps(\Delta)$, let us show
$\boxed{\sum_{t \in Fired(s')} pre(p,t) = \sigma'(id_p)(\texttt{s\_output\_token\_sum}).}$
By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$. By property of the stabilize relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `output_tokens_sum` process defined in the place design architecture:

$$\sigma'(id_p)(\texttt{sots}) = \sum_{i=0}^{\Delta(id_p)(\texttt{oan})-1} \begin{cases} \sigma'(id_p)(\texttt{oaw})[i] \texttt{ if } (\sigma'(id_p)(\texttt{otf})[i] \\ \qquad\qquad\qquad\qquad . \ \sigma'(id_p)(\texttt{oat})[i] = \texttt{basic}) \\ 0 \ otherwise \end{cases} \tag{21}$$

Rewriting the goal with (21):

$$\boxed{\sum_{t \in Fired(s')} pre(p,t) = \sum_{i=0}^{\Delta(id_p)(\texttt{oan})-1} \begin{cases} \sigma'(id_p)(\texttt{oaw})[i] \texttt{ if } (\sigma'(id_p)(\texttt{otf})[i] \\ \qquad\qquad\qquad\qquad . \ \sigma'(id_p)(\texttt{oat})[i] = \texttt{basic}) \\ 0 \ otherwise \end{cases}}$$

Let us unfold the definition of the left sum term:

$$\boxed{\begin{gathered} \sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } pre(p,t) = (\omega, \texttt{basic}) \\ 0 \ otherwise \end{cases} \\ = \\ \sum_{i=0}^{\Delta(id_p)(\texttt{oan})-1} \begin{cases} \sigma'(id_p)(\texttt{oaw})[i] \texttt{ if } (\sigma'(id_p)(\texttt{otf})[i] \\ \qquad\qquad\qquad\qquad . \ \sigma'(id_p)(\texttt{oat})[i] = \texttt{basic}) \\ 0 \ otherwise \end{cases} \end{gathered}}$$

To ease the reading, let us define functions $f \in Fired(s') \to \mathbb{N}$ and $g \in [0, |output(p)| - 1] \to \mathbb{N}$ s.t.
$f(t) = \begin{cases} \omega \texttt{ if } pre(p,t) = (\omega, \texttt{basic}) \\ 0 \ otherwise \end{cases}$
and $g(i) = \begin{cases} \sigma'(id_p)(\texttt{oaw})[i] \texttt{ if } (\sigma'(id_p)(\texttt{otf})[i] \\ \qquad\qquad\qquad\qquad . \ \sigma'(id_p)(\texttt{oat})[i] = \texttt{basic}) \\ 0 \ otherwise \end{cases}$

Then, the goal is: $\boxed{\sum_{t \in Fired(s')} f(t) = \sum_{i=0}^{\Delta(id_p)(\texttt{oan})-1} g(i)}$

Let us perform case analysis on $output(p)$; there are two cases:

- **CASE** $output(p) = \emptyset$:

By construction, <output_arcs_number⇒ 1> $\in g_p$, <output_arcs_types(0)⇒ basic> $\in i_p$, <output_transitions_fired(0)⇒ true> $\in i_p$, and <output_arcs_weights(0)⇒ 0> $\in i_p$.

By property of the elaboration relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\Delta(id_p)(\texttt{oan}) = 1$.

By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma'(id_p)(\texttt{oat})[0] = \texttt{basic}$, $\sigma'(id_p)(\texttt{otf})[0] = \texttt{true}$ and $\sigma'(id_p)(\texttt{oaw})[0] = 0$.

By property of $output(p) = \emptyset$, we can deduce

$$\sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } pre(p,t) = (\omega, \texttt{basic}) \\ 0 \textit{ otherwise} \end{cases} = 0$$

Rewriting the goal with $\Delta(id_p)(\texttt{oan}) = 1$, $\sigma'(id_p)(\texttt{oat})[0] = \texttt{basic}$, $\sigma'(id_p)(\texttt{otf})[0] = \texttt{true}$, $\sigma'(id_p)(\texttt{oaw})[0] = 0$ and $\sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } pre(p,t) = (\omega, \texttt{basic}) \\ 0 \textit{ otherwise} \end{cases} = 0$, tautology.

- **CASE** $output(p) \neq \emptyset$:

By construction, $<\texttt{oan} \Rightarrow |output(p)|> \in g_p$, and by property of the elaboration relation, we can deduce $\Delta(id_p)(\texttt{oan}) = |output(p)|$.

Rewriting the goal with $\Delta(id_p)(\texttt{oan}) = |output(p)|$: $\boxed{\sum_{t \in Fired(s')} f(t) = \sum_{i=0}^{|output(p)|-1} g(i).}$

There exists a mapping, given by the transformation function, between the set $output(p)$ and $[0, |output(p)| - 1]$.

Let $\beta \in output(p) \to [0, |output(p)| - 1]$ be that mapping.

To prove the current goal, it suffices to show that, for all $t \in Fired(s')$, if $t \in output(p)$ then $f(t) = g(\beta(t))$, and $f(t) = 0$ otherwise.

Given a $t \in Fired(s')$, there are two points to prove:

1. Assuming that $t \in output(p)$, show $f(t) = g(\beta(t))$.
2. Assuming that $t \notin output(p)$, show $f(t) = 0$.

1. Assuming that $t \in output(p)$, let us show $\boxed{f(t) = g(\beta(t)).}$
   Replacing the terms $f(t)$ and $g(\beta(t))$ by their full definition, let us show

$$\boxed{\begin{cases} \omega \texttt{ if } pre(p,t) = (\omega, \texttt{basic}) \\ 0 \textit{ otherwise} \end{cases} = \begin{cases} \sigma'(id_p)(\texttt{oaw})[\beta(t)] \texttt{ if } (\sigma'(id_p)(\texttt{otf})[\beta(t)] \\ \qquad\qquad\qquad\qquad . \ \sigma'(id_p)(\texttt{oat})[\beta(t)] = \texttt{basic}) \\ 0 \textit{ otherwise} \end{cases}}$$

As $t \in output(p)$, there exist a weight $\omega \in \mathbb{N}$ and an arc type $a \in \{\texttt{basic}, \texttt{test}, \texttt{inhib}\}$ such that $pre(p,t) = (\omega, a)$.
By construction, we have:
– $<\texttt{oat}(\beta(t)) \Rightarrow a> \in i_p$
– $<\texttt{oaw}(\beta(t)) \Rightarrow \omega> \in i_p$
By property of the stabilize relation and $<\texttt{oat}(\beta(t)) \Rightarrow a> \in i_p$, we have $\sigma'(id_p)(\texttt{oat})[\beta(t)] = a$.
Let us perform case analysis of the value of a; there are two cases:

– **CASE** $a = \texttt{inhib}$ or $a = \texttt{test}$:
In that case, $pre(p,t) \neq (\omega, \texttt{basic})$ and $\sigma'(id_p)(\texttt{oat})[\beta(t)] \neq \texttt{basic}$.
Thus, the goal can be rewritten as follows: $\boxed{0 = 0}$, tautology.

– **CASE** $a = \texttt{basic}$:
In that case, $pre(p,t) = (\omega, \texttt{basic})$ and $\sigma'(id_p)(\texttt{oat})[\beta(t)] = \texttt{basic}$.
Thus, the goal can be rewritten as follows:

$$\boxed{\omega = \begin{cases} \sigma'(id_p)(\texttt{oaw})[\beta(t)] \texttt{ if } \sigma'(id_p)(\texttt{otf})[\beta(t)] \\ 0 \textit{ otherwise} \end{cases}}$$

By property of the stabilize relation and $<\texttt{oaw}(\beta(t)) \Rightarrow \omega> \in i_p$, we have $\sigma'(id_p)(\texttt{oaw})[\beta(t)] = \omega$.
Thus, the goal can be rewritten as follows:

$$\boxed{\omega = \begin{cases} \omega \texttt{ if } \sigma'(id_p)(\texttt{otf})[\beta(t)] \\ 0 \textit{ otherwise} \end{cases}}$$

By construction, there exists an $id_{ft} \in Sigs(\Delta)$ such that:
$*$ $<\texttt{fired} \Rightarrow id_{ft}> \in o_t$
$*$ $<\texttt{otf}(\beta(t)) \Rightarrow \texttt{id}_{\texttt{ft}}> \in i_p$
Let us take an $id_{ft} \in Sigs(\Delta)$ that verifies the above properties.
By property of the stabilize relation, $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}}> \in o_t$ and $<\texttt{otf}(\beta(t)) \Rightarrow \texttt{id}_{\texttt{ft}}> \in i_p$, we can deduce $\sigma'(id_p)(\texttt{otf})[\beta(t)] = \sigma'(id_{ft}) = \sigma'(id_t)(\texttt{fired})$.
Thus, the goal can be rewritten as follows:

$$\boxed{\omega = \begin{cases} \omega \texttt{ if } \sigma'(id_t)(\texttt{fired}) \\ 0 \textit{ otherwise} \end{cases}}$$

Appealing to Lemma **??**, from $t \in Fired(s')$, we can deduce $\sigma'(id_t)(\texttt{fired}) = \texttt{true}$.
Thus, the goal can be rewritten as follows: $\boxed{\omega = \omega}$, tautology.

2. Assuming that $t \notin output(p)$, let us show $\boxed{f(t) = 0.}$
Replacing the term $f(t)$ by its full definition, let us show

$$\boxed{\begin{cases} \omega \texttt{ if } pre(p,t) = (\omega, \texttt{basic}) \\ 0 \textit{ otherwise} \end{cases} = 0}$$

As $t \notin output(p)$, then $pre(p,t) \neq (\omega, \texttt{basic})$, and we can rewrite the goal as follows: $\boxed{0 = 0}$, tautology.

□

Lemma 34 (Falling edge equal input token sum)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then* $\forall p, id_p$ *s.t.* $\gamma(p) = id_p$, $\sum_{t \in Fired(s')} post(t,p) = \sigma'_p(s\_input\_token\_sum)$.

Proof.

Given a $p \in P$ and an $id_p \in Comps(\Delta)$, let us show

$$\boxed{\sum_{t \in Fired(s')} post(t,p) = \sigma'(id_p)(\texttt{s\_input\_token\_sum}).}$$

By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$. By property of the stabilize relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `input_tokens_sum` process defined in the place design architecture:

$$\sigma'(id_p)(\texttt{sits}) = \sum_{i=0}^{\Delta(id_p)(\texttt{ian})-1} \begin{cases} \sigma'(id_p)(\texttt{iaw})[i] \texttt{ if } \sigma'(id_p)(\texttt{itf})[i] \\ 0 \textit{ otherwise} \end{cases} \tag{22}$$

Rewriting the goal with (22):

$$\boxed{\sum_{t \in Fired(s')} post(t,p) = \sum_{i=0}^{\Delta(id_p)(\texttt{ian})-1} \begin{cases} \sigma'(id_p)(\texttt{iaw})[i] \texttt{ if } \sigma'(id_p)(\texttt{otf})[i] \\ 0 \textit{ otherwise} \end{cases}}$$

Let us unfold the definition of the left sum term:

$$\boxed{\begin{gathered} \sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } post(t,p) = \omega \\ 0 \textit{ otherwise} \end{cases} \\ = \\ \sum_{i=0}^{\Delta(id_p)(\texttt{ian})-1} \begin{cases} \sigma'(id_p)(\texttt{iaw})[i] \texttt{ if } \sigma'(id_p)(\texttt{itf})[i] \\ 0 \textit{ otherwise} \end{cases} \end{gathered}}$$

Let us perform case analysis on $input(p)$; there are two cases:

- **CASE** $input(p) = \emptyset$:

  By construction, $<\texttt{input\_arcs\_number} \Rightarrow \texttt{1}> \in g_p$,
  $<\texttt{input\_transitions\_fired(0)} \Rightarrow \texttt{true}> \in i_p$, and
  $<\texttt{input\_arcs\_weights(0)} \Rightarrow \texttt{0}> \in i_p$.

  By property of the elaboration relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\Delta(id_p)(\texttt{ian}) = 1$.

  By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma'(id_p)(\texttt{itf})[0] = \texttt{true}$ and $\sigma'(id_p)(\texttt{iaw})[0] = 0$.

By property of $input(p) = \emptyset$, we can deduce $\sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } post(t,p) = \omega \\ 0 \ otherwise \end{cases} = 0$.

Rewriting the goal with $\Delta(id_p)(\texttt{ian}) = 1$, $\sigma'(id_p)(\texttt{itf})[0] = \texttt{true}$, $\sigma'(id_p)(\texttt{iaw})[0] = 0$, and $\sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } post(t,p) = \omega \\ 0 \ otherwise \end{cases} = 0$, and simplifying the goal: tautology.

- **CASE** $input(p) \neq \emptyset$:

  By construction, $<\texttt{ian} \Rightarrow |input(p)|> \in g_p$, and by property of the elaboration relation, we can deduce $\Delta(id_p)(\texttt{ian}) = |input(p)|$.

  To ease the reading, let us define functions $f \in Fired(s') \to \mathbb{N}$ and $g \in [0, |input(p)| - 1] \to \mathbb{N}$ s.t. $f(t) = \begin{cases} \omega \texttt{ if } post(t,p) = \omega \\ 0 \ otherwise \end{cases}$ and $g(i) = \begin{cases} \sigma'(id_p)(\texttt{iaw})[i] \texttt{ if } \sigma'(id_p)(\texttt{itf})[i] \\ 0 \ otherwise \end{cases}$

  Then, the goal is: $\boxed{\sum_{t \in Fired(s')} f(t) = \sum_{i=0}^{\Delta(id_p)(\texttt{ian})-1} g(i)}$

  Rewriting the goal with $\Delta(id_p)(\texttt{ian}) = |input(p)|$: $\boxed{\sum_{t \in Fired(s')} f(t) = \sum_{i=0}^{|input(p)|-1} g(i).}$

  There exists a mapping, given by the transformation function, between the set $input(p)$ and $[0, |input(p)| - 1]$.

  Let $\beta \in input(p) \to [0, |input(p)| - 1]$ be that mapping.

  To prove the current goal, it suffices to show that, for all $t \in Fired(s')$, if $t \in input(p)$ then $f(t) = g(\beta(t))$, and $f(t) = 0$ otherwise.

  Given a $t \in Fired(s')$, there are two points to prove:

  1. Assuming that $t \in input(p)$, show $f(t) = g(\beta(t))$.
  2. Assuming that $t \notin input(p)$, show $f(t) = 0$.

  1. Assuming that $t \in input(p)$, let us show $\boxed{f(t) = g(\beta(t)).}$

     Replacing the terms $f(t)$ and $g(\beta(t))$ by their full definition, let us show

     $$\boxed{\begin{array}{c} \begin{cases} \omega \texttt{ if } post(t,p) = \omega \\ 0 \ otherwise \end{cases} \\ = \\ \begin{cases} \sigma'(id_p)(\texttt{iaw})[\beta(t)] \texttt{ if } \sigma'(id_p)(\texttt{itf})[\beta(t)] \\ 0 \ otherwise \end{cases} \end{array}}$$

     As $t \in input(p)$, there exist a weight $\omega \in \mathbb{N}^*$ such that $post(t,p) = \omega$. Let us take such an ω. Thus, the goal can be rewritten as follows:

$$\omega = \begin{cases} \sigma'(id_p)(\texttt{iaw})[\beta(t)] \texttt{ if } \sigma'(id_p)(\texttt{itf})[\beta(t)] \\ 0 \ otherwise \end{cases}$$

By construction, we have $<\texttt{iaw}(\beta(t)) \Rightarrow \omega> \in i_p$, and by property of the stabilize relation, we can deduce $\sigma'(id_p)(\texttt{iaw})[\beta(t)] = \omega$. Thus, the goal can be rewritten as follows:

$$\omega = \begin{cases} \omega \texttt{ if } \sigma'(id_p)(\texttt{itf})[\beta(t)] \\ 0 \ otherwise \end{cases}$$

By construction, there exists an $id_{ft} \in Sigs(\Delta)$ such that:

– $<\texttt{fired} \Rightarrow id_{ft}> \in o_t$

– $<\texttt{itf}(\beta(t)) \Rightarrow \texttt{id}_{\texttt{ft}}> \in i_p$

Let us take an $id_{ft} \in Sigs(\Delta)$ that verifies the above properties.

By property of the stabilize relation, $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}}> \in o_t$ and $<\texttt{itf}(\beta(t)) \Rightarrow \texttt{id}_{\texttt{ft}}> \in i_p$, we can deduce $\sigma'(id_p)(\texttt{itf})[\beta(t)] = \sigma'(id_{ft}) = \sigma'(id_t)(\texttt{fired})$.

Thus, the goal can be rewritten as follows:

$$\omega = \begin{cases} \omega \texttt{ if } \sigma'(id_t)(\texttt{fired}) \\ 0 \ otherwise \end{cases}$$

Appealing to Lemma **??**, from $t \in Fired(s')$, we can deduce $\sigma'(id_t)(\texttt{fired}) = \texttt{true}$.

Thus, the goal can be rewritten as follows: $\omega = \omega$, tautology.

2. Assuming that $t \notin input(p)$, let us show $f(t) = 0$.

   Replacing the term $f(t)$ by its full definition, let us show

$$\begin{cases} \omega \texttt{ if } post(t,p) = \omega \\ 0 \ otherwise \end{cases} = 0$$

   As $t \notin output(p)$, then $post(t,p) \neq \omega$, and we can rewrite the goal as follows: $0 = 0$, tautology.

□

7.1 Falling edge and time counters

Lemma 35 (Falling edge equal time counters)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then* $\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,
$\big(u(I_s(t)) = \infty \wedge s'.I(t) \leq l(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)$
$\wedge\big(u(I_s(t)) = \infty \wedge s'.I(t) > l(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)$
$\wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) > u(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)$
$\wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) \leq u(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)$.

Proof.

Given a $t \in T_i$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show

$$\boxed{\begin{array}{l}\big(u(I_s(t)) = \infty \wedge s'.I(t) \leq l(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)\\ \wedge\big(u(I_s(t)) = \infty \wedge s'.I(t) > l(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = l(I_s(t))\big)\\ \wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) > u(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = u(I_s(t))\big)\\ \wedge\big(u(I_s(t)) \neq \infty \wedge s'.I(t) \leq u(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})\big)\end{array}}$$

By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$. By property of the elaboration, $\mathcal{H}$-VHDL rising edge and stabilize relations, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `time_counter` process defined in the transition design architecture, we can deduce:

$$\begin{aligned}\sigma(id_t)(\texttt{se}) = \texttt{true} \wedge \Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMPORAL} \wedge \sigma(id_t)(\texttt{srtc}) = \texttt{false}\\ \wedge\sigma(id_t)(\texttt{stc}) < \Delta(id_t)(\texttt{mtc}) \Rightarrow \sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc}) + 1\end{aligned} \tag{23}$$

$$\begin{aligned}\sigma(id_t)(\texttt{se}) = \texttt{true} \wedge \Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMPORAL} \wedge \sigma(id_t)(\texttt{srtc}) = \texttt{false}\\ \wedge\sigma(id_t)(\texttt{stc}) \geq \Delta(id_t)(\texttt{mtc}) \Rightarrow \sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})\end{aligned} \tag{24}$$

$$\begin{aligned}\sigma(id_t)(\texttt{se}) = \texttt{true} \wedge \Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMPORAL}\\ \wedge\sigma(id_t)(\texttt{srtc}) = \texttt{true} \Rightarrow \sigma'(id_t)(\texttt{stc}) = 1\end{aligned} \tag{25}$$

$$\sigma(id_t)(\texttt{se}) = \texttt{false} \vee \Delta(id_t)(\texttt{tt}) = \texttt{NOT\_TEMPORAL} \Rightarrow \sigma'(id_t)(\texttt{stc}) = 0 \tag{26}$$

Then, there are 4 points to show:

1. $\boxed{u(I_s(t)) = \infty \wedge s'.I(t) \leq l(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter})}$

   Assuming $u(I_s(t)) = \infty$ and $s'.I(t) \leq l(I_s(t))$, let us show
   $\boxed{s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter}).}$

   Let us perform case analysis on $t \in Sens(s.M)$; there are two cases:

   (a) **CASE** $t \notin Sens(s.M)$:

   By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we can deduce $\sigma(id_t)(\texttt{se}) = \texttt{false}$.
   Appealing to (26) and $\sigma(id_t)(\texttt{se}) = \texttt{false}$, we can deduce $\sigma'(id_t)(\texttt{stc}) = 0$.
   By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we can deduce $s'.I(t) = 0$.
   Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = 0$ and $s'.I(t) = 0$: tautology.

   (b) **CASE** $t \in Sens(s.M)$:

   By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we can deduce $\sigma(id_t)(\texttt{se}) = \texttt{true}$.
   By construction, and as $u(I_s(t)) = \infty$, we have $<\texttt{tt} \Rightarrow \texttt{TEMP\_A\_INF}> \in g_t$. By property of the elaboration relation, we have $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_INF}$.
   Let us perform case analysis on $s.reset_t(t)$; there are two cases:

i. **CASE** $s.reset_t(t) = \texttt{true}$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, $\sigma(id_t)(\texttt{srtc}) = \texttt{true}$.
Appealing to (25), $\sigma(id_t)(\texttt{se}) = \texttt{true}$, $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_INF}$ and $\sigma(id_t)(\texttt{srtc}) = \texttt{true}$, we can deduce $\sigma'(id_t)(\texttt{stc}) = 1$.
By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**), we can deduce $s'.I(t) = 1$.
Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = 1$ and $s'.I(t) = 1$: tautology.

ii. **CASE** $s.reset_t(t) = \texttt{false}$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$.
As $u(I_s(t)) = \infty$, there exists an $a \in \mathbb{N}^*$ s.t. $I_s(t) = [a, \infty]$. Let us take such an $a \in \mathbb{N}^*$.
By construction, <$\texttt{maximal\_time\_counter} \Rightarrow a$> $\in g_t$, and by property of the elaboration relation, we have $\Delta(id_t)(\texttt{mtc}) = a$.

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**), and knowing that $t \in Sens(s.M)$, $s.reset_t(t) = \texttt{false}$ and $u(I_s(t)) = \infty$, we can deduce $s'.I(t) = s.I(t) + 1$.
Rewriting the goal with $s'.I(t) = s.I(t) + 1$: $\boxed{s.I(t) + 1 = \sigma'(id_t)(\texttt{stc}).}$
We assumed that $s'.I(t) \leq l(I_s(t))$, and as $s'.I(t) = s.I(t) + 1$, then $s.I(t) + 1 \leq l(I_s(t))$, then $s.I(t) < l(I_s(t))$, then $s.I(t) < a$ since $a = l(I_s(t))$.

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, and knowing that $s.I(t) < l(I_s(t))$ and $u(I_s(t)) = \infty$, we can deduce $s.I(t) = \sigma(id_t)(\texttt{stc})$.
Appealing to $\Delta(id_t)(\texttt{mtc}) = a$, $s.I(t) = \sigma(id_t)(\texttt{stc})$ and $s.I(t) < a$, we can deduce $\sigma(id_t)(\texttt{stc}) < \Delta(id_t)(\texttt{mtc})$.
Appealing to (23), $\sigma(id_t)(\texttt{stc}) < \Delta(id_t)(\texttt{mtc})$, $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$ and $\sigma(id_t)(\texttt{se}) = \texttt{true}$, we can deduce: $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc}) + 1$.
Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc}) + 1$ and $s.I(t) = \sigma(id_t)(\texttt{stc})$: tautology.

2. $\boxed{u(I_s(t)) = \infty \wedge s'.I(t) > l(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = l(I_s(t)).}$

Assuming that $u(I_s(t)) = \infty$ and $s'.I(t) > l(I_s(t))$, let us show $\boxed{\sigma'(id_t)(\texttt{s\_time\_counter}) = l(I_s(t)).}$

As $u(I_s(t)) = \infty$, there exists an $a \in \mathbb{N}^*$ s.t. $I_s(t) = [a, \infty]$. Let us take such an $a \in \mathbb{N}^*$.

By construction, <$\texttt{maximal\_time\_counter} \Rightarrow a$> $\in g_t$, and <$\texttt{transition\_type} \Rightarrow$ TEMP_A_INF> $\in g_t$ by property of the elaboration relation, we can deduce $\Delta(id_t)(\texttt{mtc}) = a$ and $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_INF}$.

Let us perform case analysis on $t \in Sens(s.M)$:

(a) **CASE** $t \notin Sens(s.M)$:

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), and knowing that $t \in Sens(s.M)$, we can deduce $s'.I(t) = 0$. Since $l(I_s(t)) \in \mathbb{N}^*$, then $l(I_s(t)) > 0$.
Contradicts $s'.I(t) > l(I_s(t))$.

(b) **CASE** $t \in Sens(s.M)$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$ and $t \in Sens(s.M)$, we can deduce $\sigma(id_t)(\texttt{se}) = \texttt{true}$.
Let us perform case analysis on $s.reset_t(t)$; there are two cases:

i. **CASE** $s.reset_t(t) = \texttt{true}$:

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$: $s'.I(t) = 1$.
We assumed that $s'.I(t) > l(I_s(t))$, then $1 > l(I_s(t))$.
Contradicts $l(I_s(t)) > 0$.

ii. **CASE** $s.reset_t(t) = \texttt{false}$:

By property of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$ and $s.reset_t(t) = \texttt{false}$, we can deduce $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$.
By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), and knowing that $s'.I(t) > l(I_s(t))$, we can deduce

$$\begin{aligned} s'.I(t) = s.I(t) + 1 &\Rightarrow s.I(t) + 1 > l(I_s(t)) \\ &\Rightarrow s.I(t) \geq l(I_s(t)) \end{aligned}$$

Let us perform case analysis on $s.I(t) \geq l(I_s(t))$:

A. **CASE** $s.I(t) > l(I_s(t))$: $\boxed{\sigma'(id_t)(\texttt{stc}) = l(I_s(t)).}$

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we can deduce $\sigma(id_t)(\texttt{stc}) = l(I_s(t))$.
Appealing to (24), we can deduce $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})$.
Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})$ and $\sigma(id_t)(\texttt{stc}) = l(I_s(t))$: tautology.

B. **CASE** $s.I(t) = l(I_s(t))$: $\boxed{\sigma'(id_t)(\texttt{stc}) = l(I_s(t)).}$

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we can deduce $s.I(t) = \sigma(id_t)(\texttt{stc})$.
Appealing to (24), we can deduce $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})$.
Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})$, $s.I(t) = \sigma(id_t)(\texttt{stc})$ and $s.I(t) = l(I_s(t))$: tautology.

3. $\boxed{u(I_s(t)) \neq \infty \wedge s'.I(t) > u(I_s(t)) \Rightarrow \sigma'(id_t)(\texttt{s\_time\_counter}) = u(I_s(t)).}$

Assuming that $u(I_s(t)) \neq \infty$ and $s'.I(t) > u(I_s(t))$, let us show
$\boxed{\sigma'(id_t)(\texttt{s\_time\_counter}) = u(I_s(t)).}$

As $u(I_s(t)) \neq \infty$, there exists an $a \in \mathbb{N}^*$, and a $b \in \mathbb{N}^*$ s.t. $I_s(t) = [a, b]$. Let us take such an a and b.

By construction, $<\texttt{maximal\_time\_counter} \Rightarrow b> \in g_t$ and there exists
$tt \in \{\texttt{TEMP\_A\_A}, \texttt{TEMP\_A\_B}\}$ s.t. $<\texttt{transition\_type} \Rightarrow tt> \in g_t$.

By property of the elaboration relation and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, we can deduce $\Delta(id_t)(\texttt{mtc}) = b = u(I_s(t))$ and $\Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMP}$.

Let us perform case analysis on $t \in Sens(s.M)$:

(a) **CASE** $t \notin Sens(s.M)$:

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), and knowing that $t \in Sens(s.M)$, then $s'.I(t) = 0$.
Since $u(I_s(t)) \in \mathbb{N}^*$, then $u(I_s(t)) > 0$.
Contradicts $s'.I(t) > u(I_s(t))$.

(b) **CASE** $t \in Sens(s.M)$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$ and $t \in Sens(s.M)$, we can deduce $\sigma(id_t)(\texttt{se}) = \texttt{true}$.
Let us perform case analysis on $s.reset_t(t)$; there are two cases:

i. **CASE** $s.reset_t(t) = \texttt{true}$:

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we can deduce $s'.I(t) = 1$.
We assumed that $s'.I(t) > u(I_s(t))$, then we can deduce $1 > u(I_s(t))$.
Contradicts $u(I_s(t)) > 0$.

ii. **CASE** $s.reset_t(t) = \texttt{false}$:

By property of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$ and $s.reset_t(t) = \texttt{false}$, we can deduce $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$.
Let us perform case analysis on $s.I(t) > u(I_s(t))$ or $s.I(t) \leq u(I_s(t))$:

A. **CASE** $s.I(t) > u(I_s(t))$: $\boxed{\sigma'(id_t)(\texttt{stc}) = u(I_s(t)).}$

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**), we can deduce $s'.I(t) = s.I(t)$.
By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we can deduce $\sigma(id_t)(\texttt{stc}) = u(I_s(t))$.
Appealing to (24), we have $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})$.
Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})$ and $\sigma(id_t)(\texttt{stc}) = u(I_s(t))$: tautology.

B. **CASE** $s.I(t) \leq u(I_s(t))$: $\boxed{\sigma'(id_t)(\texttt{stc}) = u(I_s(t)).}$

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we can deduce $s.I(t) = \sigma(id_t)(\texttt{stc})$.
Let us perform case analysis on $s.I(t) \leq u(I_s(t))$; there are two cases:

- **CASE** $s.I(t) = u(I_s(t))$:

  Appealing to $\Delta(id_t)(\texttt{mtc}) = b = u(I_s(t))$, $s.I(t) = \sigma(id_t)(\texttt{stc})$ and $s.I(t) = u(I_s(t))$, we can deduce $\Delta(id_t)(\texttt{mtc}) \leq \sigma(id_t)(\texttt{stc})$.
  Appealing to $\Delta(id_t)(\texttt{mtc}) \leq \sigma(id_t)(\texttt{stc})$ and (24), we can deduce $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})$.
  Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc})$, $s.I(t) = \sigma(id_t)(\texttt{stc})$ and $s.I(t) = u(I_s(t))$: tautology.

- **CASE** $s.I(t) < u(I_s(t))$:

  By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**), we can deduce $s'.I(t) = s.I(t) + 1$.
  From $s'.I(t) = s.I(t) + 1$ and $s.I(t) < u(I_s(t))$, we can deduce $s'.I(t) \leq u(I_s(t))$; contradicts $s'.I(t) > u(I_s(t))$.

4. $\boxed{u(I_s(t)) \neq \infty \wedge s'.I(t) \leq u(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter}).}$

Assuming that $u(I_s(t)) \neq \infty$ and $s'.I(t) \leq u(I_s(t))$, let us show
$\boxed{s'.I(t) = \sigma'(id_t)(\texttt{s\_time\_counter}).}$

As $u(I_s(t)) \neq \infty$, there exists an $a \in \mathbb{N}^*$, and a $b \in \mathbb{N}^*$ s.t. $I_s(t) = [a, b]$. Let us take such an a and b.

By construction, $<\texttt{maximal\_time\_counter} \Rightarrow b> \in g_t$ and there exists $tt \in \{\texttt{TEMP\_A\_A,TEMP\_A\_B}\}$ s.t. $<\texttt{transition\_type} \Rightarrow tt> \in g_t$; by property of the elaboration relation, we can deduce $\Delta(id_t)(\texttt{mtc}) = b = u(I_s(t))$ and $\Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMP}$.

Let us perform case analysis on $t \in Sens(s.M)$:

(a) **CASE** $t \notin Sens(s.M)$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $\sigma(id_t)(\texttt{se}) = \texttt{false}$.
Appealing (26) and $\sigma(id_t)(\texttt{se}) = \texttt{false}$, we have $\sigma'(id_t)(\texttt{stc}) = 0$.
By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**), we have $s'.I(t) = 0$.
Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = 0$ and $s'.I(t) = 0$: tautology.

(b) **CASE** $t \in Sens(s.M)$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $\sigma(id_t)(\texttt{se}) = \texttt{true}$.
Let us perform case analysis on $s.reset_t(t)$:

i. **CASE** $s.reset_t(t) = \texttt{true}$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $\sigma(id_t)(\texttt{srtc}) = \texttt{true}$.
Appealing to (25), $\Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMP}$, $\sigma(id_t)(\texttt{se}) = \texttt{true}$ and $\sigma(id_t)(\texttt{srtc}) = \texttt{true}$, we have $\sigma'(id_t)(\texttt{stc}) = 1$.
By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**), we have $s'.I(t) = 1$.
Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = 1$ and $s'.I(t) = 1$, tautology.

ii. **CASE** $s.reset_t(t) = \texttt{false}$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$.
Let us perform case analysis on $s.I(t) > u(I_s(t))$ or $s.I(t) \leq u(I_s(t))$:

A. **CASE** $s.I(t) > u(I_s(t))$:

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s.I(t) = s'.I(t)$, and thus, $s'.I(t) > u(I_s(t))$. Contradicts $s'.I(t) \leq u(I_s(t))$.

B. **CASE** $s.I(t) \leq u(I_s(t))$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $s.I(t) = \sigma(id_t)(\texttt{stc})$.

- **CASE** $s.I(t) < u(I_s(t))$:
  From $s.I(t) < u(I_s(t))$, $s.I(t) = \sigma(id_t)(\texttt{stc})$ and
  $\Delta(id_t)(\texttt{mtc}) = b = u(I_s(t))$, we can deduce $\sigma(id_t)(\texttt{stc}) < \Delta(id_t)(\texttt{mtc})$.
  From (23), $\sigma(id_t)(\texttt{se}) = \texttt{true}$, $\Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMP}$, $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$ and $\sigma(id_t)(\texttt{stc}) < \Delta(id_t)(\texttt{mtc})$, we can deduce
  $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc}) + 1$.
  By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**), we can deduce $s'.I(t) = s.I(t) + 1$.
  Rewriting the goal with $\sigma'(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{stc}) + 1$ and $s'.I(t) = s.I(t) + 1$, tautology.
- **CASE** $s.I(t) = u(I_s(t))$:
  By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**), we know that $s'.I(t) = s.I(t)+1$. We assumed that $s'.I(t) \leq u(I_s(t))$; thus, $s.I(t) + 1 \leq u(I_s(t))$.

Contradicts $s.I(t) = u(I_s(t))$.

□

7.2 Falling edge and condition values

Lemma 36 (Falling edge equal condition values)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall c \in \mathcal{C}, id_c \in Ins(\Delta)$ s.t. $\gamma(c) = id_c$, $s'.cond(c) = \sigma'(id_c)$.*

Proof.

Given a $c \in \mathcal{C}$ and an $id_c \in Ins(\Delta)$ s.t. $\gamma(c) = id_c$, let us show $\boxed{s'.cond(c) = \sigma'(id_c).}$

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**), we have $s'.cond(c) = E_c(\tau, c)$.

By definition of $\gamma, E_c, \tau \vdash s \overset{\downarrow}{\approx} \sigma$, we have $\sigma(id_c) = E_c(\tau, c)$

By property of the $\mathcal{H}$-VHDL falling edge, the stabilize relations and $id_c \in Ins(\Delta)$, we have $\sigma'(id_c) = \sigma(id_c) = E_c(\tau, c)$.

Rewriting the goal with $s'.cond(c) = E_c(\tau, c)$ and $\sigma'(id_c) = E_c(\tau, c)$, tautology.

□

7.3 Falling edge and action executions

Lemma 37 (Falling edge equal action executions)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall a \in \mathcal{A}, id_a \in Outs(\Delta)$ s.t. $\gamma(a) = id_a$, $s'.ex(a) = \sigma'(id_a)$.*

Proof.

Given an $a \in \mathcal{A}$ and an $id_a \in Outs(\Delta)$ s.t. $\gamma(a) = id_a$, let us show $\boxed{s'.ex(a) = \sigma'(id_a).}$

By property of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**):

$$s'.ex(a) = \sum_{p \in marked(s.M)} \mathbb{A}(p, a) \tag{27}$$

By construction, the generated `action` process is a part of design d's behavior, i.e. there exist an $sl \subseteq Sigs(\Delta)$ and an $ss_a \in ss$ s.t. $\texttt{ps}(\texttt{action}, \emptyset, sl, ss) \in d.cs$.

By construction id_a is only assigned in the body of the `action` process during the initialization or a falling edge phase.

Let $pls(a)$ be the set of actions associated to action a, i.e. $pls(a) = \{p \in P \mid \mathbb{A}(p, a) = true\}$. Then, depending on $pls(a)$, there are two cases of assignment of output port id_a:

- **CASE** $pls(a) = \emptyset$:

By construction, $\texttt{id}_\texttt{a} \Leftarrow \texttt{false} \in ss_{a\downarrow}$ where $ss_{a\downarrow}$ is the part of the "`action`" process body executed during a falling edge phase.

By property of the $\mathcal{H}$-VHDL falling edge relation, the stabilize relation and $\texttt{ps}(\texttt{action}, \emptyset, sl, ss_a) \in d.cs$, we can deduce $\sigma'(id_a) = false$.

By property of $\sum_{p \in marked(s.M)} \mathbb{A}(p,a)$ and $pls(a) = \emptyset$, we can deduce $\sum_{p \in marked(s.M)} \mathbb{A}(p,a) = \texttt{false}$.

Rewriting the goal with (27), $\sigma'(id_a) = false$ and $\sum_{p \in marked(s.M)} \mathbb{A}(p,a) = \texttt{false}$, tautology.

- **CASE** $pls(a) \neq \emptyset$:
  By construction, $\texttt{id}_\texttt{a} \Leftarrow \texttt{id}_{\texttt{mp}_0} + \cdots + \texttt{id}_{\texttt{mp}_n} \in ss_{a\downarrow}$, where $id_{mp_i} \in Sigs(\Delta)$, $ss_{a\downarrow}$ is the part of the `action` process body executed during the falling edge phase, and $n = |pls(a)| - 1$.

  By property of the $\mathcal{H}$-VHDL falling edge relation, the stabilize relation, and $\texttt{ps}(\texttt{action}, \emptyset, sl, ss) \in d.cs$:

  $$\sigma'(id_a) = \sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}) \tag{28}$$

  Rewriting the goal with (27) and (28):
  $\boxed{\sum_{p \in marked(s.M)} \mathbb{A}(p,a) = \sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}).}$

  Let us reason on the value of $\sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n})$; there are two cases:

  – **CASE** $\sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}) = \texttt{true}$:
    Then, we can rewrite the goal as follows: $\boxed{\sum_{p \in marked(s.M)} \mathbb{A}(p,a) = \texttt{true}.}$

    To prove the above goal, let us show $\boxed{\exists p \in marked(s.M)\ s.t.\ \mathbb{A}(p,a) = \texttt{true}.}$

    From $\sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}) = \texttt{true}$, we can deduce that $\exists id_{mp_i}\ s.t.\ \sigma(id_{mp_i}) = \texttt{true}$. Let us take an id_{mp_i} s.t. $\sigma(id_{mp_i}) = \texttt{true}$.

    By construction, there exist a $p \in pls(a)$, an $id_p \in Comps(\Delta)$, g_p, i_p and o_p such that:

    * $\gamma(p) = id_p$
    * $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$
    * $<\texttt{marked} \Rightarrow \texttt{id}_{\texttt{mp}_\texttt{i}}> \in o_p$

    Let us take such a p, id_p, g_p, i_p and o_p.

    By property of stable σ and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce $\sigma(id_{mp_i}) = \sigma(id_p)(\texttt{marked})$.

    By property of stable σ, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `determine_marked` process defined in the place design architecture, we can deduce:

    $$\sigma(id_p)(\texttt{marked}) = \sigma(id_p)(\texttt{sm}) > 0 \tag{29}$$

    From $\sigma(id_{mp_i}) = \sigma(id_p)(\texttt{marked})$, (29) and $\sigma(id_{mp_i}) = \texttt{true}$, we can deduce that $\sigma(id_p)(\texttt{marked}) = \texttt{true}$ and $(\sigma(id_p)(\texttt{sm}) > 0) = \texttt{true}$.

By property of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $s.M(p) = \sigma(id_p)(\texttt{sm})$.
From $s.M(p) = \sigma(id_p)(\texttt{sm})$ and $(\sigma(id_p)(\texttt{sm}) > 0) = \texttt{true}$, we can deduce $p \in marked(s.M)$, i.e. $s.M(p) > 0$.
Let us use p to prove the goal: $\boxed{\mathbb{A}(p, a) = \texttt{true}.}$
By definition of $p \in pls(a)$, $\mathbb{A}(p, a) = \texttt{true}.$

– **CASE** $\sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}) = \texttt{false}$:

Then, we can rewrite the goal as follows: $\boxed{\sum_{p \in marked(s.M)} \mathbb{A}(p, a) = \texttt{false}.}$
To prove the above goal, let us show $\boxed{\forall p \in marked(s.M)\ s.t.\ \mathbb{A}(p, a) = \texttt{false}.}$
Given a $p \in marked(s.M)$, let us show $\boxed{\mathbb{A}(p, a) = \texttt{false}.}$
Let us perform case analysis on $\mathbb{A}(p, a)$; there are 2 cases:

∗ **CASE** $\mathbb{A}(p, a) = \texttt{false}.$

∗ **CASE** $\mathbb{A}(p, a) = \texttt{true}$:
By construction, there exist an $id_p \in Comps(\Delta)$, g_{tp}, i_p, o_p and $id_{mp_i} \in Sigs(\Delta)$ such that:
· $\gamma(p) = id_p$
· $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$
· $<\texttt{marked} \Rightarrow \texttt{id}_{\texttt{mp}_\texttt{i}}> \in o_p$
Let us take such a id_p, g_p, i_p, o_p and id_{mp_i}.
By property of stable σ, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and $<\texttt{marked} \Rightarrow \texttt{id}_{\texttt{mp}_\texttt{i}}> \in o_p$, we can deduce $\sigma(id_{mp_i}) = \sigma(id_p)(\texttt{marked})$.
By property of stable σ, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `determine_marked` process defined in the place design architecture, we can deduce:

$$\sigma(id_p)(\texttt{marked}) = (\sigma(id_p)(\texttt{sm}) > 0) \tag{30}$$

From $\sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}) = \texttt{false}$, we can deduce $\sigma(id_{mp_i}) = \texttt{false}$.
From $\sigma(id_p)(\texttt{marked}) = \texttt{false}$, we can deduce $(\sigma(id_p)(\texttt{sm}) > 0) = \texttt{false}$.
By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $s.M(p) = \sigma(id_p)(\texttt{sm})$, and thus, we can deduce that $s.M(p) = 0$ (equivalent to $(s.M(p) > 0) = \texttt{false}$).
Contradicts $p \in marked(s.M)$ (i.e, $s.M(p) > 0$).

□

7.4 Falling edge and function executions

Lemma 38 (Falling edge equal function executions)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall f \in \mathcal{F}, id_f \in Outs(\Delta)$ s.t. $\gamma(f) = id_f$, $s'.ex(f) = \sigma'(id_f)$.*

Proof.

Given an $f \in \mathcal{F}$ and an $id_f \in Outs(\Delta)$ *s.t.* $\gamma(f) = id_f$, let us show $\boxed{s'.ex(f) = \sigma'(id_f).}$

By property of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we can deduce $s.ex(f) = s'.ex(f)$.
By construction, id_f is an output port identifier of Boolean type in the $\mathcal{H}$-VHDL design d assigned by the `function` process only during the initialization or during a rising edge phase.
By property of the $\mathcal{H}$-VHDL rising edge, stabilize relations, and the `function` process, we can deduce $\sigma(id_f) = \sigma'(id_f)$.
Rewriting the goal with $s.ex(f) = s'.ex(f)$ and $\sigma(id_f) = \sigma'(id_f)$, $\boxed{s.ex(f) = \sigma(id_f).}$
By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, $s.ex(f) = \sigma(id_f)$. □

7.5 Falling edge and firable transitions

Lemma 39 (Falling edge equal firable)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $t \in Firable(s') \Leftrightarrow \sigma'(id_t)(\texttt{s\_firable}) = \texttt{true}$.*

Proof.

Given a $t \in T$ and $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show that
$\boxed{t \in Firable(s') \Leftrightarrow \sigma'(id_t)(\texttt{s\_firable}) = \texttt{true}.}$

The proof is in two parts:

1. Assuming that $t \in Firable(s')$, let us show $\boxed{\sigma'(id_t)(\texttt{s\_firable}) = \texttt{true}.}$

   Appealing to Lemma 40: $\sigma'(id_t)(\texttt{s\_firable}) = \texttt{true}.$

2. Assuming that $\sigma'(id_t)(\texttt{s\_firable}) = \texttt{true}$, let us show $\boxed{t \in Firable(s').}$

   Appealing to Lemma 41: $t \in Firable(s').$

□

Lemma 40 (Falling edge equal firable 1)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $t \in Firable(s') \Rightarrow$*

$\sigma'(id_t)(\texttt{s\_firable}) = \texttt{true}$.

Proof.

Given a $t \in T$ and $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, and assuming that $t \in Firable(s')$, let us show $\boxed{\sigma'(id_t)(\texttt{s\_firable}) = \texttt{true}.}$

By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$. By property of the $\mathcal{H}$-VHDL falling edge relation, the stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `firable` process defined in the transition design architecture, we can deduce:

$$\sigma'(id_t)(\texttt{sfa}) = \sigma(id_t)(\texttt{se}) \;.\; \sigma(id_t)(\texttt{scc}) \;.\; \texttt{checktc}(\Delta(id_t), \sigma(id_t)) \tag{31}$$

Term $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$ is defined as follows:

$$\begin{aligned}
&\texttt{checktc}(\Delta(id_t), \sigma(id_t)) \\
&= \\
&\Big(\texttt{not}\ \sigma(id_t)(\texttt{srtc}) \;. \\
&\quad \big[\big(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP_A_B} \;.\; (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1) \\
&\qquad\qquad \;.\; (\sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1)\big) \\
&\quad + (\Delta(id_t)(\texttt{tt}) = \texttt{TEMP_A_A} \;.\; (\sigma(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{A}) - 1)) \\
&\quad + (\Delta(id_t)(\texttt{tt}) = \texttt{TEMP_A_INF} \;.\; (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1))\big]\Big) \\
&+ \big(\sigma(id_t)(\texttt{srtc}) \;.\; \Delta(id_t)(\texttt{tt}) \neq \texttt{NOT_TEMP} \;.\; \sigma(id_t)(\texttt{A}) = 1\big) \\
&+ \Delta(id_t)(\texttt{tt}) = \texttt{NOT_TEMP}
\end{aligned} \tag{32}$$

Rewriting the goal with (31): $\boxed{\sigma(id_t)(\texttt{se}) \;.\; \sigma(id_t)(\texttt{scc}) \;.\; \texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}.}$
Then, there are three points to prove:

1. $\boxed{\sigma(id_t)(\texttt{se}) = \texttt{true}}$:

   From $t \in Firable(s')$, we can deduce $t \in Sens(s'.M)$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s.M = s'.M$, and thus, we can deduce $t \in Sens(s.M)$.

   By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we know that $t \in Sens(s.M)$ implies $\sigma(id_t)(\texttt{se}) = \texttt{true}.$

2. $\boxed{\sigma(id_t)(\texttt{scc}) = \texttt{true}}$:

   By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$:

$$\sigma(id_t)(\texttt{scc}) = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t, c) = 1 \\ \texttt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t, c) = -1 \end{cases} \tag{33}$$

where $conds(t) = \{c \in \mathcal{C} \mid \mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1\}$.

Rewriting the goal with (33): $\boxed{\prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau,c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} = \texttt{true}.}$

To ease the reading, let us define $f(c) = \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau,c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}$.

Let us reason by induction on the left term of the goal:

- **BASE CASE**: $\texttt{true} = \texttt{true}.$
- **INDUCTION CASE**:

$$\prod_{c' \in conds(t) \setminus \{c\}} f(c') = \texttt{true}$$

$$\boxed{f(c)\ .\ \prod_{c' \in conds(t) \setminus \{c\}} f(c') = \texttt{true}.}$$

Rewriting the goal with the induction hypothesis, simplifying the goal, and unfolding the definition of $f(c)$: $\boxed{\begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau,c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} = \texttt{true}.}$

As $c \in conds(t)$, let us perform case analysis on $\mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1$:

(a) **CASE** $\mathbb{C}(t,c) = 1$: $\boxed{E_c(\tau,c) = \texttt{true}.}$

By definition of $t \in Firable(s')$, we can deduce that $s'.cond(c) = \texttt{true}$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we have $s'.cond(c) = E_c(\tau, c)$. Thus, $E_c(\tau,c) = \texttt{true}.$

(b) $\mathbb{C}(t,c) = -1$: $\boxed{\texttt{not}\ E_c(\tau,c) = \texttt{true}.}$

By definition of $t \in Firable(s')$, we can deduce that $s'.cond(c) = \texttt{false}$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we have $s'.cond(c) = E_c(\tau, c)$. Thus, $\texttt{not}\ E_c(\tau,c) = \texttt{true}.$

3. $\boxed{\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}}$:

By definition of $t \in Firable(s')$, we have $t \notin T_i \vee s'.I(t) \in I_s(t)$. Let us perform case analysis on $t \notin T_i \vee s'.I(t) \in I_s(t)$:

(a) **CASE** $t \notin T_i$: $\boxed{\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}}$

By construction, $<\texttt{transition\_type} \Rightarrow \texttt{NOT\_TEMP}> \in g_t$, and by property of the elaboration relation, we have $\Delta(id_t)(\texttt{tt}) = \texttt{NOT\_TEMP}$.
From $\Delta(id_t)(\texttt{tt}) = \texttt{NOT\_TEMP}$, and by definition of $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$, we can deduce $\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}.$

(b) **CASE** $s'.I(t) \in I_s(t)$: $\boxed{\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}}$

From $s'.I(t) \in I_s(t)$, we can deduce that $t \in T_i$. Thus, by construction, there exists $tt \in \{\texttt{TEMP\_A\_B}, \texttt{TEMP\_A\_A}, \texttt{TEMP\_A\_INF}\}$ s.t. $<\texttt{transition\_type} \Rightarrow tt> \in g_t$. By property of the elaboration relation, we have $\Delta(id_t)(\texttt{tt}) = tt$, and thus, we know $\Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMP}$. Therefore, we can simplify the term $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$ as follows:

$$
\begin{aligned}
&\texttt{checktc}(\Delta(id_t), \sigma(id_t)) \\
&= \\
&\Big(\texttt{not}\ \sigma(id_t)(\texttt{srtc})\ . \\
&\quad \big[\big(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_B}\ .\ (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1) \\
&\qquad\qquad .\ (\sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1)\big) \\
&\quad + (\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_A}\ . \\
&\qquad (\sigma(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{A}) - 1)) \\
&\quad + (\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_INF}\ . \\
&\qquad (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1))\big]\Big) \\
&+ \big(\sigma(id_t)(\texttt{srtc})\ .\ \sigma(id_t)(\texttt{A}) = 1\big)
\end{aligned}
\tag{34}
$$

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $s.reset_t(t) = \sigma(id_t)(\texttt{srtc})$.
Let us perform case analysis on the value $s.reset_t(t)$:

i. **CASE** $s.reset_t(t) = \texttt{true}$: $\boxed{\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}}$

From $s.reset_t(t) = \sigma(id_t)(\texttt{srtc})$, we can deduce that $\sigma(id_t)(\texttt{srtc}) = \texttt{true}$.
From $\sigma(id_t)(\texttt{srtc}) = \texttt{true}$, we can simplify the term $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$ as follows:

$$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \big(\sigma(id_t)(\texttt{A}) = 1\big) \tag{35}$$

Rewriting the goal with (35), and simplifying the goal: $\boxed{\sigma(id_t)(\texttt{A}) = 1.}$

By definition of $E_c, \tau \vdash s \overset{\downarrow}{\rightarrow} s'$ (Rule **??**), from $t \in Sens(s.M)$ and $s.reset_t(t) = \texttt{true}$, we can deduce $s'.I(t) = 1$. We know that $s'.I(t) \in I_s(t)$, and thus, we have $1 \in I_s(t)$.
By definition of $1 \in I_s(t)$, there exist an $a \in \mathbb{N}^*$ and a $ni \in \mathbb{N}^* \sqcup \{\infty\}$ s.t. $I_s(t) = [a, ni]$ and $1 \in [a, ni]$.
By definition of $1 \in [a, ni]$, we have $a \leq 1$, and since $a \in \mathbb{N}^*$, we can deduce $a = 1$.
By construction, $<\texttt{time\_A\_value} \Rightarrow a> \in i_t$, and by property of stable σ, we have $\sigma(id_t)(\texttt{A}) = a = 1.$

ii. **CASE** $s.reset_t(t) = \texttt{false}$: $\boxed{\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}}$

From $s.reset_t(t) = \sigma(id_t)(\texttt{srtc})$, we can deduce $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$.

From $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$, we can simplify the term $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$ as follows:

$$\begin{array}{c} \texttt{checktc}(\Delta(id_t), \sigma(id_t)) \\ = \\ \big(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_B} \quad . (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1) \\ \qquad\qquad\qquad\qquad\quad . (\sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1)\big) \\ +(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_A} \; . (\sigma(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{A}) - 1)) \\ +(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_INF} \; . (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1)) \end{array} \tag{36}$$

Let us perform case analysis on $I_s(t)$; there are two cases:

- **CASE** $I_s(t) = [a, b]$ where $a, b \in \mathbb{N}^*$; then, either $a = b$ or $a \neq b$:
  - **CASE** $a = b$:
    Then, we have $I_s(t) = [a, a]$, and by construction $<\texttt{transition\_type}\Rightarrow \texttt{TEMP\_A\_A}> \in g_t$. By property of the elaboration relation, we have $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_A}$; thus we can simplify the $\texttt{checktc}$ term as follows:

    $$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = (\sigma(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{A}) - 1) \tag{37}$$

    Rewriting the goal with (37), and simplifying the goal:
    $\boxed{\sigma(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{A}) - 1.}$

    From $s'.I(t) \in [a, a]$, we can deduce that $s'.I(t) = a$. Let us perform case analysis on $s.I(t) < u(I_s(t))$ or $s.I(t) \geq u(I_s(t))$:
    - **CASE** $s.I(t) < u(I_s(t))$:

      By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $s.I(t) = \sigma(id_t)(\texttt{stc})$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we have $s'.I(t) = s.I(t)+1$. From $s'.I(t) = a$ and $s'.I(t) = s.I(t)+1$, we can deduce $a - 1 = s.I(t)$.
      By construction, $<\texttt{time\_A\_value} \Rightarrow a> \in i_t$, and by property of stable σ, we have $\sigma(id_t)(\texttt{A}) = a$.
      Rewriting the goal with $\sigma(id_t)(\texttt{A}) = a$, $s.I(t) = \sigma(id_t)(\texttt{stc})$, and $a - 1 = s.I(t)$: tautology.
    - **CASE** $s.I(t) \geq u(I_s(t))$:

      In the case where $s.I(t) > u(I_s(t))$, then $s.I(t) > a$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we have $s.I(t) = s'.I(t) = a$. Then, $a > a$ is a contradiction.

      In the case where $s.I(t) = u(I_s(t))$, then $s.I(t) = a$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we have $s'.I(t) = s.I(t) + 1$. Then, we have $s'.I(t) = a$ and $s'.I(t) = a + 1$. Then, $a = a + 1$ is a contradiction.
  - **CASE** $a \neq b$: $\boxed{\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}}$
    Then, we have $I_s(t) = [a, b]$, and by construction $<\texttt{transition\_type}\Rightarrow \texttt{TEMP\_A\_B}> \in g_t$. By property of the elaboration relation, we have $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_B}$; thus we can simplify the term $\texttt{checktc}$ as follows:

    $$\begin{array}{c} \texttt{checktc}(\Delta(id_t), \sigma(id_t)) \\ = \\ (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1) \; . \; (\sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1) \end{array} \tag{38}$$

Rewriting the goal with (38), and simplifying the goal:
$\boxed{(\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1) \wedge (\sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1).}$

Let us perform case analysis on $s.I(t) < u(I_s(t))$ or $s.I(t) \geq u(I_s(t))$:

∗ **CASE** $s.I(t) < u(I_s(t))$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $s.I(t) = \sigma(id_t)(\texttt{stc})$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$(Rule **??**), we have $s'.I(t) = s.I(t) + 1$. By definition of $s'.I(t) \in [a, b]$:
$\Rightarrow a \leq s'.I(t) \leq b$.
$\Rightarrow a \leq s'.I(t) \wedge s'.I(t) \leq b$
$\Rightarrow a \leq s.I(t) + 1 \wedge s.I(t) + 1 \leq b$
$\Rightarrow a - 1 \leq s.I(t) \wedge s.I(t) \leq b - 1$
By construction, <`time_A_value` $\Rightarrow a>$ $\in i_t$ and <`time_B_value` $\Rightarrow b>$ $\in i_t$, and by property of stable σ, we have $\sigma(id_t)(\texttt{A}) = a$ and $\sigma(id_t)(\texttt{B}) = b$.
Rewriting the goal with $\sigma(id_t)(\texttt{A}) = a$, $\sigma(id_t)(\texttt{B}) = b$ and $s.I(t) = \sigma(id_t)(\texttt{stc})$:
$a - 1 \leq s.I(t) \wedge s.I(t) \leq b - 1$.

∗ **CASE** $s.I(t) \geq u(I_s(t))$:

In the case where $s.I(t) > u(I_s(t))$, then $s.I(t) > b$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we have $s.I(t) = s'.I(t) = b$. Then, $b > b$ is a contradiction.

In the case where $s.I(t) = u(I_s(t))$, then $s.I(t) = b$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we have $s'.I(t) = s.I(t) + 1$.
By definition of $s'.I(t) \in [a, b]$, we have $s'.I(t) \leq b$:
$\Rightarrow s.I(t) + 1 \leq b$
$\Rightarrow$ $b + 1 \leq b$ is contradiction.

- **CASE** $I_s(t) = [a, \infty]$ where $a \in \mathbb{N}^*$: $\boxed{\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}}$
By construction <`transition_type` $\Rightarrow$ `TEMP_A_INF`$> \in g_t$. By property of the elaboration relation, we have $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_INF}$; thus we can simplify the term `checktc` as follows:

$$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1)) \tag{39}$$

Rewriting the goal with (39), and simplifying the goal:
$\boxed{\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1.}$

From $s'.I(t) \in [a, \infty]$, we can deduce $a \leq s'.I(t)$. Then, let us perform case analysis on $s.I(t) \leq l(I_s(t))$ or $s.I(t) > l(I_s(t))$:

– **CASE** $s.I(t) \leq l(I_s(t))$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $s.I(t) = \sigma(id_t)(\texttt{stc})$.
By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we have $s'.I(t) = s.I(t) + 1$:
$\Rightarrow s'.I(t) \geq a$
$\Rightarrow s.I(t) + 1 \geq a$
$\Rightarrow s.I(t) \geq a - 1$

By construction, $<$`time_A_value` $\Rightarrow a>$ $\in i_t$, and by property of stable σ, we have $\sigma(id_t)(\texttt{A}) = a$.
Rewriting the goal with $\sigma(id_t)(\texttt{A}) = a$ and $s.I(t) = \sigma(id_t)(\texttt{stc})$:
$s.I(t) \geq a - 1$.

– **CASE** $s.I(t) > l(I_s(t))$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $\sigma(id_t)(\texttt{stc}) = l(I_s(t)) = a$.
By construction, $<$`time_A_value` $\Rightarrow a>$ $\in i_t$, and by property of stable σ, we have $\sigma(id_t)(\texttt{A}) = a$.
Rewriting the goal with $\sigma(id_t)(\texttt{stc}) = a$ and $\sigma(id_t)(\texttt{A}) = a$: $a \geq a - 1$.

□

Lemma 41 (Falling Edge Equal Firable 2)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $\sigma'(id_t)(\texttt{s\_firable}) = \texttt{true} \Rightarrow t \in Firable(s')$.*

Proof.

Given a $t \in T$ and $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, and assuming that $\sigma'(id_t)(\texttt{s\_firable}) = \texttt{true}$, let us show $\boxed{t \in Firable(s').}$

By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$. By property of the $\mathcal{H}$-VHDL falling edge relation, the stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `firable` process defined in the transition design architecture, we can deduce:

$$\sigma'(id_t)(\texttt{sfa}) = \sigma(id_t)(\texttt{se}) \;.\; \sigma(id_t)(\texttt{scc}) \;.\; \texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true} \tag{40}$$

From (40), we can deduce:

$$\sigma(id_t)(\texttt{se}) = \texttt{true} \tag{41}$$
$$\sigma(id_t)(\texttt{scc}) = \texttt{true} \tag{42}$$
$$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true} \tag{43}$$

Term $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$ as the same definition as in Lemma Falling edge equal firable 1.
By definition of $t \in Firable(s')$, there are three points to prove:

1. $\boxed{t \in Sens(s'.M)}$
2. $\boxed{\forall c \in \mathcal{C},\ \mathbb{C}(t, c) = 1 \Rightarrow s'.cond(c) = \texttt{true} \text{ and } \mathbb{C}(t, c) = -1 \Rightarrow s'.cond(c) = \texttt{false}}$
3. $\boxed{t \notin T_i \vee s'.I(t) \in I_s(t)}$

Let us prove these three points:

1. $\boxed{t \in Sens(s'.M)}$:

   By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s.M = s'.M$. Rewriting the goal with $s.M = s'.M$: $\boxed{t \in Sens(s.M).}$

   By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $\sigma(id_t)(\texttt{se}) = \texttt{true} \Leftrightarrow t \in Sens(s.M)$.

   From $\sigma(id_t)(\texttt{se}) = \texttt{true}$, we can deduce: $t \in Sens(s.M)$.

2. $\boxed{\forall c \in \mathcal{C},\ \mathbb{C}(t,c) = 1 \Rightarrow s'.cond(c) = \texttt{true} \text{ and } \mathbb{C}(t,c) = -1 \Rightarrow s'.cond(c) = \texttt{false}}$

   Given a $c \in \mathcal{C}$, there are two points to prove:

   (a) $\boxed{\mathbb{C}(t,c) = 1 \Rightarrow s'.cond(c) = \texttt{true}.}$

   (b) $\boxed{\mathbb{C}(t,c) = -1 \Rightarrow s'.cond(c) = \texttt{false}.}$

   Let us prove these two points:

   (a) Assuming that $\mathbb{C}(t,c) = 1$, let us show $\boxed{s'.cond(c) = \texttt{true}.}$

   By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have:

$$\sigma(id_t)(\texttt{scc}) = \prod_{c' \in conds(t)} \begin{cases} E_c(\tau, c') & if\ \mathbb{C}(t,c') = 1 \\ \texttt{not}(E_c(\tau, c')) & if\ \mathbb{C}(t,c') = -1 \end{cases} = \texttt{true} \tag{44}$$

   where $conds(t) = \{c_i \in \mathcal{C} \mid \mathbb{C}(t,c_i) = 1 \vee \mathbb{C}(t,c_i) = -1\}$.
   From $\mathbb{C}(t,c) = 1$, we can deduce $c \in conds(t)$. By definition of the product expression, we have:

$$E_c(\tau, c)\ .\ \prod_{c' \in conds(t) \setminus \{c\}} \begin{cases} E_c(\tau, c') & if\ \mathbb{C}(t,c') = 1 \\ \texttt{not}(E_c(\tau, c')) & if\ \mathbb{C}(t,c') = -1 \end{cases} = \texttt{true} \tag{45}$$

   From (45), we can deduce that $E_c(\tau, c) = \texttt{true}$.
   By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we have $s'.cond(c) = E_c(\tau, c)$.
   Rewriting the goal with $s'.cond(c) = E_c(\tau, c)$ and $E_c(\tau, c) = \texttt{true}$: tautology.

   (b) Assuming that $\mathbb{C}(t,c) = -1$, let us show $\boxed{s'.cond(c) = \texttt{false}.}$

   By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have:

$$\sigma(id_t)(\texttt{scc}) = \prod_{c' \in conds(t)} \begin{cases} E_c(\tau, c') & if\ \mathbb{C}(t,c') = 1 \\ \texttt{not}(E_c(\tau, c')) & if\ \mathbb{C}(t,c') = -1 \end{cases} = \texttt{true} \tag{46}$$

   where $conds(t) = \{c' \in \mathcal{C} \mid \mathbb{C}(t,c') = 1 \vee \mathbb{C}(t,c') = -1\}$.

From $\mathbb{C}(t,c) = -1$, we can deduce $c \in conds(t)$. By definition of the product expression, we have:

$$\texttt{not}\ E_c(\tau,c)\ .\ \prod_{c' \in conds(t)\setminus\{c\}} \begin{cases} E_c(\tau,c') & if\ \mathbb{C}(t,c') = 1 \\ \texttt{not}(E_c(\tau,c')) & if\ \mathbb{C}(t,c') = -1 \end{cases} = \texttt{true} \tag{47}$$

From (47), we can deduce that $E_c(\tau,c) = \texttt{false}$.

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), we have $s'.cond(c) = E_c(\tau,c)$.

Rewriting the goal with $s'.cond(c) = E_c(\tau,c)$ and $E_c(\tau,c) = \texttt{false}$: tautology.

3. $\boxed{t \notin T_i \vee s'.I(t) \in I_s(t)}$

Reasoning on $\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}$, there are 3 cases:

(a) $\big(\texttt{not}\ \sigma(id_t)(\texttt{srtc})\ .\ [\dots]\big) = \texttt{true}$[a]

(b) $\big(\sigma(id_t)(\texttt{srtc})\ .\ \Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMP}\ .\ \sigma(id_t)(\texttt{A}) = 1\big) = \texttt{true}$

(c) $\big(\Delta(id_t)(\texttt{tt}) = \texttt{NOT\_TEMP}\big) = \texttt{true}$

(a) **CASE** $\big(\texttt{not}\ \sigma(id_t)(\texttt{srtc})\ .\ [\dots]\big) = \texttt{true}$:

Then, we can deduce $\texttt{not}\ \sigma(id_t)(\texttt{srtc}) = \texttt{true}$ and $[\dots] = \texttt{true}$.

From $\texttt{not}\ \sigma(id_t)(\texttt{srtc}) = \texttt{true}$, we can deduce $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$, and from $[\dots] = \texttt{true}$, we have three other cases:

i. **CASE** $\big(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_B}\ .\ (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1)\ .\ (\sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1)\big) = \texttt{true}$

ii. **CASE** $(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_A}\ .\ (\sigma(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{A}) - 1)) = \texttt{true}$

iii. **CASE** $(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_INF}\ .\ (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1)) = \texttt{true}$

Let us prove the goal is these three contexts:

i. **CASE** $\big(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_B}\ .\ (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1)\ .\ (\sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1)\big) = \texttt{true}$:

Then, converting Boolean equalities into intuitionistic predicates, we have:

- $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_B}$
- $\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1$
- $\sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1$

By property of the elaboration relation, and $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_B}$, there exist $a, b \in \mathbb{N}^*$ s.t. $I_s(t) = [a,b]$. Let us take such an a and b. Then, let us show $\boxed{s'.I(t) \in I_s(t).}$

Rewriting the goal with $I_s(t) = [a,b]$: $\boxed{s'.I(t) \in [a,b].}$

By construction, $<\texttt{time\_A\_value} \Rightarrow a>$ and $<\texttt{time\_B\_value} \Rightarrow b>$, and by property of stable σ, we have $\sigma(id_t)(\texttt{A}) = a$ and $\sigma(id_t)(\texttt{B}) = b$.

Rewriting the goal with $\sigma(id_t)(\texttt{A}) = a$ and $\sigma(id_t)(\texttt{B}) = b$, and by definition of $\in$: $\boxed{\sigma(id_t)(\texttt{A}) \leq s'.I(t) \leq \sigma(id_t)(\texttt{B}).}$

Now, let us perform case analysis on $s.I(t) \leq u(I_s(t))$ or $s.I(t) > u(I_s(t))$:

- **CASE** $s.I(t) \leq u(I_s(t))$:

  By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $s.I(t) = \sigma(id_t)(\texttt{stc})$.
  From $\sigma(id_t)(\texttt{se}) = \texttt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$, we can deduce $s.reset_t(t) = \texttt{false}$. Then, by definition of $E_c, \tau \vdash s \overset{\downarrow}{\rightarrow} s'$ (Rule **??**), we have $s'.I(t) = s.I(t) + 1$.
  $\Rightarrow$ $\boxed{\sigma(id_t)(\texttt{A}) \leq s.I(t) + 1 \leq \sigma(id_t)(\texttt{B})}$ (by $s'.I(t) = s.I(t) + 1$)
  $\Rightarrow$ $\boxed{\sigma(id_t)(\texttt{A}) \leq \sigma(id_t)(\texttt{stc}) + 1 \leq \sigma(id_t)(\texttt{B})}$ (by $s.I(t) = \sigma(id_t)(\texttt{stc})$)
  $\Rightarrow$ $\boxed{\sigma(id_t)(\texttt{A}) - 1 \leq \sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1}$
  We assumed $\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1$ and $\sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1$, and thus we can deduce: $\sigma(id_t)(\texttt{A}) - 1 \leq \sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1$

- **CASE** $s.I(t) > u(I_s(t))$:

  By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $\sigma(id_t)(\texttt{stc}) = u(I_s(t)) = b$.
  Then, from $\sigma(id_t)(\texttt{stc}) \leq \sigma(id_t)(\texttt{B}) - 1$, $\sigma(id_t)(\texttt{stc}) = u(I_s(t)) = b$ and $\sigma(id_t)(\texttt{B}) = b$, we can deduce the following contradiction:
  $\sigma(id_t)(\texttt{B}) \leq \sigma(id_t)(\texttt{B}) - 1$.

ii. $(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_A} \;.\; (\sigma(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{A}) - 1)) = \texttt{true}$:
Then, converting Boolean equalities into logic predicates, we have:

- $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_A}$
- $\sigma(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{A}) - 1$

By property of the elaboration relation, and $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_A}$, there exist $a \in \mathbb{N}^*$ s.t. $I_s(t) = [a, a]$. Let us take such an a. Then, let us show $\boxed{s'.I(t) \in I_s(t).}$
Rewriting the goal with $I_s(t) = [a, a]$: $\boxed{s'.I(t) \in [a, a].}$
By construction, <time_A_value $\Rightarrow a>$, and by property of stable σ, we have $\sigma(id_t)(\texttt{A}) = a$.
Rewriting the goal with $\sigma(id_t)(\texttt{A}) = a$, unfolding the definition of $\in$, and simplifying the goal: $\boxed{s'.I(t) = \sigma(id_t)(\texttt{A}).}$
Now, let us perform case analysis on $s.I(t) \leq u(I_s(t))$ or $s.I(t) > u(I_s(t))$:

- **CASE** $s.I(t) \leq u(I_s(t))$:

  By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $s.I(t) = \sigma(id_t)(\texttt{stc})$.
  From $\sigma(id_t)(\texttt{se}) = \texttt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$, we can deduce $s.reset_t(t) = \texttt{false}$. Then, by definition of $E_c, \tau \vdash s \overset{\downarrow}{\rightarrow} s'$ (Rule **??**), we have $s'.I(t) = s.I(t) + 1$.
  $\Rightarrow$ $\boxed{s.I(t) + 1 = \sigma(id_t)(\texttt{A})}$ (by $s'.I(t) = s.I(t) + 1$)
  $\Rightarrow$ $\boxed{\sigma(id_t)(\texttt{stc}) + 1 = \sigma(id_t)(\texttt{A})}$ (by $s.I(t) = \sigma(id_t)(\texttt{stc})$)
  $\Rightarrow$ $\sigma(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{A}) - 1$ (assumption)

- **CASE** $s.I(t) > u(I_s(t))$:

  By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $\sigma(id_t)(\texttt{stc}) = u(I_s(t)) = a$.
  Then, from $\sigma(id_t)(\texttt{stc}) = \sigma(id_t)(\texttt{A}) - 1$, $\sigma(id_t)(\texttt{stc}) = u(I_s(t)) = a$, $\sigma(id_t)(\texttt{A}) = a$, and

$a \in \mathbb{N}^*$, we can derive the following contradiction:
$\sigma(id_t)(\texttt{A}) = \sigma(id_t)(\texttt{A}) - 1.$

iii. $(\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_INF} \;.\; (\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1)) = \texttt{true}$:
Then, converting Boolean equalities into logic predicates, we have:

- $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_INF}$
- $\sigma(id_t)(\texttt{stc}) \geq \sigma(id_t)(\texttt{A}) - 1$

By property of the elaboration relation, and $\Delta(id_t)(\texttt{tt}) = \texttt{TEMP\_A\_INF}$, there exist $a \in \mathbb{N}^*$ s.t. $I_s(t) = [a, \infty]$. Let us take such an a. Then, let us show $\boxed{s'.I(t) \in I_s(t).}$
Rewriting the goal with $I_s(t) = [a, \infty]$: $\boxed{s'.I(t) \in [a, \infty].}$
By construction, $<\texttt{time\_A\_value} \Rightarrow a>$, and by property of stable σ, we have $\sigma(id_t)(\texttt{A}) = a$.
Rewriting the goal with $\sigma(id_t)(\texttt{A}) = a$, unfolding the definition of $\in$, and simplifying the goal:
$\boxed{\sigma(id_t)(\texttt{A}) \leq s'.I(t).}$
Now, let us perform case analysis on $s.I(t) \leq l(I_s(t))$ or $s.I(t) > l(I_s(t))$:

- **CASE** $s.I(t) \leq l(I_s(t))$:

  By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $s.I(t) = \sigma(id_t)(\texttt{stc})$.
  From $\sigma(id_t)(\texttt{se}) = \texttt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$, we can deduce $s.reset_t(t) = \texttt{false}$. Then, by definition of $E_c, \tau \vdash s \overset{\downarrow}{\to} s'$ (Rule **??**), we have $s'.I(t) = s.I(t) + 1$.
  $\Rightarrow$ $\boxed{\sigma(id_t)(\texttt{A}) \leq s.I(t) + 1}$ (by $s'.I(t) = s.I(t) + 1$)
  $\Rightarrow$ $\boxed{\sigma(id_t)(\texttt{A}) \leq \sigma(id_t)(\texttt{stc}) + 1}$ (by $s.I(t) = \sigma(id_t)(\texttt{stc})$)
  $\Rightarrow$ $\sigma(id_t)(\texttt{A}) - 1 \leq \sigma(id_t)(\texttt{stc})$ (assumption)

- **CASE** $s.I(t) > l(I_s(t))$:

  By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, we have $\sigma(id_t)(\texttt{stc}) = l(I_s(t)) = a$.
  From $\sigma(id_t)(\texttt{se}) = \texttt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)(\texttt{srtc}) = \texttt{false}$, we can deduce $s.reset_t(t) = \texttt{false}$. Then, by definition of $E_c, \tau \vdash s \overset{\downarrow}{\to} s'$ (Rule **??**), we have $s'.I(t) = s.I(t) + 1$.
  $\Rightarrow$ $\boxed{\sigma(id_t)(\texttt{A}) \leq s.I(t) + 1}$ (by $s'.I(t) = s.I(t) + 1$)
  $\Rightarrow$ $\boxed{a \leq s.I(t) + 1}$ (by $\sigma(id_t)(\texttt{A}) = a$)
  $\Rightarrow$ $\boxed{a < s.I(t)}$
  $\Rightarrow$ $l(I_s(t)) < s.I(t)$ (assumption)

(b) $\big(\sigma(id_t)(\texttt{srtc}) \;.\; \Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMP} \;.\; \sigma(id_t)(\texttt{A}) = 1\big) = \texttt{true}$
Then, converting Boolean equalities into logic predicates, we have:

- $\sigma(id_t)(\texttt{srtc}) = \texttt{true}$
- $\Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMP}$
- $\sigma(id_t)(\texttt{A}) = 1$

By property of the elaboration relation, and $\Delta(id_t)(\texttt{tt}) \neq \texttt{NOT\_TEMP}$, there exist an $a \in \mathbb{N}^*$ and a $ni \in \mathbb{N}^* \sqcup \{\infty\}$ s.t. $I_s(t) = [a, ni]$. Let us take such an a and ni.

By construction, $<\texttt{time\_A\_value} \Rightarrow a> \in i_t$, and by property of stable σ, we have $\sigma(id_t)(\texttt{A}) = a$. Thus, we can deduce $a = 1$ and $I_s(t) = [1, ni]$.

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\approx} \sigma$, from $\sigma(id_t)(\texttt{se}) = \texttt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)(\texttt{srtc}) = \texttt{true}$, we can deduce $s.reset_t(t) = \texttt{true}$.

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$ (Rule **??**), $t \in Sens(s.M)$ and $s.reset_t(t) = \texttt{true}$, we have $s'.I(t) = 1$.

Now, let us show $\boxed{s'.I(t) \in I_s(t)}$.

Rewriting the goal with $s'.I(t) = 1$ and $I_s(t) = [1, ni]$: $1 \in [1, ni]$.

(c) $\big(\Delta(id_t)(\texttt{tt}) = \texttt{NOT\_TEMP}\big) = \texttt{true}$

Let us show $\boxed{t \notin T_i.}$

By property of the elaboration relation and $\Delta(id_t)(\texttt{tt}) = \texttt{NOT\_TEMP}$, we have $t \notin T_i$.

□

[a] See equation (32) for the full definition.

Lemma 42 (Falling edge equal not firable)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $t \notin Firable(s') \Leftrightarrow \sigma'(id_t)(\texttt{s\_firable}) = \texttt{false}$.*

Proof.

Proving the above lemma is trivial by appealing to Lemma 39 and by reasoning on contrapositives. □

7.6 Falling edge and fired transitions

Definition 14 (Fired)**.** *A transition $t \in T$ is said to be fired at the SITPN state $s = <M, I, reset_t, ex, cond>$, iff there exists a subset $Fset \subseteq T$ such that $IsFiredSet(s, Fset)$ and $t \in Fset$.*

Definition 15 (IsFiredSet)**.** *Given an $sitpn \in SITPN$, a SITPN state $s \in S(sitpn)$, and a subset $Fset \subseteq T$, the $IsFiredSet$ relation is defined as follows:*
$IsFiredSet(s, Fset) \equiv IsFiredSetAux(s, T, \emptyset, Fset)$

Definition 16 (IsFiredSetAux)**.** *The $IsFiredSetAux$ relation is defined by the following rules:*

FSetEmp
$$\frac{}{IsFiredSetAux(s, \emptyset, F, F)}$$

FSetFired
$$\frac{t \in Firable(s) \quad t \in Sens(s.M - \sum_{t_i \in Pr(t,F)} pre(t_i)) \quad IsFiredSetAux(s, T_s, F \cup \{t\}, Fset)}{IsFiredSetAux(s, T_s \cup \{t\}, F, Fset)} \quad \begin{array}{l} \nexists t' \in T_s \text{ s.t. } t' \succ t \\ Pr(t,F) = \{t' \mid t' \succ t \wedge t' \in F\} \end{array}$$

FSetNotFirable
$$\frac{t \notin Firable(s) \quad IsFiredSetAux(s, T_s, F, Fset)}{IsFiredSetAux(s, T_s \cup \{t\}, F, Fset)} \quad \nexists t' \in T_s \text{ s.t. } t' \succ t$$

FSetNotSens
$$\frac{t \notin Sens(s.M - \sum_{t_i \in Pr(t,F)} pre(t_i)) \quad IsFiredSetAux(s, T_s, F, Fset)}{IsFiredSetAux(s, T_s \cup \{t\}, F, Fset)} \quad \begin{array}{l} \nexists t' \in T_s \text{ s.t. } t' \succ t \\ Pr(t,F) = \{t' \mid t' \succ t \wedge t' \in F\} \end{array}$$

Lemma 43 (Falling edge equal fired set)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall t \in T$, $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $\forall Fset \subseteq T$, s.t. $IsFiredSet(s', Fset)$, $t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$.*

Proof.

Given a $t \in T$, and $id_t \in Comps(\Delta)$, and a $Fset \subseteq T$ s.t. $IsFiredSet(s', Fset)$, let us show $\boxed{t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}.}$

By definition of $IsFiredSet(s', Fset)$, we have $IsFiredSetAux(s', T, \emptyset, Fset)$.
Then, we can appeal to Lemma 44 to solve the goal, but first we must prove the following *extra hypothesis* (i.e, one of the premise of Lemma 44):

$\boxed{\begin{array}{l} \forall t' \in T, id_{t'} \in Comps(\Delta) \text{ s.t. } \gamma(t') = id_{t'}, \\ (t' \in \emptyset \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in \emptyset \ \vee \ t' \in T). \end{array}}$

Given a $t' \in T$ and an $id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$, there are two points to prove:

1. $\boxed{t' \in \emptyset \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}}$
2. $\boxed{\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in \emptyset \ \vee \ t' \in T}$

Let us show these two points:

1. Assuming $t' \in \emptyset$, let us show $\boxed{\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}.}$

   $t' \in \emptyset$ is a contradiction.

2. Assuming $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$, let us show $\boxed{t' \in \emptyset \ \vee \ t' \in T.}$

   By definition, $t' \in T$.

□

Lemma 44 (Falling edge equal fired set aux)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $\forall F \subseteq T$, $T_s \subseteq T$, $Fset \subseteq T$, assume that:*

- $IsFiredSetAux(s', T_s, F, Fset)$
- *EH (Extra. Hypothesis):*
  $\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ *s.t.* $\gamma(t') = id_{t'}$,
  $(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{\textit{fired}}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{\textit{fired}}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s)$.

then $t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{\textit{fired}}) = \texttt{true}$.

Proof.

Given a $t \in T$, an $id_t \in Comps(\Delta)$, a $T_s, F, Fset \subseteq T$, and assuming $IsFiredSetAux(s', T_s, F, Fset)$, let us show

$\big(\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s)\big) \Rightarrow t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$.

Let us use rule induction on $IsFiredSetAux(s', T_s, F, Fset)$. Let us define the property P taken into account in the induction scheme as follows

$$P(s', T_s, F, Fset)$$
$$\equiv$$
$$(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge \big(\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s)\big) \Rightarrow t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$$

- **CASE** FSetEmp: we must show $P(s', \emptyset, F, F)$, i.e.

  $\big(\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
  $(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in \emptyset)\big) \Rightarrow t \in F \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$.

  Assuming

  $\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
  $(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in \emptyset)$

  we can easily show $t \in F \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$.

- **CASE** FSetFired:

  Assuming

  – $t \in Firable(s')$

– $t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i))$

– $IsFiredSetAux(s', T_s, F \cup \{t\}, Fset)$

– $\nexists t' \in T_s \ s.t. \ t' \succ t$

– $Pr(t, F) = \{t' \mid t' \succ t \wedge t' \in F\}$

and the induction hypothesis (i.e. $P(s', T_s, F \cup \{t\}, Fset)$)

$\big(\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in F \cup \{t\} \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true})$
$\wedge\ (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \cup \{t\}\ \vee\ t' \in T_s)\big) \Rightarrow$
$t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$

we must show

$\big(\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true})$
$\wedge\ (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F\ \vee\ t' \in T_s \cup \{t\})\big) \Rightarrow$
$t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$

Assuming the following hypothesis that we will call EH (for Extra Hypothesis)

$\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F\ \vee\ t' \in T_s \cup \{t\})$

we must show

$t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$

Appealing to the induction hypothesis, to prove the current goal, it is sufficient to prove that

$\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in F \cup \{t\} \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true})$
$\wedge\ (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \cup \{t\}\ \vee\ t' \in T_s)$

Given a $t' \in T$, an $id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$, we must show that

$(t' \in F \cup \{t\} \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true})$
$\wedge\ (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \cup \{t\}\ \vee\ t' \in T_s)$

There are two points to prove

1. Assuming $t' \in F \cup \{t\}$, then $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$
2. Assuming $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$, then $t' \in F \cup \{t\} \ \vee \ t' \in T_s$

1. Assuming $t' \in F \cup \{t\}$, let us show $\boxed{\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}}$. Let us perform case analysis on $t' \in F \cup \{t\}$; there are 2 cases:
   - **CASE** $t' \in F$: Appealing to EH, the goal is trivially proved.
   - **CASE** $t' = t$: Then, $id_t = id_{t'}$, and we must show $\boxed{\sigma'(id_t)(\texttt{fired}) = \texttt{true}}$.
     By definition of id_t, there exist a g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$.
     By property of the stabilize relation and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `fired_evaluation` process defined in the `transition` design architecture:

     $$\sigma(id_t)(\texttt{fired}) = \sigma(id_t)(\texttt{sfa}) \ . \ \sigma(id_t)(\texttt{spc})$$

     Rewriting the goal with the above equation: $\boxed{\sigma(id_t)(\texttt{sfa}) \ . \ \sigma(id_t)(\texttt{spc}) = \texttt{true}.}$
     Then, there are two points to prove:
     (a) $\boxed{\sigma(id_t)(\texttt{sfa}) = \texttt{true}.}$
     Appealing to Lemma 39, and since $t \in Firable(s')$, we can deduce $\sigma(id_t)(\texttt{sfa}) = \texttt{true}.$
     (b) $\boxed{\sigma(id_t)(\texttt{spc}) = \texttt{true}.}$
     Appealing to Lemma 45, and since $t \in Sens(s'M - \sum_{t_i \in Pr(t,F)} pre(t_i))$, we can deduce $\sigma(id_t)(\texttt{spc}) = \texttt{true}.$
2. Assuming $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$, let us show $\boxed{t' \in F \cup \{t\} \ \vee \ t' \in T_s}$. Appealing to EH, we can deduce that $t' \in F \ \vee \ t' \in T_s \cup \{t\}$. Then, the goal is trivially shown.

- **CASE** FSetNotFirable: Assuming
  - $t \notin Firable(s')$
  - $IsFiredSetAux(s', T_s, F, Fset)$
  - $\nexists t' \in T_s \ s.t. \ t' \succ t$

  and the induction hypothesis (i.e. $P(s', T_s, F, Fset)$)

  $$\begin{array}{l}\big(\forall t' \in T, \ id_{t'} \in Comps(\Delta) \text{ s.t. } \gamma(t') = id_{t'}, \\ (t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \\ \wedge \ (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s)\big) \Rightarrow \\ t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}\end{array}$$

  we must show

$\big(\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true})$
$\wedge\ (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F\ \vee\ t' \in T_s \cup \{t\})\big) \Rightarrow$
$t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$

Assuming the following hypothesis that we will call EH (for Extra Hypothesis)

$\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F\ \vee\ t' \in T_s \cup \{t\})$

we must show

$t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$

Appealing to the induction hypothesis, to prove the current goal, it is sufficient to prove that

$\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F\ \vee\ t' \in T_s)$

Given a $t' \in T$, an $id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$, we must show that

$(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F\ \vee\ t' \in T_s)$

There are two points to prove

1. Assuming $t' \in F$, then $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$
2. Assuming $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$, then $t' \in F\ \vee\ t' \in T_s$

1. Assuming $t' \in F$, let us show $\boxed{\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}}$.
   Appealing to EH, the goal is trivially shown.
2. Assuming $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$, let us show $\boxed{t' \in F\ \vee\ t' \in T_s}$.
   Appealing to EH, we can deduce $t' \in F\ \vee\ t' \in T_s \cup \{t\}$. Let us perform case analysis on $t' \in F\ \vee\ t' \in T_s \cup \{t\}$; there are 2 cases:
   - **CASE** $t' \in F$: trivially shown, as it is an assumption.
   - **CASE** $t' \in T_s \cup \{t\}$: In the case where $t' \in T_s$, the goal is trivially shown. In the case where $t' = t$, we can prove a contradiction based on $t \notin Firable(s')$ and $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$.
     Since $t = t'$, then $id_t = id_{t'}$, and we know that $\sigma'(id_t)(\texttt{fired}) = \texttt{true}$.
     By definition of id_t, there exist a g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$.

By property of the stabilize relation and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `fired_evaluation` process defined in the `transition` design architecture, we can deduce

$$\sigma(id_t)(\texttt{fired}) = \sigma(id_t)(\texttt{sfa}) \ . \ \sigma(id_t)(\texttt{spc}) = \texttt{true}$$

Thus, we have

$$\sigma(id_t)(\texttt{sfa}) = \texttt{true}$$

and, appealing to Lemma 39, we can deduce $t \in Firable(s')$, which directly contradicts $t \notin Firable(s')$.

- **CASE** FSetNotSens: Assuming
  - $t \notin Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i))$
  - $IsFiredSetAux(s', T_s, F, Fset)$
  - $\nexists t' \in T_s \ s.t. \ t' \succ t$
  - $Pr(t, F) = \{t' \mid t' \succ t \wedge t' \in F\}$

  and the induction hypothesis (i.e. $P(s', T_s, F, Fset)$)

$$\begin{array}{l}\big(\forall t' \in T,\ id_{t'} \in Comps(\Delta) \text{ s.t. } \gamma(t') = id_{t'},\\ (t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true})\\ \wedge\ (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s)\big) \Rightarrow\\ t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}\end{array}$$

  we must show

$$\begin{array}{l}\big(\forall t' \in T,\ id_{t'} \in Comps(\Delta) \text{ s.t. } \gamma(t') = id_{t'},\\ (t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true})\\ \wedge\ (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s \cup \{t\})\big) \Rightarrow\\ t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}\end{array}$$

  Assuming the following hypothesis, which we will call EH (for Extra Hypothesis)

$$\begin{array}{l}\forall t' \in T,\ id_{t'} \in Comps(\Delta) \text{ s.t. } \gamma(t') = id_{t'},\\ (t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s \cup \{t\})\end{array}$$

  we must show

$$t \in Fset \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{true}$$

  Appealing to the induction hypothesis, to prove the current goal, it is sufficient to prove that

$$\forall t' \in T,\ id_{t'} \in Comps(\Delta) \text{ s.t. } \gamma(t') = id_{t'},$$
$$(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s)$$

Given a $t' \in T$, an $id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$, we must show that

$$(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s)$$

There are two points to prove

1. Assuming $t' \in F$, then $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$
2. Assuming $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$, then $t' \in F \ \vee \ t' \in T_s$

1. Assuming $t' \in F$, let us show $\boxed{\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}}$.
   Appealing to EH, the goal is trivially shown.
2. Assuming $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$, let us show $\boxed{t' \in F \ \vee \ t' \in T_s}$.
   Appealing to EH, we can deduce $t' \in F \ \vee \ t' \in T_s \cup \{t\}$. Let us perform case analysis on $t' \in F \ \vee \ t' \in T_s \cup \{t\}$; there are 2 cases:
   - **CASE** $t' \in F$: trivially shown, as it is an assumption.
   - **CASE** $t' \in T_s \cup \{t\}$: In the case where $t' \in T_s$, the goal is trivially shown. In the case where $t' = t$, we can prove a contradiction based on $t \notin Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i))$ and $\sigma'(id_{t'})(\texttt{fired}) = \texttt{true}$.
     Since $t = t'$, then $id_t = id_{t'}$, and we know that $\sigma'(id_t)(\texttt{fired}) = \texttt{true}$.
     By definition of id_t, there exist a g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$.
     By property of the stabilize relation and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `fired_evaluation` process defined in the `transition` design architecture, we can deduce

     $$\sigma(id_t)(\texttt{fired}) = \sigma(id_t)(\texttt{sfa}) \ . \ \sigma(id_t)(\texttt{spc}) = \texttt{true}$$

     Thus, we have

     $$\sigma(id_t)(\texttt{spc}) = \texttt{true}$$

     and, appealing to Lemma 45, we can deduce $t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i))$, which directly contradicts $t \notin Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i))$.

□

Lemma 45 (Stabilize compute priority combination after falling edge)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $\forall T_s, F$, $Fset \subseteq T$ assume that:*

- $t \in Firable(s')$
- $\nexists t' \in T_s$ *s.t.* $t' \succ t$
- *EH:* $\forall t' \in T, id_{t'} \in Comps(\Delta)$ *s.t.* $\gamma(t') = id_{t'}$,
  $(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{\textit{fired}}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{\textit{fired}}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s)$.

then $t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i)) \Leftrightarrow \sigma'(id_t)(\texttt{\textit{spc}}) = \texttt{true}$

Proof.

Given a $t \in T$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, a T_s, F, $Fset \subseteq T$ and assuming

- $t \in Firable(s')$
- $\nexists t' \in T_s$ *s.t.* $t' \succ t$
- EH: $\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
  $(t' \in F \Rightarrow \sigma'(id_{t'})(\texttt{fired}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\texttt{fired}) = \texttt{true} \Rightarrow t' \in F \ \vee \ t' \in T_s)$.

let us show

$$t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i)) \Leftrightarrow \sigma'(id_t)(\texttt{spc}) = \texttt{true}.$$

By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$. By property of the stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$, and through the examination of the `priority_authorization_evaluation` process defined in the `transition` design architecture, we can deduce:

$$\sigma'(id_t)(\texttt{spc}) = \prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{pauths})[i]$$

Rewriting the goal with the above equation:

$$t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i)) \Leftrightarrow \prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}.$$

Then, the proof is in two parts:

1. $t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i)) \Rightarrow \prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}$

2. $\prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{pauths})[i] = \texttt{true} \Rightarrow t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i))$

Let us prove both sides of the equivalence:

1. Assuming that $t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i))$, let us show

   $\boxed{\prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}.}$

   Let us perform case analysis on $input(t)$; there are 2 cases:

   - **CASE** $input(t) = \emptyset$:
     By construction, $<\texttt{input\_arcs\_number} \Rightarrow 1> \in g_t$ and $<\texttt{priority\_authorizations(0)} \Rightarrow \texttt{true}> \in i_t$.
     By property of the elaboration relation, we have $\Delta(id_t)(\texttt{ian}) = 1$, and by property of the stabilize relation, we have $\sigma'(id_t)(\texttt{pauths})[0] = \texttt{true}$.
     Rewriting the goal with $\Delta(id_t)(\texttt{ian}) = 1$ and $\sigma'(id_t)(\texttt{pauths})[0] = \texttt{true}$, and simplifying the goal: tautology.
   - **CASE** $input(t) \neq \emptyset$:
     Then, let us show an equivalent goal:
     $\boxed{\forall i \in [0, \Delta(id_t)(\texttt{ian}) - 1],\ \sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}.}$
     Given an $i \in [0, \Delta(id_t)(\texttt{ian}) - 1]$, let us show $\boxed{\sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}.}$
     By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in g_t$.
     By property of the elaboration relation, we have $\Delta(id_t)(\texttt{ian}) = |input(t)|$. Then, we can deduce $i \in [0, |input(t)| - 1]$.
     By construction, for all $i \in [0, |input(t)| - 1]$, there exist a $p \in input(t)$ and an $id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$, there exist a g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and there exist a $j \in [0, |output(p)|]$ and an $id_{ji} \in Sigs(\Delta)$ s.t.
     $<\texttt{input\_arcs\_valid(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in i_t$ and $<\texttt{output\_arcs\_valid(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in o_t$. Let us take such a $p \in input(t)$, $id_p \in Comps(\Delta)$, g_p, i_p, o_p, $j \in [0, |output(p)|]$ and $id_{ji} \in Sigs(\Delta)$.

     Now, let us perform case analysis on the nature of the arc connecting p and t; there are 2 cases:

     – **CASE** $pre(p,t) = (\omega, \texttt{test})$ or $pre(p,t) = (\omega, \texttt{inhib})$:
       By construction, $<\texttt{priority\_authorizations(i)} \Rightarrow \texttt{true}> \in i_t$, and by property of the stabilize relation: $\sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}.$
     – **CASE** $pre(p,t) = (\omega, \texttt{basic})$:
       Let us define $output_c(p) = \{t \in T \mid \exists \omega,\ pre(p,t) = (\omega, \texttt{basic})\}$, the set of output transitions of p that are in conflict. Then, there are two cases, one for each way to solve the conflicts between the output transitions of p:
       * **CASE** For all pair of transitions in $output_c(p)$, all conflicts are solved by mutual exclusion:
         By construction, $<\texttt{priority\_authorizations(i)} \Rightarrow \texttt{true}> \in i_t$, and by property of the stabilize relation: $\sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}.$

$*$ **CASE** The priority relation is a strict total order over the set $output_c(p)$:

By construction, there exists an $id'_{ji} \in Sigs(\Delta)$ s.t.
<$\texttt{priority\_authorizations(i)} \Rightarrow \texttt{id}'_{\texttt{ji}}$> $\in i_t$ and
<$\texttt{priority\_authorizations(j)} \Rightarrow \texttt{id}'_{\texttt{ji}}$> $\in o_p$.
By property of the stabilize relation, $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$ and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we can deduce:

$$\sigma'(id_t)(\texttt{pauths})[i] = \sigma'(id'_{ji}) = \sigma'(id_p)(\texttt{pauths})[j]$$

Rewriting the goal with the above equation: $\boxed{\sigma'(id_p)(\texttt{pauths})[j] = \texttt{true}.}$
By property of the stabilize relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, and through the examination of the `priority_evaluation` process defined in the `place` design behavior, we can deduce:

$$\sigma'(id_p)(\texttt{pauths})[j] = (\sigma'(id_p)(\texttt{sm}) \geq \texttt{vsots} + \sigma'(id_p)(\texttt{oaw})[j]) \tag{48}$$

Let us define the `vsots` term as follows:

$$\texttt{vsots} = \sum_{i=0}^{j-1} \begin{cases} \sigma'(id_p)(\texttt{oaw})[i] \texttt{ if } \sigma'(id_p)(\texttt{otf})[i]. \\ \qquad\qquad\qquad\qquad \sigma'(id_p)(\texttt{oat})[i] = \texttt{basic} \\ 0 \textit{ otherwise} \end{cases} \tag{49}$$

Rewriting the goal with (48): $\boxed{\sigma'(id_p)(\texttt{sm}) \geq \texttt{vsots} + \sigma'(id_p)(\texttt{oaw})[j]}$
By definition of $t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i))$, we can deduce:
$s'.M(p) \geq \sum_{t_i \in Pr(t,F)} pre(p, t_i) + \omega$.
Then, there are three points to prove:
(a) $\boxed{s'.M(p) = \sigma'(id_p)(\texttt{sm})}$
(b) $\boxed{\omega = \sigma'(id_p)(\texttt{oaw})[j]}$
(c) $\boxed{\sum_{t_i \in Pr(t,F)} pre(p, t_i) = \texttt{vsots}}$
Let us prove these three points:
(a) $\boxed{s'.M(p) = \sigma'(id_p)(\texttt{sm})}$
Appealing to Lemma 32, $s'.M(p) = \sigma'(id_p)(\texttt{sm})$.
(b) $\boxed{\omega = \sigma'(id_p)(\texttt{oaw})[j]}$
By construction, and as $pre(p, t) = (\omega, \texttt{basic})$, we know that <$\texttt{output\_arcs\_weights(j)} \Rightarrow \omega$> $\in i_p$.
By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$:
$\omega = \sigma'(id_p)(\texttt{oaw})[j]$.
(c) $\boxed{\sum_{t_i \in Pr(t,F)} pre(p, t_i) = \texttt{vsots}}$
Let us replace the left and right term of the equality by their full definition:

$$\sum_{t_i \in Pr(t,F)} \begin{cases} \omega \texttt{ if } pre(p,t_i) = (\omega, \texttt{basic}) \\ 0 \textit{ otherwise} \end{cases} = \sum_{i=0}^{j-1} \begin{cases} \sigma'(id_p)(\texttt{oaw})[i] \texttt{ if } \sigma'(id_p)(\texttt{otf})[i]. \\ \qquad\qquad\qquad\qquad \sigma'(id_p)(\texttt{oat})[i] = \texttt{basic} \\ 0 \textit{ otherwise} \end{cases}$$

Now, we must reason on the priority status of transition t regarding the group of conflicting output transitions of p. There 2 cases:

* **CASE** t is the top-priority transition in the group of conflicting output transitions of p: In that case, the set $Pr(t,F)$ is empty and, by construction, $j = 0$. Thus, the goal is a tautology $0 = 0$.
* **CASE** t is not the top-priority transition in the group of conflicting output transitions of p:

In that case, we know that there is a least one element in $Pr(t,F)$ and the index $j > 0$.
Let us replace the sum terms in the goal by equivalent terms:

$$\sum_{t_i \in Pr_p} \begin{cases} \omega \texttt{ if } pre(p,t_i) = (\omega, \texttt{basic}) \texttt{ and } t_i \in F \\ 0 \textit{ otherwise} \end{cases} = \sum_{i \in IPr_p} \begin{cases} \sigma'(id_p)(\texttt{oaw})[i] \texttt{ if } \sigma'(id_p)(\texttt{otf})[i] \\ 0 \textit{ otherwise} \end{cases}$$

Let us define the set Pr_p as

$$Pr_p = \{t_i \mid t_i \succ t \wedge \exists \omega \textit{ s.t. } pre(p,t_i) = (\omega, \texttt{basic})\}$$

and set IPr_p as

$$IPr_p = \{i \mid i \in [0, j-1] \wedge \sigma'(id_p)(\texttt{oat})[i] = \texttt{basic}\}$$

Let us define $f(t_i)$ as

$$f(t_i) = \begin{cases} \omega \texttt{ if } pre(p,t_i) = (\omega, \texttt{basic}) \texttt{ and } t_i \in F \\ 0 \textit{ otherwise} \end{cases}$$

and $g(i)$ as

$$g(i) = \begin{cases} \sigma'(id_p)(\texttt{oaw})[i] \texttt{ if } \sigma'(id_p)(\texttt{otf})[i] \\ 0 \textit{ otherwise} \end{cases}$$

then, we must prove $\boxed{\sum_{t_i \in Pr_p} f(t_i) = \sum_{i \in IPr_p} g(i).}$

To prove the above equality, it is sufficient to prove that there exists a bijection β from Pr_p to IPr_p such that for all $t_i \in Pr_p$, $f(t_i) = g(\beta(t_i))$. Let us use the function β that takes a $t_i \in Pr_p$ and yields the index denoting the position of t_i in the priority-ordered version of set Pr_p. We assumed that a total order existed over the conflicting output transitions of place p, then there exists a total ordering of the transitions of set Pr_p, i.e. the conflicting output transitions of

place p with a higher priority than t. By property of the HILECOP transformation function, we know that the index returned by the function β belongs to the interval $[0, j-1]$ and verifies $\sigma'(id_p)(\texttt{oat})[i] = \texttt{basic}$. Given a $t_i \in Pr_p$, we must show $\boxed{f(t_i) = g(\beta(t_i)).}$
Let us unfold terms $f(t_i)$ and $g(\beta(t_i))$ to their full definition:

$$\begin{cases} \omega \texttt{ if } pre(p,t_i) = (\omega, \texttt{basic}) \texttt{ and } t_i \in F \\ 0 \textit{ otherwise} \end{cases}$$
$$=$$
$$\begin{cases} \sigma'(id_p)(\texttt{oaw})[\beta(t_i)] \texttt{ if } \sigma'(id_p)(\texttt{otf})[\beta(t_i)] \\ 0 \textit{ otherwise} \end{cases}$$

By construction, there exists an $id_{t_i} \in Comps(\Delta)$ such that $\gamma(t_i) = id_{t_i}$, and there exist g_{t_i}, i_{t_i} and o_{t_i} such that $\texttt{comp}(id_{t_i}, \texttt{transition}, g_{t_i}, i_{t_i}, o_{t_i}) \in d.cs$.

By property of the function β and by construction, we can deduce that the element of index $\beta(t_i)$ of the `otf` input port of PCI id_p is connected the `fired` output port of TCI id_{t_i}. Thus, there exists an $id_{\beta i} \in Sigs(\Delta)$ s.t. $<\texttt{otf}(\beta(t_i)) \Rightarrow \texttt{id}_{\beta i}> \in i_p$ and
$<\texttt{fired} \Rightarrow \texttt{id}_{\beta i}> \in o_{t_i}$.
By property of the stabilize relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$ and
$\texttt{comp}(id_{t_i}, \texttt{transition}, g_{t_i}, i_{t_i}, o_{t_i}) \in d.cs$, we have

$$\sigma'(id_{t_i})(\texttt{fired}) = \sigma'(id_{\beta i}) = \sigma'(id_p)(\texttt{otf})[\beta(t_i)]$$

then, we can rewrite the goal with the above equation

$$\begin{cases} \omega \texttt{ if } pre(p,t_i) = (\omega, \texttt{basic}) \texttt{ and } t_i \in F \\ 0 \textit{ otherwise} \end{cases}$$
$$=$$
$$\begin{cases} \sigma'(id_p)(\texttt{oaw})[\beta(t_i)] \texttt{ if } \sigma'(id_{t_i})(\texttt{fired}) \\ 0 \textit{ otherwise} \end{cases}$$

By property of the function β and by construction, we can deduce that the element of index $\beta(t_i)$ of the `oaw` input port of PCI id_p is connected to a constant value denoting the weight of the arc between place p and transition t_i. Thus, we have

$$<\texttt{oaw}(\beta(t_i)) \Rightarrow \omega> \in i_p \text{ where } pre(p,t_i) = (\omega, \texttt{basic})$$

By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$, we have

$$\sigma'(id_p)(\texttt{oaw})[\beta(t_i)] = \omega$$

then, we can rewrite the goal with the above equation

$$\begin{cases} \omega \texttt{ if } pre(p,t_i) = (\omega, \texttt{basic}) \texttt{ and } t_i \in F \\ 0 \textit{ otherwise} \end{cases}$$
$$=$$
$$\begin{cases} \omega \texttt{ if } \sigma'(id_{t_i})(\texttt{fired}) \\ 0 \textit{ otherwise} \end{cases}$$

Finally, proving the goal comes down to proving

$$t_i \in F \Leftrightarrow \sigma'(id_{t_i})(\texttt{fired}) = \texttt{true}$$

Let us prove both sense of the equivalence:

(a) Assuming $t_i \in F$, let us show $\boxed{\sigma'(id_{t_i})(\texttt{fired}) = \texttt{true}}$.
Appealing to EH, proving the goal is trivial.

(b) Assuming $\sigma'(id_{t_i})(\texttt{fired}) = \texttt{true}$, let us show $\boxed{t_i \in F}$.
Appealing to EH, we have $t_i \in F \vee t_i \in T_s$. There are two cases: either $t_i \in F$ or $t_i \in T$. In the case where $t_i \in T$, we can show a contradiction with the fact that t is a top-priority transition in set T_s. By definition, transition t_i has a higher firing priority than t, and thus, if t_i belongs to set T_s, then t is no longer a top-priority transition of set T_s; whence the contradiction.

2. Assuming that $\prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}$, let us show

$$\boxed{t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i)).}$$

By definition of $t \in Sens(s'.M - \sum_{t_i \in Pr(t,F)} pre(t_i))$:

$$\begin{array}{l} \forall p \in P, \omega \in \mathbb{N}^*, \\ \big((pre(p,t) = (\omega, \texttt{basic}) \vee pre(p,t) = (\omega, \texttt{test})) \Rightarrow s'.M(p) - \sum_{t_i \in Pr(t,F)} pre(p,t_i) \geq \omega\big) \\ \wedge \ \big(pre(p,t) = (\omega, \texttt{inhib}) \Rightarrow s'.M(p) - \sum_{t_i \in Pr(t,F)} pre(p,t_i) < \omega\big) \end{array}$$

Given a $p \in P$ and an $\omega \in \mathbb{N}^*$, let us show

$$\begin{array}{l} \big((pre(p,t) = (\omega, \texttt{basic}) \vee pre(p,t) = (\omega, \texttt{test})) \Rightarrow s'.M(p) - \sum_{t_i \in Pr(t,F)} pre(p,t_i) \geq \omega\big) \\ \wedge \ \big(pre(p,t) = (\omega, \texttt{inhib}) \Rightarrow s'.M(p) - \sum_{t_i \in Pr(t,F)} pre(p,t_i) < \omega\big) \end{array}$$

By construction, there exists an $id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$. By construction and by definition of id_p, there exist g_p, i_p, o_p s.t. $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$.

To prove the goal, there are different cases:

(a) Assuming that $pre(p,t) = (\omega, \texttt{test})$, let us show $\boxed{s'.M(p) - \sum_{t_i \in Pr(t,F)} pre(p,t_i) \geq \omega.}$

Then, assuming that the priority relation is well-defined, there exists no transition t_i connected by a `basic` arc to p that verifies $t_i \succ t$. This is because t is connected to p by a `test` arc; thus, t is not in conflict with the other output transitions of p; thus, there is no relation of priority between t and the other output transitions of p.

Then, we can deduce that $\sum_{t_i \in Pr(t,F)} pre(p,t_i) = 0$.

Then, the new goal is $s'.M(p) \geq \omega$.

Knowing that $t \in Firable(s')$, thus, $t \in Sens(s'.M)$, thus, we have $s'.M(p) \geq \omega$.

(b) Assuming that $pre(p,t) = (\omega, \texttt{inhib})$, let us show $\boxed{s'.M(p) - \sum_{t_i \in Pr(t,F)} pre(p,t_i) < \omega.}$

Use the same strategy as above.

(c) Assuming that $pre(p,t) = (\omega, \texttt{basic})$, let us show $\boxed{s'.M(p) - \sum_{t_i \in Pr(t,F)} pre(p,t_i) \geq \omega.}$

Then, there are two cases:

i. **CASE** For all pair of transitions in $output_c(p)$, all conflicts are solved by mutual exclusion.
Then, assuming that the priority relation is well-defined, it must not be defined over the set $output_c(t)$, and we know that $t \in output_c(p)$ since $pre(p,t) = (\omega, \texttt{basic})$.
Then, there exists no transition t_i connected to p by a `basic` arc that verifies $t_i \succ t$.
Then, we can deduce $\sum_{t_i \in Pr(t,F)} pre(p,t_i) = 0$.
Then, the new goal is $s'.M(p) \geq \omega$.
We know $t \in Firable(s')$, thus, $t \in Sens(s'.M)$, thus, $s'.M(p) \geq \omega$.

ii. **CASE** The priority relation is a strict total order over the set $output_c(p)$.
By construction, there exists $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$. By construction and by definition of id_t, there exist g_t, i_t, o_t s.t. $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$.
By construction, there exist $j \in [0, |input(t)|-1]$, $k \in [0, |output(t)|-1]$, and $id_{kj} \in Sigs(\Delta)$ s.t. $<\texttt{priority\_authorizations(j)} \Rightarrow \texttt{id}_{\texttt{kj}}> \in i_t$ and
$<\texttt{priority\_authorizations(k)} \Rightarrow \texttt{id}_{\texttt{kj}}> \in o_p$. Let us take such an j, k and id_{kj}.
From $\prod_{i=0}^{\Delta(id_t)(\texttt{ian})-1} \sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}$, we can deduce that for all $i \in [0, \Delta(id_t)(\texttt{ian})-1]$, $\sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}$.
By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in g_t$, and by property of the elaboration relation, we have $\Delta(id_t)(\texttt{ian}) = |input(t)|$. Then, from $j \in [0, |input(t)|-1]$, we can deduce $j \in [0, \Delta(id_t)(\texttt{ian})-1]$. And, from $\forall i \in [0, \Delta(id_t)(\texttt{ian})-1]$, $\sigma'(id_t)(\texttt{pauths})[i] = \texttt{true}$, we can deduce $\sigma'(id_t)(\texttt{pauths})[j] = \texttt{true}$.
By property of the stabilize relation, $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$ and $\texttt{comp}(id_t, \texttt{transition}, g_t, i_t, o_t) \in d.cs$:

$$\sigma'(id_p)(\texttt{pauths})[k] = \sigma'(id_{kj}) = \sigma'(id_t)(\texttt{pauths})[j] = \texttt{true} \tag{50}$$

By property of the stabilize relation and $\texttt{comp}(id_p, \texttt{place}, g_p, i_p, o_p) \in d.cs$:

$$\sigma'(id_p)(\texttt{pauths})[k] = (\sigma'(id_p)(\texttt{sm}) \geq \texttt{vsots} + \sigma'(id_p)(\texttt{oaw})[k]) \tag{51}$$

Let us define the `vsots` term as follows:

$$\texttt{vsots} = \sum_{i=0}^{k-1} \begin{cases} \sigma'(id_p)(\texttt{oaw})[i] \text{ if } \sigma'(id_p)(\texttt{otf})[i]. \\ \qquad\qquad\qquad\qquad \sigma'(id_p)(\texttt{oat})[i] = \texttt{basic} \\ 0 \ otherwise \end{cases} \tag{52}$$

From (50) and (51), we can deduce that $\sigma'(id_p)(\texttt{sm}) \geq \texttt{vsots} + \sigma'(id_p)(\texttt{oaw})[k]$.
Then, there are three points to prove:

A. $\boxed{s'.M(p) = \sigma'(id_p)(\texttt{sm})}$

B. $\boxed{\omega = \sigma'(id_p)(\texttt{oaw})[k]}$

C. $\boxed{\sum_{t_i \in Pr(t,F)} pre(p, t_i) = \texttt{vsots}}$

See 1 for the remainder of the proof.

□

Lemma 46 (Falling edge equal not fired)**.** *For all sitpn, b, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, $\sigma_\downarrow$, σ' that verify the hypotheses of Definition 13, then $\forall t, id_t$ s.t. $\gamma(t) = id_t$, $t \notin Fired(s') \Leftrightarrow \sigma'(id_t)(\texttt{fired}) = \texttt{false}$.*

Proof.

Proving the above lemma is trivial by appealing to Lemma **??** and by reasoning on contrapositives.

□