UNIVERSITY NAME

DOCTORAL THESIS

Thesis Title

Author:
John SMITH

Supervisor:
Dr. James SMITH

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Research Group Name
Department or School Name

May 4, 2021

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSITY NAME

Abstract

Faculty Name
Department or School Name

Doctor of Philosophy

Thesis Title

by John SMITH

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

For/Dedicated to/To my…

Chapter 1

Proving semantic preservation in HILECOP

- Change σ_{injr} and σ_{injf} into σ_i.
- Define the $\texttt{Inject}_{\downarrow}$ and $\texttt{Inject}_{\uparrow}$ relations.
- Keep the *sitpn* argument in the SITPN full execution relation, but remove it from the SITPN execution, cycle and state transition relations.
- Make a remark on the differentiation of boolean operators and intuitionistic logic operators
- Explain and illustrate the equivalence relation between SITPN and VHDL.

1.1 Preliminary Definitions

Definition 1 (SITPN-to-$\mathcal{H}$-VHDL Design Binder)**.** *Given a $sitpn \in SITPN$ and a $\mathcal{H}$-VHDL design $d \in design$, a SITPN-to-$\mathcal{H}$-VHDL design binder $\gamma \in WM(sitpn, d)$ is a tuple $<PMap, TMap, \mathcal{C}_{id}, \mathcal{A}_{id}, \mathcal{F}_{id}, CMap, AMap, FMap>$ where:*

- $sitpn = <P, T, pre, test, inhib, post, M_0, \succ, \mathcal{A}, \mathcal{C}, \mathcal{F}, \mathbb{A}, \mathbb{C}, \mathbb{F}, I_s>$
- $d = \texttt{design}\ id_{ent}\ id_{arch}\ gens\ ports\ sigs\ behavior$
- $PMap \in P \rightarrow P_{id}$ *where* $P_{id} = \{id \mid \texttt{comp}(id, "place", gm, ipm, opm) \in behavior\}$
- $TMap \in T \rightarrow T_{id}$ *where* $T_{id} = \{id \mid \texttt{comp}(id, "transition", gm, ipm, opm) \in behavior\}$
- $\mathcal{C}_{id} \subseteq \{id \mid (\texttt{in}, id, t) \in ports \wedge id \notin \{"clk", "rst"\}\}$
- $\mathcal{A}_{id} \subseteq \{id \mid (\texttt{out}, id, t) \in ports\}$
- $\mathcal{F}_{id} \subseteq \{id \mid (\texttt{out}, id, t) \in ports\}$
- $CMap \in \mathcal{C} \rightarrow \mathcal{C}_{id}$
- $AMap \in \mathcal{A} \rightarrow \mathcal{A}_{id}$
- $FMap \in \mathcal{F} \rightarrow \mathcal{F}_{id}$

Definition 2 (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 $\Delta \in ElDesign(d, \mathcal{D})$ of design d, and a binder $\gamma \in WM(sitpn, d)$, the environment $E_p \in (\mathbb{N} \times \{\uparrow, \downarrow\}) \rightarrow Ins(\Delta) \rightarrow value$, that yields the value of the primary input ports of Δ at a given simulation cycle and a given clock event, and the environment E_c, that yields the value of conditions of $sitpn$ at a given execution cycle, are similar, noted $\gamma \vdash E_p \stackrel{env}{=} E_c$, iff for all $\tau \in \mathbb{N}$, $clk \in \{\uparrow, \downarrow\}$, $c \in \mathcal{C}$, $id_c \in Ins(\Delta)$ s.t. $\gamma(c) = id_c$, $E_p(\tau, clk)(id_c) = E_c(\tau)(c)$.*

1.1.1 State Similarity

Definition 3 (General State 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 $\gamma \in WM(sitpn, d)$, an SITPN state $s \in S(sitpn)$ and a design state $\sigma \in \Sigma(\Delta)$ are similar, written $\gamma \vdash s \sim \sigma$ iff*

1. $\forall p \in P, id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p,\ s.M(p) = \sigma(id_p)(''s\_marking'')$.
2. $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,
$\big(upper(I_s(t)) = \infty \wedge s.I(t) \leq lower(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(''s\_time\_counter'')\big)$
$\wedge\big(upper(I_s(t)) = \infty \wedge s.I(t) > lower(I_s(t)) \Rightarrow \sigma(id_t)(''s\_time\_counter'') = lower(I_s(t))\big)$
$\wedge\big(upper(I_s(t)) \neq \infty \wedge s.I(t) > upper(I_s(t)) \Rightarrow \sigma(id_t)(''s\_time\_counter'') = upper(I_s(t))\big)$
$\wedge\big(upper(I_s(t)) \neq \infty \wedge s.I(t) \leq upper(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(''s\_time\_counter'')\big)$.
3. $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, s.reset_t(t) = \sigma(id_t)(''s\_reinit\_time\_counter'')$.
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 4 (Post Rising Edge State 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 $\gamma \in WM(sitpn, d)$, a clock cycle count $\tau \in \mathbb{N}$, and an SITPN execution environment $E_c \in \mathbb{N} \rightarrow \mathcal{C} \rightarrow \mathbb{B}$, an SITPN state $s \in S(sitpn)$ and a design state $\sigma \in \Sigma(\Delta)$ are similar after a rising edge happening at clock cycle count τ, written $\gamma, E_c, \tau \vdash s \stackrel{\uparrow}{\sim} \sigma$ iff*

1. $\forall p \in P, id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p,\ s.M(p) = \sigma(id_p)(''s\_marking'')$.
2. $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,
$\big(upper(I_s(t)) = \infty \wedge s.I(t) \leq lower(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(''s\_time\_counter'')\big)$
$\wedge\big(upper(I_s(t)) = \infty \wedge s.I(t) > lower(I_s(t)) \Rightarrow \sigma(id_t)(''s\_time\_counter'') = lower(I_s(t))\big)$
$\wedge\big(upper(I_s(t)) \neq \infty \wedge s.I(t) > upper(I_s(t)) \Rightarrow \sigma(id_t)(''s\_time\_counter'') = upper(I_s(t))\big)$
$\wedge\big(upper(I_s(t)) \neq \infty \wedge s.I(t) \leq upper(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(''s\_time\_counter'')\big)$.
3. $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, s.reset_t(t) = \sigma(id_t)(''s\_reinit\_time\_counter'')$.
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. $\gamma(t) = id_t, t \in Sens(s.M) \Leftrightarrow \sigma(id_t)(''s\_enabled'') = \texttt{true}$.

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

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

Definition 5 (Post Falling Edge State 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 $\gamma \in WM(sitpn, d)$, an SITPN state $s \in S(sitpn)$ and a design state $\sigma \in \Sigma(\Delta)$ are similar after a falling edge, written $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$ iff $\gamma \vdash s \sim \sigma$ (Def. 3, general state similarity) and*

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

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

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

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

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

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

Definition 6 (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 $\theta_\sigma \in \texttt{list}(\Sigma(\Delta))$ are similar, written $\gamma \vdash \theta_s \sim \theta_\sigma$, according to the following rules:*

$$\text{SIMTRACENIL}\ \frac{}{\gamma \vdash [\,] \sim [\,]} \qquad \text{SIMTRACECONS}\ \frac{\gamma \vdash s \sim \sigma \qquad \gamma \vdash \theta_s \sim \theta_\sigma}{\gamma \vdash (s :: \theta_s) \sim (\sigma :: \theta_\sigma)}$$

1.1.2 Equality between big operator expressions

Many times in the proceeding of the following proof, the equality between two sum or product expressions must be estbalished; for instance:
$\sum_{a \in A} f(a) = \sum_{b \in B} g(b)$ where A and B are finite sets, $f \in \mathbb{A} \to \mathbb{N}$ and $g \in B \to \mathbb{N}$
To prove such an equality, Theorem 1 is used, considering that the sum operator used in the above equation is a big operator over the triplet $<\mathbb{N}, 0, +>$. A big operator is defined as follows:

Definition 7 (Big Operator). *Given a triplet $<A, *, e>$ such that A is a set, $* \in A \to A \to A$ is a commutative and associative operator over A, and $e \in A$ is a neutral element of $*$, then for all finite set B, and application $f \in B \to A$, a big operator Ω is recursively defined as follows: $\underset{b \in B}{\Omega} f(b) =$*
$$\begin{cases} e & if\ B = \emptyset \\ f(b) * \underset{b' \in B \setminus \{b\}}{\Omega} f(b') & otherwise \end{cases}$$

Then, we can prove the following theorem concerning the equality between two big operator expressions.

Theorem 1 (Big Operator Equality). *For all a triplet* $<A, *, e>$ *such that* A *is a set,* $* \in A \to A \to A$ *is a commutative and associative operator over* A*, and* $e \in A$ *is a neutral element of* $*$*, and for all finite sets* B *and* C*, and applications* $f \in B \to A$ *and* $g \in C \to A$*, assume that:*

- *there exists an injection* $\iota \in B \to C$ *s.t.* $\forall b \in B,\ f(b) = g(\iota(b))$
- $|B| = |C|$

then $\underset{b \in B}{\Omega} f(b) = \underset{c \in C}{\Omega} g(c)$.

Proof. Let us reason by induction over $\underset{b \in B}{\Omega} f(b)$:

- **BASE CASE** $B = \emptyset$:
  Then $|C| = |B| = 0$, and $C = \emptyset$. By definition of Ω:

$$\underset{b \in B}{\Omega} f(b) = e \tag{1.1}$$
$$\underset{c \in C}{\Omega} g(c) = e \tag{1.2}$$

  Rewriting the goal with (1.1) and (1.2), tautology.

- **INDUCTION CASE** $B \neq \emptyset$:

  For all finite set C' verifying:
  - $\exists$ an injection $\iota' \in B \setminus \{b\} \to C'$ s.t. $\forall b' \in B \setminus \{b\},\ f(b') = g(\iota(b'))$
  - $|B \setminus \{b\}| = |C'|$

  then $f(b) * \underset{b' \in B \setminus \{b\}}{\Omega} f(b') = f(b) * \underset{c' \in C'}{\Omega} g(c)$

  The goal is $\boxed{f(b) * \underset{b' \in B \setminus \{b\}}{\Omega} f(b') = \underset{c \in C}{\Omega} g(c)}$

  Let us take $\iota \in B \to C$ s.t. $\forall b \in B,\ f(b) = g(\iota(b))$, then:

$$f(b) = g(\iota(b)) \tag{1.3}$$

  Also, by definition of Ω:

$$\underset{c \in C}{\Omega} g(c) = g(\iota(b)) * \underset{c' \in C \setminus \{\iota(b)\}}{\Omega} \tag{1.4}$$

  Rewriting the goal with (1.4) and (1.3),
  $\boxed{f(b) * \underset{b' \in B \setminus \{b\}}{\Omega} f(b') = f(b) * \underset{c' \in C \setminus \{\iota(b)\}}{\Omega} g(c')}$

  Let us apply the induction hypothesis with $C' = C \setminus \{\iota(b)\}$; then there are two points to prove:

  1. $\boxed{|B \setminus \{b\}| = |C \setminus \{\iota(b)\}|.}$ Trivial as $|B| = |C|$.
  2. $\boxed{\exists \text{ an injection } \iota' \in B \setminus \{b\} \to C \setminus \{\iota(b)\} \text{ s.t. } \forall b' \in B \setminus \{b\}, f(b') = g(\iota'(b'))}$

Let us define a $\iota' \in B \setminus \{b\} \rightarrow C \setminus \{\iota(b)\}$ as follows: $\forall b' \in B \setminus \{b\},\ \iota'(b) = \iota(b)$. Let us show that this definition is correct by proving that
$\boxed{\forall b' \in B \setminus \{b\},\ \iota(b') \in C \setminus \{\iota(b)\}.}$

Given a $b' \in B \setminus \{b\}$, let us show $\boxed{\iota(b') \in C \setminus \{\iota(b)\}.}$

By definition of ι, $\iota(b') \in C$; then, there are 2 cases:

- **CASE** $\iota(b') = \iota(b)$, then by definition of ι as an injective function: $b' = b$. Then, $b \in B \setminus \{b\}$ is a contradicti
- **CASE** $\iota(b') \in C \setminus \{\iota(b)\}.$

Now let us get back to the previous goal. Using ι' to prove it, there are 2 points to prove:

- $\boxed{\iota' \text{ is injective.}}$ Trivial, by definition of ι'.
- $\boxed{\forall b' \in B \setminus \{b\},\ f(b') = g(\iota'(b')).}$ Trivial, by definition of ι'.

□

Add a remark on how to convert a sequence of indexes into a finite set, and what is the cardinality of the finite set:
$\underset{i=n}{\overset{m}{\Omega}} f(i)$ then $|[n,m]| = (m-n)+1$ when $m \geq n$

1.2 Behavior Preservation Theorem

1.2.1 Proof Notations

- Frame box for pending goals: $\boxed{\forall n \in \mathbb{N},\ n > 0 \vee n = 0}$
- Red frame box for completed goals: $\texttt{true} = \texttt{true}$
- Green frame box for induction hypotheses:

$\forall n \in \mathbb{N},\ n + 1 > 0$

- **CASE** to denote a case during a proof by case analysis.

Make a list of all signals and constants of the T and P components, and their related aliases.

Constants and signals reference			
Full name	*Alias*	*Category*	*Type*
$''input\_conditions''$	$''ic''$	input port (T)	$\mathbb{B}$
$''input\_conditions''$	$''ic''$	input port (T)	$\mathbb{B}$
$''reinit\_time''$	$''rt''$	input port (T)	$\mathbb{B}$
$''input\_arcs\_valid''$	$''iav''$	input port (T)	$\mathbb{B}$
$''fired''$	$''f''$	output port (T)	$\mathbb{B}$
$''s\_condition\_combination''$	$''scc''$	internal signal (T)	$\mathbb{B}$
$''s\_reinit\_time\_counter''$	$''srtc''$	internal signal (T)	$\mathbb{B}$
$''s\_priority\_combination''$	$''spc''$	internal signal (T)	$\mathbb{B}$
$''s\_fired''$	$''sf''$	internal signal (T)	$\mathbb{B}$
$''s\_firable''$	$''sfa''$	internal signal (T)	$\mathbb{B}$
$''s\_enabled''$	$''se''$	internal signal (T)	$\mathbb{B}$
$''input\_arcs\_number''$	$''ian''$	generic constant (T)	$\mathbb{N}$
$''transition\_type''$	$''tt''$	generic constant (T)	$\{\texttt{NOT\_TEMP}, \texttt{TEMP\_A\_B}, \texttt{TEMP\_A\_A}, \texttt{TEMP\_A\_INF}\}$
$''conditions\_number''$	$''cn''$	generic constant (T)	$\mathbb{N}$
$''maximal\_time\_counter''$	$''mtc''$	generic constant (T)	$\mathbb{N}$
$''s\_marking''$	$''sm''$	internal signal (P)	$\mathbb{N}$
$''s\_output\_token\_sum''$	$''sots''$	internal signal (P)	$\mathbb{N}$
$''s\_input\_token\_sum''$	$''sits''$	internal signal (P)	$\mathbb{N}$
$''reinit\_transition\_time''$	$''rtt''$	output port (P)	$\mathbb{B}$
$''output\_arcs\_types''$	$''oat''$	input port (P)	$\{\texttt{BASIC}, \texttt{TEST}, \texttt{INHIB}\}$
$''output\_arcs\_weights''$	$''oaw''$	input port (P)	$\mathbb{N}$
$''output\_transition\_fired''$	$''otf''$	input port (P)	$\mathbb{B}$
$''input\_arcs\_weights''$	$''iaw''$	input port (P)	$\mathbb{N}$
$''input\_transition\_fired''$	$''itf''$	input port (P)	$\mathbb{B}$

1.2.2 Behavior Preservation Theorem and Proof

Theorem 2 (Behavior Preservation). *For all* $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\tau \in \mathbb{N}$, $E_c \in \mathbb{N} \rightarrow \mathcal{C} \rightarrow \mathbb{B}$, $\theta_s \in \texttt{list}(S(sitpn))$ *s.t.*

- *SITPN* $sitpn$ *translates into design* d*:* $\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$
- *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 exists $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}})$ *s.t. for all* $E_p \in (\mathbb{N} \times \{\uparrow, \downarrow\}) \rightarrow Ins(\Delta) \rightarrow value$ *verifying*

- *Simulation/Execution environments are similar:* $\gamma \vdash E_p \overset{env}{=} E_c$.

then there exists $\theta_\sigma \in \texttt{list}(\Sigma(\Delta))$ *s.t.*

- *Under the HILECOP design store* $\mathcal{D}_{\mathcal{H}}$ *and with an empty generic constant dimensioning function, design* d *yields the simulation trace* θ_σ *after* τ *simulation cycles, starting from its initial state:* $\mathcal{D}_{\mathcal{H}}, \Delta, \emptyset, E_p, \tau \vdash \mathrm{d} \xrightarrow{full} \theta_\sigma$

- *Traces θ_s and θ_σ are similar:* $\theta_s \sim \theta_\sigma$

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

By definition of the $\mathcal{H}$-VHDL full simulation relation:
$\mathcal{D}_{\mathcal{H}}, \Delta, \emptyset, E_p, \tau \vdash \mathrm{d} \xrightarrow{full} \theta_\sigma \equiv \exists \sigma_e, \sigma_0 \in \Sigma(\Delta),\ \mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$ and $\mathcal{D}_{\mathcal{H}}, \Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$ and $\mathcal{D}_{\mathcal{H}}, E_p, \Delta, \tau, \sigma_0 \vdash \mathrm{d.cs} \rightarrow \theta_\sigma$.

Use Elaboration, Initialization and Simulation theorems to show that there exists a $\Delta, \theta_\sigma, \sigma_e$ and σ_0 such that $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$ and $\mathcal{D}_{\mathcal{H}}, \Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$ and $\mathcal{D}_{\mathcal{H}}, E_p, \Delta, \tau, \sigma_0 \vdash \mathrm{d.cs} \rightarrow \theta_\sigma$.

Use Full Bisimulation theorem to show traces similarity.

$\square$

Theorem 3 (Elaboration). *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$ s.t.*

- $\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$

then there exists $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}}), \sigma_e \in \Sigma(\Delta)$ s.t.

- $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$

Theorem 4 (Initialization). *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}})$, $\sigma_e \in \Sigma(\Delta)$ s.t.*

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

then there exists $\sigma_0 \in \Sigma(\Delta)$ s.t.

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

Theorem 5 (Simulation). *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}})$, $\sigma_e, \sigma_0 \in \Sigma(\Delta)$ s.t.*

- $\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$ *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 for all $E_p \in (\mathbb{N} \times \{\uparrow, \downarrow\}) \rightarrow Ins(\Delta) \rightarrow value$, $\tau \in \mathbb{N}$, there exists $\theta_\sigma \in \texttt{list}(\Sigma(\Delta))$ s.t.

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

1.2.3 Bisimulation Theorem and Proof

Theorem 6 (Full Bisimulation). *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\tau \in \mathbb{N}$, $E_c \in \mathbb{N} \rightarrow \mathcal{C} \rightarrow \mathbb{B}$, $\theta_s \in \texttt{list}(S(sitpn))$, $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}})$, $E_p \in (\mathbb{N} \times \{\uparrow, \downarrow\}) \rightarrow Ins(\Delta) \rightarrow value$, $\theta_\sigma \in \texttt{list}(\Sigma(\Delta))$ s.t.*

- $\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$

- $\gamma \vdash E_p \stackrel{env}{=} E_c$

- $E_c, \tau \vdash sitpn \xrightarrow{full} \theta_s$
- $\mathcal{D}_{\mathcal{H}}, \Delta, \emptyset, E_p, \tau \vdash \mathrm{d} \xrightarrow{full} \theta_\sigma$

then $\theta_s \sim \theta_\sigma$

Proof. Case analysis on τ (2 CASES).

- **CASE** $\tau = 0$. By definition of the SITPN full execution and the $\mathcal{H}$-VHDL full simulation relations:
  - $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$
  - $\Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$
  - $\theta_s = [s_0]$ and $\theta_\sigma = [\sigma_0]$

  $\boxed{\gamma \vdash s_0 \sim \sigma_0}$ (by def. of similar execution trace relation). Solved by applying Lemma Similar Initial States.

- **CASE** $\tau > 0$. By definition of the SITPN full execution and the $\mathcal{H}$-VHDL full execution relations:
  - $E_c, \tau \vdash s_0 \xrightarrow{\uparrow_0} s_0$
  - $E_c, \tau \vdash s_0 \xrightarrow{\downarrow} s$
  - $E_c, \tau - 1 \vdash sitpn, s \rightarrow \theta_s$
  - $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$
  - $\Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$
  - $E_p, \Delta, \tau, \sigma_0 \vdash \mathrm{d.cs} \rightarrow \theta$

  $\boxed{\gamma \vdash (s_0 :: s :: \theta_s) \sim (\sigma_0 :: \theta)}$

  By definition of the $\mathcal{H}$-VHDL full simulation relation, we know:

  - $E_p, \Delta, \tau, \sigma_0 \vdash d.cs \xrightarrow{\uparrow,\downarrow} \sigma$
  - $E_p, \Delta, \tau - 1, \sigma \vdash \mathrm{d.cs} \rightarrow \theta_\sigma$

  where $\theta = \sigma :: \theta_\sigma$.

  Rewriting θ as $\sigma :: \theta_\sigma$, $\boxed{\gamma \vdash (s_0 :: s :: \theta_s) \sim (\sigma_0 :: \sigma :: \theta_\sigma)}$

  3 subgoals (by def. of Execution Trace Similarity).

  1. $\gamma \vdash s_0 \sim \sigma_0$ (solved by applying Lemma Similar Initial States).
  2. $\gamma \vdash s \sim \sigma$ (solved by applying Lemma First Cycle).
  3. $\gamma \vdash \theta_s \sim \theta_\sigma$ (solved by applying Lemma Bisimulation).

□

Lemma 1 (First Cycle). *For all* $sitpn \in SITPN, d \in design, \gamma \in WM(sitpn, d), s \in S(sitpn), \Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}}), \sigma_e, \sigma_0, \sigma \in \Sigma(\Delta)$, $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$, $E_p \in (\mathbb{N} \times \{\uparrow, \downarrow\}) \to Ins(\Delta) \to value$, *assume that:*

- $\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$ *and* $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$ *and* $\gamma \vdash E_p \stackrel{env}{=} E_c$
- σ_0 *is the initial state of* Δ*:* $\Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$
- *First execution cycle for d:* $E_p, \Delta, \tau, \sigma_0 \vdash d.cs \xrightarrow{\uparrow,\downarrow} \sigma$
- *Particular first execution cycle for sitpn (first rising edge is idle):*

  $E_c, \tau \vdash s_0 \xrightarrow{\uparrow_0} s_0$ *and* $E_c, \tau \vdash s_0 \xrightarrow{\downarrow} s$

then $\gamma \vdash s \stackrel{\downarrow}{\sim} \sigma$.

Proof. Let's show that the first execution cycle leads to two states verifying the Post Falling Edge State Similarity relation: $\boxed{\gamma \vdash s \stackrel{\downarrow}{\sim} \sigma.}$

By definition of the $\mathcal{H}$-VHDL cycle relation, we have:

- $\texttt{Inject}_{\uparrow}(\sigma_0, E_p, \tau, \sigma_{injr})$ and $\Delta, \sigma_{injr} \vdash \text{d.cs} \xrightarrow{\uparrow} \sigma_r$ and $\Delta, \sigma_r \vdash \text{d.cs} \xrightarrow{\theta} \sigma'$
- $\texttt{Inject}_{\downarrow}(\sigma', E_p, \tau, \sigma_{injf})$ and $\Delta, \sigma_{injf} \vdash \text{d.cs} \xrightarrow{\downarrow} \sigma_f$ and $\Delta, \sigma_f \vdash \text{d.cs} \xrightarrow{\theta'} \sigma$

Then, we can apply the Falling Edge lemma to solve $\boxed{\gamma \vdash s \stackrel{\downarrow}{\sim} \sigma.}$

One premise of the Falling Edge lemma remains to be proved: $\boxed{\gamma, E_c, \tau \vdash s_0 \stackrel{\uparrow}{\sim} \sigma'.}$

Then, we can apply the First Rising Edge lemma to solve $\boxed{\gamma, E_c, \tau \vdash s_0 \stackrel{\uparrow}{\sim} \sigma'.}$

□

Lemma 2 (Bisimulation). *For all* $sitpn$, d, γ, E_p, E_c, τ, s, θ_s, σ, θ_σ, Δ, σ_e, *assume that:*

- $\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$ *and* $\gamma \vdash E_p \stackrel{env}{=} E_c$ *and* $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} \Delta, \sigma_e$
- *Starting states are similar as intended after a falling edge:* $\gamma \vdash s \stackrel{\downarrow}{\sim} \sigma$
- $E_c, \tau \vdash sitpn, s \to \theta_s$
- $E_p, \Delta, \tau, \sigma \vdash d.cs \to \theta_\sigma$

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

Proof. Induction on τ.

- Base case, $\tau = 0$: traces are empty, trivial.
- Induction case, $\tau > 0$:

$\forall s, \sigma, \theta_s, \theta_\sigma$ s.t. $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$ and $E_c, \tau - 1 \vdash sitpn, s \rightarrow \theta_s$ and $E_p, \Delta, \tau - 1, \sigma \vdash d.cs \rightarrow \theta_\sigma$ then $\gamma \vdash \theta_s \sim \theta_\sigma$.

By definition of the SITPN execution and the $\mathcal{H}$-VHDL simulation relations for $\tau > 0$:

- $E, \tau \vdash sitpn, s \xrightarrow{\uparrow,\downarrow} s'$ and $E_c, \tau - 1 \vdash sitpn, s \rightarrow \theta_s$.
- $E_p, \Delta, \tau, \sigma \vdash \text{d.cs} \xrightarrow{\uparrow,\downarrow} \sigma'$ and $E_p, \Delta, \tau - 1, \sigma \vdash d.cs \rightarrow \theta_\sigma$.

$\boxed{\gamma \vdash (s' :: \theta_s) \sim (\sigma' :: \theta_\sigma)}$.

2 subgoals (by def. of Execution Trace Similarity).

1. $\boxed{\gamma \vdash s' \sim \sigma'}$ (solved with Step).
2. $\boxed{\gamma \vdash \theta_s \sim \theta_\sigma}$ (solved with Step and IH).

□

Lemma 3 (Step). *For all $sitpn$, d, γ, E_p, E_c, τ, s, s'', σ, σ'', Δ, σ_e, assume that:*

- $\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$ *and* $E_p \overset{env}{=} E_c$ *and* $\mathcal{D}_{\mathcal{H}}, \varnothing \vdash d \xrightarrow{elab} \Delta, \sigma_e$
- $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$
- *From state s to s'' in one execution cycle:* $E_c, \tau \vdash sitpn, s \xrightarrow{\uparrow,\downarrow} s''$
- *From state σ to σ'' in one simulation cycle:* $E_p, \Delta, \tau, \sigma \vdash d.cs \xrightarrow{\uparrow,\downarrow} \sigma''$

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

Proof. By def. of the SITPN and $\mathcal{H}$-VHDL cycle relations:

- $E_c, \tau \vdash sitpn, s \xrightarrow{\uparrow} s'$ and $E_c, \tau \vdash sitpn, s' \xrightarrow{\downarrow} s''$
- $\texttt{Inject}_{\uparrow}(\sigma, E_p, \tau, \sigma_{injr})$ and $\Delta, \sigma_{injr} \vdash \text{d.cs} \xrightarrow{\uparrow} \sigma_r$ and $\Delta, \sigma_r \vdash \text{d.cs} \xrightarrow{\theta} \sigma'$
- $\texttt{Inject}_{\downarrow}(\sigma', E_p, \tau, \sigma_{injf})$ and $\Delta, \sigma_{injf} \vdash \text{d.cs} \xrightarrow{\downarrow} \sigma_f$ and $\Delta, \sigma_f \vdash \text{d.cs} \xrightarrow{\theta'} \sigma''$

Solved by applying Rising Edge and then "Falling Edge" lemmas. □

1.3 Initial States

Definition 8 (Initial State Hypotheses). *Given an $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}}), \sigma_e, \sigma_0 \in \Sigma(\Delta)$, assume that:*

- *SITPN $sitpn$ translates into design d:* $\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$

- *Δ is the elaborated version of d, σ_e is the default state of Δ, i.e, state of Δ where all signals have their default value:*
  $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$
- *σ_0 is the initial state of Δ: $\Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$*

Lemma 4 (Similar Initial States)**.** *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}}), \sigma_e, \sigma_0 \in \Sigma(\Delta)$ that verify the hypotheses of Def. 8, then $\gamma \vdash s_0 \sim \sigma_0$.*

Proof. By definition of State Similarity, 6 subgoals.

1. $\forall p \in P, id_p \in Comps(\Delta), \sigma_p^0 \in \Sigma(\Delta(id_p))$ $s.t.$ $\gamma(p) = id_p$ and $\sigma_0(id_p) = \sigma_p^0$, $s_0.M(p) = \sigma_p^0(''s\_marking'')$.
2. $\forall t \in T_i, id_t \in Comps(\Delta), \sigma_t^0 \in \Sigma(\Delta(id_t))$ $s.t.$ $\gamma(t) = id_t$ and $\sigma_0(id_t) = \sigma_t^0$,
   $upper(I_s(t)) = \infty \wedge s_0.I(t) \leq lower(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t^0(''s\_tc'') \wedge$
   $upper(I_s(t)) = \infty \wedge s_0.I(t) > lower(I_s(t)) \Rightarrow \sigma_t^0(''s\_tc'') = lower(I_s(t)) \wedge$
   $upper(I_s(t)) \neq \infty \wedge s_0.I(t) > upper(I_s(t)) \Rightarrow \sigma_t^0(''s\_tc'') = upper(I_s(t)) \wedge$
   $upper(I_s(t)) \neq \infty \wedge s_0.I(t) \leq upper(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t^0(''s\_tc'')$.
3. $\forall t \in T_i, id_t \in Comps(\Delta), \sigma_t^0 \in \Sigma(\Delta(id_t))$ $s.t.$ $\gamma(t) = id_t$ and $\sigma_0(id_t) = \sigma_t^0$,
   $s_0.reset_t(t) = \sigma_t^0(''s\_reinit\_time\_counter'')$.
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 Lemma Initial States Equal Marking to solve 1.
- Apply Lemma Initial States Equal Time Counters to solve 2.
- Apply Lemma Initial States Equal Reset Orders to solve 3.
- Apply Lemma Initial States Equal Condition Values to solve 4.
- Apply Lemma Initial States Equal Action Executions to solve 5.
- Apply Lemma Initial States Equal Function Executions to solve 6.

□

1.3.1 Initial states and marking

Lemma 5 (Initial States Equal Marking)**.** *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}}), \sigma_e, \sigma_0 \in \Sigma(\Delta)$ that verify the hypotheses of Def. 8, then $\forall p \in P, id_p \in Comps(\Delta), \sigma_p^0 \in \Sigma(\Delta(id_p))$ s.t. $\gamma(p) = id_p$ and $\sigma_0(id_p) = \sigma_p^0$, $s_0.M(p) = \sigma_p^0(''s\_marking'')$.*

Proof. Given a $p \in P$, an $id_p \in Comps(\Delta)$ and a $\sigma_p^0 \in \Sigma(\Delta(id_p))$ s.t. $\gamma(p) = id_p$ and $\sigma_0(id_p) = \sigma_p^0$, let's show that
$\boxed{s_0.M(p) = \sigma_p^0(''s\_marking'').}$

By definition of id_p, there exist gm_p, ipm_p, opm_p s.t. $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$.

By property of the $\mathcal{H}$-VHDL initialization relation, the P design behavior (process "`marking`"), and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$, then $\sigma_p^0(''s\_marking'') = \sigma_p^0(''initial\_marking'')$.

Rewriting $\sigma_p^0(''s\_marking'')$ as $\sigma_p^0(''initial\_marking'')$, $\boxed{\sigma_p^0(''initial\_marking'') = s_0.M(p).}$

By construction, $<\texttt{id}_\texttt{p}\texttt{.initial\_marking} \Rightarrow M_0(p)> \in ipm_p$. By property of the $\mathcal{H}$-VHDL initialization relation, and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$, then $\sigma_p^0(''initial\_marking'') = M_0(p)$.

By definition of s_0, rewriting $s_0.M(p)$ as $M_0(p)$, $\sigma_p^0(''initial\_marking'') = s_0.M(p).$

□

1.3.2 Initial states and time counters

Lemma 6 (Initial States Equal Time Counters). *For all $sitpn \in SITPN, d \in design, \gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_\mathcal{H}), \sigma_e, \sigma_0 \in \Sigma(\Delta)$ that verify the hypotheses of Def. 8, then $\forall t \in T_i, id_t \in Comps(\Delta), \sigma_t^0 \in \Sigma(\Delta(id_t))$ s.t. $\gamma(t) = id_t$ and $\sigma_0(id_t) = \sigma_t^0$,*
$upper(I_s(t)) = \infty \wedge s_0.I(t) \leq lower(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t^0(''s\_tc'') \wedge$
$upper(I_s(t)) = \infty \wedge s_0.I(t) > lower(I_s(t)) \Rightarrow \sigma_t^0(''s\_tc'') = lower(I_s(t)) \wedge$
$upper(I_s(t)) \neq \infty \wedge s_0.I(t) > upper(I_s(t)) \Rightarrow \sigma_t^0(''s\_tc'') = upper(I_s(t)) \wedge$
$upper(I_s(t)) \neq \infty \wedge s_0.I(t) \leq upper(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t^0(''s\_tc'')$.

Proof. Given a $t \in T_i$, an $id_t \in Comps(\Delta)$ and a $\sigma_t^0 \in \Sigma(\Delta(id_t))$ s.t. $\gamma(t) = id_t$ and $\sigma_0(id_t) = \sigma_t^0$, let's show that:

1. $\boxed{upper(I_s(t)) = \infty \wedge s_0.I(t) \leq lower(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t^0(''s\_tc'')}$
2. $\boxed{upper(I_s(t)) = \infty \wedge s_0.I(t) > lower(I_s(t)) \Rightarrow \sigma_t^0(''s\_tc'') = lower(I_s(t))}$
3. $\boxed{upper(I_s(t)) \neq \infty \wedge s_0.I(t) > upper(I_s(t)) \Rightarrow \sigma_t^0(''s\_tc'') = upper(I_s(t))}$
4. $\boxed{upper(I_s(t)) \neq \infty \wedge s_0.I(t) \leq upper(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t^0(''s\_tc'')}$

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$.

Then, let's show the 4 previous subgoals.

1. Assume $upper(I_s(t)) = \infty \wedge s_0.I(t) \leq lower(I_s(t))$, then show $\boxed{s_0.I(t) = \sigma_t^0(''s\_tc'').}$

   Rewriting $s_0.I(t)$ as 0, by definition of s_0, $\boxed{\sigma_t^0(''s\_tc'') = 0.}$

   By property of the $\mathcal{H}$-VHDL initialization relation, the T design behavior (process "`time_counter`"), and $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$, then $\sigma_t^0(''s\_tc'') = 0.$

2. Assume $upper(I_s(t)) = \infty \wedge s_0.I(t) > lower(I_s(t))$, then show $\boxed{\sigma_t^0(''s\_tc'') = lower(I_s(t))}$. By definition, $lower(I_s(t)) \in \mathbb{N}^*$ and $s_0.I(t) = 0$. Then, $lower(I_s(t)) < 0$ is a contradiction.

3. Assume $upper(I_s(t)) \neq \infty \wedge s_0.I(t) > upper(I_s(t))$, then show $\boxed{\sigma_t^0(''s\_tc'') = upper(I_s(t))}$. By definition, $upper(I_s(t)) \in \mathbb{N}^*$ and $s_0.I(t) = 0$. Then, $upper(I_s(t)) < 0$ is a contradiction.

4. Assume $upper(I_s(t)) \neq \infty \wedge s_0.I(t) \leq upper(I_s(t))$, then show $\boxed{s_0.I(t) = \sigma_t^0(''s\_tc'')}$.

   Rewriting $s_0.I(t)$ as 0, by definition of s_0, $\boxed{\sigma_t^0(''s\_tc'') = 0.}$

   By property of the $\mathcal{H}$-VHDL initialization relation, the T design behavior (process "`time_counter`"), and $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$, then $\sigma_t^0(''s\_tc'') = 0.$

□

1.3.3 Initial states and reset orders

Lemma 7 (Initial States Equal Reset Orders)**.** *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}})$, $\sigma_e, \sigma_0 \in \Sigma(\Delta)$ that verify the hypotheses of Def. 8, then $\forall t \in T_i, id_t \in Comps(\Delta), \sigma_t^0 \in \Sigma(\Delta(id_t))$ s.t. $\gamma(t) = id_t$ and $\sigma_0(id_t) = \sigma_t^0$, $s_0.reset_t(t) = \sigma_t^0(''s\_reinit\_time\_counter'')$.*

Proof. Given a $t \in T_i$, an $id_t \in Comps(\Delta)$ and a $\sigma_t^0 \in \Sigma(\Delta(id_t))$ s.t. $\gamma(t) = id_t$, let's show that $\boxed{s_0.reset_t(t) = \sigma_t^0(''s\_reinit\_time\_counter'')}$.

Rewriting $s_0.reset_t(t)$ as $false$, by definition of s_0, $\boxed{\sigma_t^0(''s\_reinit\_time\_counter'') = false.}$

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$.

By property of the $\mathcal{H}$-VHDL initialization relation, the T design behavior (process `reinit_time_counter_evaluation`), and $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$,

we know $\sigma_t^0(''s\_reinit\_time\_counter'') = \prod_{i=0}^{\Delta(id_t)(''in\_arcs\_nb'')-1} \sigma_t^0(''rt'')(i)$, where $\Delta(id_t)(''in\_arcs\_nb'')$ is the value of the generic constant $''in\_arcs\_nb''$ stored in the elaborated design $\Delta(id_t)$ (which, by property of the $\mathcal{H}$-VHDL elaboration relation, is an elaborated version of the T design).

Rewriting $\sigma_t^0(''s\_reinit\_time\_counter'')$ as $\prod_{i=0}^{\Delta(id_t)(''in\_arcs\_nb'')-1} \sigma_t^0(''rt'')(i)$,

$$\boxed{\prod_{i=0}^{\Delta(id_t)(''in\_arcs\_nb'')-1} \sigma_t^0(''rt'')(i) = false.}$$

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).

Case analysis on $input(t)$ (2 CASES).

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

  By construction, $<\mathtt{id_t.in\_arcs\_nb} \Rightarrow 1> \in gm_t$, and by property of the elaboration relation, $\Delta(id_t)(''in\_arcs\_nb'') = 1$. By construction, $< \mathtt{id_t.rt(0)} \Rightarrow false > \in ipm_t$, and by property of the initialization relation, $\sigma_t^0(''rt'')(0) = false$.

  Rewriting $\Delta(id_t)(''in\_arcs\_nb'')$ as 1 and $\sigma_t^0(''rt'')(0)$ as $false$,
  $$\prod_{i=0}^{\Delta(''in\_arcs\_nb'')-1} \sigma_t^0(''rt'')(i) = \sigma_t^0(''rt'')(0) = false.$$

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

  We know $\prod_{i=0}^{\Delta(id_t)(''in\_arcs\_nb'')-1} \sigma_t^0(''rt'')(i) = false \equiv \exists i \in [0, \Delta(id_t)(''in\_arcs\_nb'') - 1]\ s.t.\ \sigma_t^0(''rt'')(i) = false$.
  $\boxed{\exists i \in [0, \Delta(id_t)(''in\_arcs\_nb'') - 1]\ s.t.\ \sigma_t^0(''rt'')(i) = false.}$
  Since $input(t) \neq \emptyset$, $\exists p\ s.t.\ p \in input(t)$. Let's 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 definition of id_p, there exist $gm_p, ipm_p, opm_p\ s.t.\ \mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$.

  By construction, for all $p \in P, t \in T$ s.t. $p \in input(t)$ and $t \in output(p)$, for all id_p, id_t s.t. $\gamma(p) = id_p$ and $\gamma(t) = id_t$, for all gm_p, ipm_p, opm_p s.t. $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$ and gm_t, ipm_t, opm_t s.t. $\mathtt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$, there exist $i \in [0, |input(t)| - 1]$, $j \in [0, |output(p)| - 1]$, id_{ji} s.t. $<\mathtt{id_p.rtt(j)} \Rightarrow id_{ji}> \in opm_p$ and $<\mathtt{id_t.rt(i)} \Rightarrow id_{ji}> \in ipm_t$. Let's take such a i, j and id_{ji}.

  By construction, for all $t \in T$ s.t. $input(t) \neq \emptyset$, id_t, gm_t, ipm_t, opm_t s.t. $\gamma(t) = id_t$ and $\mathtt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$, then $<\mathtt{id_t.in\_arcs\_nb} \Rightarrow |input(t)|> \in gm_t$.

  By property of the $\mathcal{H}$-VHDL elaboration relation and $<\mathtt{id_t.in\_arcs\_nb} \Rightarrow |input(t)|> \in gm_t$, we know $\Delta(id_t)(''in\_arcs\_nb) = |input(t)|$.

  Rewriting $\Delta(id_t)(''in\_arcs\_nb)$ as $|input(t)|$, we have $i \in [0, \Delta(id_t)(''in\_arcs\_nb) - 1]$. Let's take that i to prove the goal.

  $\boxed{\sigma_t^0(''rt'')(i) = false.}$

  By property of the $\mathcal{H}$-VHDL initialization relation and $<\mathtt{id_t.rt(i)} \Rightarrow id_{ji}> \in ipm_t$, we know $\sigma_t^0(''rt'')(i) = \sigma_0(''id_{ji}'')$.

  Rewriting $\sigma_t^0(''rt'')(i)$ as $\sigma_0(''id_{ji}'')$, $\boxed{\sigma_0(''id_{ji}'') = false.}$

By property of the $\mathcal{H}$-VHDL elaboration and initialization relations, and $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$, there exists a $\sigma_p^0 \in \Sigma(\Delta(id_p))$ s.t. $\sigma_0(id_p) = \sigma_p^0$.

By property of the $\mathcal{H}$-VHDL initialization relation and $< \texttt{id}_\texttt{p}.\texttt{rtt}(\texttt{j}) \Rightarrow id_{ji} > \in opm_p$, we know $\sigma_0("id_{ji}") = \sigma_p^0("rtt")(j)$.

Rewriting $\sigma_0("id_{ji}")$ as $\sigma_p^0("rtt")(j)$, $\boxed{\sigma_p^0("rtt")(j) = false.}$

By property of the $\mathcal{H}$-VHDL initialization relation, the P design behavior (process `reinit_transitions_time_evaluation`), and $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$, we know that for all $j \in [0, \Delta(id_p)("out\_arcs\_nb") - 1]$, $\sigma_p^0("rtt")(j) = false$.

By construction, for all $p \in P$ s.t. $output(p) \neq \emptyset$, $id_p \in Comps(\Delta), gm_p, ipm_p, opm_p$ s.t. $\gamma(p) = id_p$ and $\texttt{comp}(id_p, "transition", gm_p, ipm_p, opm_p) \in d.cs$, then $<\texttt{id}_\texttt{p}.\texttt{out\_arcs\_nb} \Rightarrow |output(p)|> \in gm_p$.

By property of the $\mathcal{H}$-VHDL elaboration relation and $<\texttt{id}_\texttt{p}.\texttt{out\_arcs\_nb} \Rightarrow |output(p)|> \in gm_p$, we know $\Delta(id_p)("out\_arcs\_nb") = |output(p)|$.

Rewriting $|output(p)|$ as $\Delta(id_p)("out\_arcs\_nb)$, we have $j \in [0, \Delta(id_p)("out\_arcs\_nb) - 1]$. Then, we can deduce $\sigma_p^0("rtt")(j) = false$.

□

1.3.4 Initial states and condition values

Lemma 8 (Initial States Equal Condition Values). *For all $sitpn \in SITPN, d \in design, \gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_\mathcal{H}), \sigma_e, \sigma_0 \in \Sigma(\Delta)$ that verify the hypotheses of Def. 8, 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's show that $\boxed{s_0.cond(c) = \sigma_0(id_c).}$

Rewriting $s_0.cond(c)$ as $false$, by definition of s_0, $\boxed{\sigma_0(id_c) = false.}$
By construction, id_c is an input port identifier of boolean type in the $\mathcal{H}$-VHDL design d.
By property, of the $\mathcal{H}$-VHDL elaboration relation, $\sigma_e(id_c) = false$, where $false$ is the default value associated to signals of the boolean type during the elaboration (see definition of default value in chapter $\mathcal{H}$-VHDL semantics).
By property of the $\mathcal{H}$-VHDL initialization relation, we have $\sigma_e(id_c) = \sigma_0(id_c)$ (i.e, input ports are not assigned during the initialization phase).
Rewriting $\sigma_e(id_c)$ as $false$, $\sigma_0(id_c) = false$.

□

1.3.5 Initial states and action executions

Correction: id_f is assigned by the reset block of the function process

Lemma 9 (Initial States Equal Action Executions). *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}})$, $\sigma_e, \sigma_0 \in \Sigma(\Delta)$ that verify the hypotheses of Def. 8, 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's show that $\boxed{s_0.ex(a) = \sigma_0(id_a).}$

Rewriting $s_0.ex(a)$ as $false$, by definition of s_0, $\boxed{\sigma_0(id_a) = false.}$
By construction, id_a is an output port identifier of boolean type in the $\mathcal{H}$-VHDL design d.
By property, of the $\mathcal{H}$-VHDL elaboration relation, $\sigma_e(id_a) = false$, where $false$ is the default value associated to signals of the boolean type during the elaboration (see definition of default value in chapter $\mathcal{H}$-VHDL semantics).
By construction, we know that the output port identifier id_a is assigned in the generated `action` process, only at the falling edge phase of the simulation cycle (i.e, the assignment takes place in a `falling` statement block).
By property of the $\mathcal{H}$-VHDL initialization relation, and we have $\sigma_e(id_a) = \sigma_0(id_a)$ (i.e, process `action` is idle during the initialization phase).
Rewriting $\sigma_e(id_a)$ as $false$, $\sigma_0(id_a) = false.$

□

1.3.6 Initial states and function executions

Correction: id_f is assigned by the reset block of the function process

Lemma 10 (Initial States Equal Function Executions). *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}})$, $\sigma_e, \sigma_0 \in \Sigma(\Delta)$ that verify the hypotheses of Def. 8, 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's show that $\boxed{s_0.ex(f) = \sigma_0(id_f).}$

Rewriting $s_0.ex(f)$ as $false$, by definition of s_0, $\boxed{\sigma_0(id_f) = false.}$
By construction, id_f is an output port identifier of boolean type in the $\mathcal{H}$-VHDL design d.
By property, of the $\mathcal{H}$-VHDL elaboration relation, $\sigma_e(id_f) = false$, where $false$ is the default value associated to signals of the boolean type during the elaboration (see definition of default value in chapter $\mathcal{H}$-VHDL semantics).
By construction, we know that the output port identifier id_f is assigned in the generated `function` process (i.e, `function` is the process identifier), only at the rising edge phase of the simulation cycle (i.e, the assignment takes place in a `rising` statement block).
By property of the $\mathcal{H}$-VHDL initialization relation, and we have $\sigma_e(id_f) = \sigma_0(id_f)$ (i.e, process `function` is idle during the initialization phase).
Rewriting $\sigma_e(id_f)$ as $false$, $\sigma_0(id_f) = false.$

□

1.4 First Rising Edge

Definition 9 (First Rising Edge Hypotheses). *Given an* $sitpn \in SITPN, d \in design, \gamma \in WM(sitpn, d), \Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}}), \sigma_e, \sigma_0, \sigma_i, \sigma_{\uparrow}, \sigma \in \Sigma(\Delta)$, $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$, $E_p \in (\mathbb{N} \times \{\uparrow, \downarrow\}) \to Ins(\Delta) \to value$, $\tau \in \mathbb{N}$, *assume that:*

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

Lemma 11 (First Rising Edge). *For all* $sitpn, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_{\uparrow}, \sigma$, E_c, E_p, τ *that verify the hypotheses of Def. 9, then* $\gamma, E_c, \tau \vdash s_0 \stackrel{\uparrow}{\sim} \sigma$*.*

Proof. By definition of Post Rising Edge State Similarity, 6 subgoals.

1. $\forall p \in P, id_p \in Comps(\Delta), \sigma_p \in \Sigma(\Delta(id_p))$ $s.t.$ $\gamma(p) = id_p$ and $\sigma(id_p) = \sigma_p$, $s_0.M(p) = \sigma_p(''s\_marking'')$.
2. $\forall t \in T_i, id_t \in Comps(\Delta), \sigma_t \in \Sigma(\Delta(id_t))$ $s.t.$ $\gamma(t) = id_t$ and $\sigma(id_t) = \sigma_t$,
$upper(I_s(t)) = \infty \wedge s_0.I(t) \leq lower(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t(''s\_tc'') \wedge$
$upper(I_s(t)) = \infty \wedge s_0.I(t) > lower(I_s(t)) \Rightarrow \sigma_t(''s\_tc'') = lower(I_s(t)) \wedge$
$upper(I_s(t)) \neq \infty \wedge s_0.I(t) > upper(I_s(t)) \Rightarrow \sigma_t(''s\_tc'') = upper(I_s(t)) \wedge$
$upper(I_s(t)) \neq \infty \wedge s_0.I(t) \leq upper(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t(''s\_tc'')$.
3. $\forall t \in T_i, id_t \in Comps(\Delta), \sigma_t \in \Sigma(\Delta(id_t))$ $s.t.$ $\gamma(t) = id_t$ and $\sigma(id_t) = \sigma_t$,
$s_0.reset_t(t) = \sigma_t(''s\_reinit\_time\_counter'')$.
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. $\forall t \in T_i, id_t \in Comps(\Delta)$ $s.t.$ $\gamma(t) = id_t$,
$t \in Sens(s.M) \Leftrightarrow \sigma(id_t)(''s\_enabled'') = \mathtt{true}$.
7. $\forall t \in T, id_t \in Comps(\Delta)$ $s.t.$ $\gamma(t) = id_t$,
$$\sigma(id_t)(''s\_condition\_combination'') = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \mathtt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}$$
where $conds(t) = \{c \in \mathcal{C} \mid \mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1\}$.

– Apply Lemma First Rising Edge Equal Marking to solve 1.

– Apply Lemma First Rising Edge Equal Time Counters to solve 2.

– Apply Lemma First Rising Edge Equal Reset Orders to solve 3.

– Apply Lemma "First Rising Edge Equal Action Executions" to solve 4.

– Apply Lemma "First Rising Edge Equal Function Executions " to solve 5.

– Apply Lemma "Rising Edge Equal Sensitized" to solve 6.

– Apply Lemma "Rising Edge Equal Condition Combination" to solve 7.

□

1.4.1 First rising edge and marking

Lemma 12 (First Rising Edge Equal Marking). *For all $sitpn, d, \gamma, \Delta$, $\sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma$, E_c, E_p, τ that verify the hypotheses of Def. 9, then $\forall p \in P, id_p \in Comps(\Delta), \sigma_p \in \Sigma(\Delta(id_p))$ s.t. $\gamma(p) = id_p$ and $\sigma(id_p) = \sigma_p$, $s_0.M(p) = \sigma_p(''s\_marking'')$.*

Proof. Given a p, id_p, σ_p s.t. $\gamma(p) = id_p$ and $\sigma(id_p) = \sigma_p$, let us show that $\boxed{s_0.M(p) = \sigma_p(''s\_marking'').}$
By definition of id_p, there exist gmp_p, ipm_p, opm_p s.t. $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$.

By property of the $\mathcal{H}$-VHDL elaboration relation, the $\mathcal{H}$-VHDL initialization relation, the $\texttt{Inject}_\uparrow$ relation, the $\mathcal{H}$-VHDL rising edge relation and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$, there exist a $\sigma_p^e, \sigma_p^0, \sigma_p^{injr}, \sigma_p^r \in \Sigma(\Delta)$ s.t. $\sigma_e(id_p) = \sigma_p^e$ and $\sigma_0(id_p) = \sigma_p^0$ and $\sigma_i(id_p) = \sigma_p^{injr}$ and $\sigma_r(id_p) = \sigma_p^r$
.

> From the elaboration to the end of the first rising edge phase, an internal state is associated with the P component instance id_p in the component store of the top-level design d.

By property of the $\mathcal{H}$-VHDL rising edge relation, the P design behavior (process "`marking`"), and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$, then
$\sigma_p^r(''s\_marking'') = \sigma_p^{injr}(''s\_marking'') + \sigma_p^{injr}(''s\_input\_token\_sum'') - \sigma_p^{injr}(''s\_output\_token\_sum'')$.

> Result of the execution of the process "`marking`" that performs the signal assignment `s_marking` $\Leftarrow$ `s_marking` $+$ `s_input_token_sum` $-$ `s_output_token_sum`.

By property of the $\mathcal{H}$-VHDL stabilize relation, the P design behavior (process "`marking`"), and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$, then $\sigma_p^r(''s\_marking'') = \sigma_p(''s\_marking'')$.

> As it is only assigned by the process "`marking`", and as the process "`marking`" is never executed during the stabilization phase, the "`s_marking`" signal has an invariant value during the stabilization phase.

Rewriting $\sigma_p(''s\_marking'')$ as $\sigma_p^r(''s\_marking'')$, and $\sigma_p^r(''s\_marking'')$ as
$\sigma_p^{injr}(''s\_marking'') + \sigma_p^{injr}(''s\_input\_token\_sum'') - \sigma_p^{injr}(''s\_output\_token\_sum'')$,
$\boxed{s_0.M(p) = \sigma_p^{injr}(''s\_marking'') + \sigma_p^{injr}(''s\_input\_token\_sum'') - \sigma_p^{injr}(''s\_output\_token\_sum'').}$

By property of the $\texttt{Inject}_\uparrow$ relation, $\sigma_p^{injr}(''s\_marking'') = \sigma_p^0(''s\_marking'')$ and
$\sigma_p^{injr}(''s\_input\_token\_sum'') = \sigma_p^0(''s\_input\_token\_sum'')$ and
$\sigma_p^{injr}(''s\_output\_token\_sum'') = \sigma_p^0(''s\_output\_token\_sum'')$. Rewriting the above,
$\boxed{s_0.M(p) = \sigma_p^0(''s\_marking'') + \sigma_p^0(''s\_input\_token\_sum'') - \sigma_p^0(''s\_output\_token\_sum'').}$

Detail the two lemmas giving this property.

By property of the $\mathcal{H}$-VHDL initialization relation, $\sigma_p^0(\text{"}s\_input\_token\_sum\text{"}) = 0$ and $\sigma_p^0(\text{"}s\_output\_token\_sum\text{"}) = 0$. Rewriting the above, $\boxed{s_0.M(p) = \sigma_p^0(\text{"}s\_marking\text{"}).}$

Applying the Initial States Equal Marking lemma, $s_0.M(p) = \sigma_p^0(\text{"}s\_marking\text{"})$. □

1.4.2 First rising edge and time counters

Lemma 13 (First Rising Edge Equal Time Counters). *For all $sitpn, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_\uparrow, \sigma, E_c, E_p, \tau$ that verify the hypotheses of Def. 9, then*
$\forall t \in T_i, id_t \in Comps(\Delta), \sigma_t \in \Sigma(\Delta(id_t))$ *s.t.* $\gamma(t) = id_t$ *and* $\sigma(id_t) = \sigma_t$,
$upper(I_s(t)) = \infty \wedge s_0.I(t) \le lower(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t(\text{"}s\_tc\text{"}) \wedge$
$upper(I_s(t)) = \infty \wedge s_0.I(t) > lower(I_s(t)) \Rightarrow \sigma_t(\text{"}s\_tc\text{"}) = lower(I_s(t)) \wedge$
$upper(I_s(t)) \ne \infty \wedge s_0.I(t) > upper(I_s(t)) \Rightarrow \sigma_t(\text{"}s\_tc\text{"}) = upper(I_s(t)) \wedge$
$upper(I_s(t)) \ne \infty \wedge s_0.I(t) \le upper(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t(\text{"}s\_tc\text{"})$.

Proof. Given a $t \in T_i$, an $id_t \in Comps(\Delta)$ and a $\sigma_t \in \Sigma(\Delta(id_t))$ $s.t.$ $\gamma(t) = id_t$ and $\sigma(id_t) = \sigma_t$, let's show that:

1. $\boxed{upper(I_s(t)) = \infty \wedge s_0.I(t) \le lower(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t(\text{"}s\_tc\text{"})}$
2. $\boxed{upper(I_s(t)) = \infty \wedge s_0.I(t) > lower(I_s(t)) \Rightarrow \sigma_t(\text{"}s\_tc\text{"}) = lower(I_s(t))}$
3. $\boxed{upper(I_s(t)) \ne \infty \wedge s_0.I(t) > upper(I_s(t)) \Rightarrow \sigma_t(\text{"}s\_tc\text{"}) = upper(I_s(t))}$
4. $\boxed{upper(I_s(t)) \ne \infty \wedge s_0.I(t) \le upper(I_s(t)) \Rightarrow s_0.I(t) = \sigma_t(\text{"}s\_tc\text{"})}$

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$.

By property of the $\mathcal{H}$-VHDL elaboration relation, the $\mathcal{H}$-VHDL initialization relation, the $\texttt{Inject}_\uparrow$ relation, the $\mathcal{H}$-VHDL rising edge relation and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$, there exist a $\sigma_t^e, \sigma_t^0, \sigma_t^{injr}, \sigma_t^r \in \Sigma(\Delta)$ s.t. $\sigma_e(id_t) = \sigma_t^e$ and $\sigma_0(id_t) = \sigma_t^0$ and $\sigma_i(id_t) = \sigma_t^{injr}$ and $\sigma_r(id_t) = \sigma_t^r$
.

> From the elaboration to the end of the first rising edge phase, an internal state is associated with the T component instance id_t in the component store of the top-level design d.

Then, let's show the 4 previous subgoals.

1. Assume $upper(I_s(t)) = \infty \wedge s_0.I(t) \le lower(I_s(t))$, then show $\boxed{s_0.I(t) = \sigma_t(\text{"}s\_tc\text{"}).}$
   By property of the $\texttt{Inject}_\uparrow$ relation, the $\mathcal{H}$-VHDL rising edge and stabilize relations, and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$, $\sigma_t(\text{"}s\_tc\text{"}) = \sigma_t^0(\text{"}s\_tc\text{"})$.

   > The above equality is deduced from the two following facts:
   >
   > - The process "`time_counter`" is the only process that assigns signal `s_tc` in the T component behavior, and it is never executed during the rising edge and stabilization phases.

- The values of component instances' internal signals are invariant through the $\texttt{Inject}_{\uparrow}$ relation.

Rewriting $\sigma_t(\text{"}s\_tc\text{"})$ as $\sigma_t^0(\text{"}s\_tc\text{"})$, $\boxed{s_0.I(t) = \sigma_t^0(\text{"}s\_tc\text{"}).}$

Applying the Initial States Equal Time Counters lemma, $s_0.I(t) = \sigma_t^0(\text{"}s\_tc\text{"})$.

2. Assume $upper(I_s(t)) = \infty \wedge s_0.I(t) > lower(I_s(t))$, then show $\boxed{\sigma_t(\text{"}s\_tc\text{"}) = lower(I_s(t))}$. By definition, $lower(I_s(t)) \in \mathbb{N}^*$ and $s_0.I(t) = 0$. Then, $lower(I_s(t)) < 0$ is a contradiction.

3. Assume $upper(I_s(t)) \neq \infty \wedge s_0.I(t) > upper(I_s(t))$, then show $\boxed{\sigma_t(\text{"}s\_tc\text{"}) = upper(I_s(t))}$. By definition, $upper(I_s(t)) \in \mathbb{N}^*$ and $s_0.I(t) = 0$. Then, $upper(I_s(t)) < 0$ is a contradiction.

4. Assume $upper(I_s(t)) \neq \infty \wedge s_0.I(t) \leq upper(I_s(t))$, then show $\boxed{s_0.I(t) = \sigma_t(\text{"}s\_tc\text{"})}$.

   By property of the $\texttt{Inject}_{\uparrow}$ relation, the $\mathcal{H}$-VHDL rising edge and stabilize relations, and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$, $\sigma_t(\text{"}s\_tc\text{"}) = \sigma_t^0(\text{"}s\_tc\text{"})$.

   Rewriting $\sigma_t(\text{"}s\_tc\text{"})$ as $\sigma_t^0(\text{"}s\_tc\text{"})$, $\boxed{s_0.I(t) = \sigma_t^0(\text{"}s\_tc\text{"}).}$

   Applying the Initial States Equal Time Counters lemma, $s_0.I(t) = \sigma_t^0(\text{"}s\_tc\text{"})$.

□

1.4.3 First rising edge and reset orders

Lemma 14 (First Rising Edge Equal Reset Orders). *For all $sitpn, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_{\uparrow}, \sigma, E_c, E_p, \tau$ that verify the hypotheses of Def. 9, then*
$\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,
$s_0.reset_t(t) = \sigma(id_t)(\text{"}s\_reinit\_time\_counter\text{"})$.

Proof. Given a $t \in T$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show that $\boxed{s_0.reset_t(t) = \sigma(id_t)(\text{"}srtc\text{"}).}$
By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$.
By property of the $\mathcal{H}$-VHDL stabilize relation and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$, then $\sigma(id_t)(\text{"}srtc\text{"}) = \sum_{i=0}^{\Delta(id_t)(\text{"}input\_arcs\_number\text{"})-1} \sigma(id_t)(\text{"}reinit\_time\text{"})[i]$.

$$\boxed{s_0.reset_t(t) = \sum_{i=0}^{\Delta(id_t)(\text{"}ian\text{"})-1} \sigma(id_t)(\text{"}rt\text{"})[i].}$$

Case analysis on $input(t)$ (2 CASES):

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

  By construction, $<\texttt{input\_arcs\_number} \Rightarrow \texttt{1}> \in gm_t$, and by property of the $\mathcal{H}$-VHDL elaboration relation, then $\Delta(id_t)(\text{"}ian\text{"}) = 1$. By construction, $< \texttt{reinit\_time(0)} \Rightarrow \texttt{false} > \in ipm_t$, and by property of the $\mathcal{H}$-VHDL stabilize relation, $\sigma(id_t)(''rt'')[0] = false$.

Rewriting $\Delta(id_t)(''ian'')$ as 1 and $\sigma(id_t)(''rt'')[0]$ as $false$, and by definition of s_0, $s_0.reset_t(t) = \sum_{i=0}^{\Delta(''ian'')-1} \sigma(id_t$

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

  By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in gm_t$, and by property of the $\mathcal{H}$-VHDL elaboration relation, then $\Delta(id_t)(''ian'') = |input(t)|$.

  Rewriting $\Delta(id_t)(''ian'')$ as $|input(t)|$, $\boxed{s_0.reset_t(t) = \sum_{i=0}^{|input(t)|-1} \sigma(id_t)(''rt'')[i].}$

  By definition of s_0, $s_0.reset_t(t) = false$. Rewriting $s_0.reset_t(t)$ as $false$, $\boxed{\sum_{i=0}^{|input(t)|-1} \sigma(id_t)(''rt'')[i] = false.}$

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

  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 definition of id_p, there exist gm_p, ipm_p, opm_p s.t. $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$.By construction for all $i \in [0, |input(t)| - 1]$, there exist $j \in [0, |output(p)| - 1]$ and $id_{ji} \in Sigs(\Delta)$ s.t. $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in opm_p$ and $<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in ipm_t$.

  By property of the $\mathcal{H}$-VHDL stabilize relation, $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in opm_p$ and $<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in ipm_t$, then $\sigma(id_t)(''rt'')[i] = \sigma(id_{ji}) = \sigma(id_p)(''rtt'')[j]$.

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

  By property of the $\mathcal{H}$-VHDL rising edge and stabilize relations,

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

  Rewriting the goal with the above equation,

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

Add a lemma + proof in section initial states for fired = false after initialization.

By property of the $\mathcal{H}$-VHDL initialization and the $\texttt{Inject}_{\uparrow}$ relations, then $\sigma_0(id_p)(''otf'')[j] = false$. Rewriting $\sigma_0(id_p)(''otf'')[j]$ as $false$ and simplifying the goal,

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

Add a lemma + proof in section initial states for output token sum = 0 after initialization.

By property of the $\mathcal{H}$-VHDL initialization and the $\texttt{Inject}_{\uparrow}$ relations, then $\sigma_0(id_p)(''sots'') = 0$. Rewriting $\sigma_0(id_p)(''sots'')$ as 0 and simplifying the goal, $false = false$

□

1.4.4 First rising edge and action executions

Lemma 15 (First Rising Edge Equal Action Executions). *For all $sitpn, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_{\uparrow}, \sigma, E_c, E_p, \tau$ that verify the hypotheses of Def. 9, 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 $\boxed{s_0.ex(a) = \sigma(id_a).}$

Rewriting $s_0.ex(a)$ as $false$, by definition of s_0, $\boxed{\sigma(id_a) = false.}$
By construction, id_a is an output port identifier of boolean type in the $\mathcal{H}$-VHDL design d assigned only during a falling edge phase in the ``$\texttt{action}$'' process.
By property of the $\mathcal{H}$-VHDL $\texttt{Inject}_{\uparrow}$, rising edge and stabilize relations, then $\sigma(id_a) = \sigma_0(id_a)$.
Thanks to the Lemma Initial States Equal Action Executions, $\sigma_0(id_a) = false$.
Rewriting $\sigma(id_a)$ as $\sigma_0(id_a)$, and $\sigma_0(id_a)$ as $false$, $false = false.$

□

1.4.5 First rising edge and function executions

Lemma 16 (First Rising Edge Equal Function Executions). *For all $sitpn, d, \gamma, \Delta, \sigma_e, \sigma_0, \sigma_i, \sigma_{\uparrow}, \sigma, E_c, E_p, \tau$ that verify the hypotheses of Def. 9, 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, $\boxed{\sigma(id_f) = false.}$
By construction, the ``$\texttt{function}$'' process is a part of design d's behavior, i.e $\texttt{ps}(''function'', \emptyset, sl, ss) \in d.cs$.
By construction id_f is an output port of design d, and it is only assigned in the body of the ``$\texttt{function}$'' process. Let $trs(f)$ be the set of transitions associated to function f, i.e $trs(f) = \{t \in T \mid \mathbb{F}(t, f) = true\}$. Then, depending on $trs(f)$, there are two cases of assignment of output port id_f:

- **CASE** $trs(f) = \emptyset$:
  By construction, $\mathtt{id_f} \Leftarrow \mathtt{false} \in ss_{\uparrow}$ where $ss_{\uparrow}$ is the part of the ``function'' process body executed during the rising edge phase.

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

- **CASE** $trs(f) \neq \emptyset$:
  By construction, $\mathtt{id_f} \Leftarrow \mathtt{id_{ft_0}} + \cdots + \mathtt{id_{ft_n}} \in ss_{\uparrow}$ 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 a internal signal of design d.

  By property of the $\mathtt{Inject}_{\uparrow}$, the $\mathcal{H}$-VHDL rising edge and stabilize relation, then $\sigma(id_f) = \sigma_0(id_{ft_0}) + \cdots + \sigma_0(id_{ft_n})$.

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

  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 definition of id_{t_i}, there exist gm_{t_i}, ipm_{t_i} and opm_{t_i} s.t. $\mathtt{comp}(id_{t_i}, "transition", gm_{t_i}, ipm_{t_i}, opm_{t_i}) \in d.cs$.

  By construction, $<\mathtt{fired} \Rightarrow \mathtt{id_{ft_i}}> \in opm_{t_i}$, and by property of the initialization relation $\sigma_0(id_{ft_i}) = \sigma_0(id_{t_i})("fired")$.

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

  By property of the initialization relation, we know that for all $t \in T$ and $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, then $\sigma_0(id_t)("fired") = false$.

  Rewriting all $\sigma_0(id_{t_i})("fired")$ as $false$ and simplifying the goal, then
  $false = false.$

$\square$

1.5 Rising Edge

Definition 10 (Rising Edge Hypotheses). *Given an $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$, $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}})$, $E_p \in (\mathbb{N} \times \{\uparrow,\downarrow\}) \to Ins(\Delta) \to value$, $\tau \in \mathbb{N}$, $s, s' \in S(sitpn)$, $\sigma_e, \sigma, \sigma_i, \sigma_{\uparrow}, \sigma' \in \Sigma(\Delta)$, assume that:*

- $\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$ *and* $\gamma \vdash E_p \stackrel{env}{=} E_c$ *and* $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash \mathrm{d} \xrightarrow{elab} \Delta, \sigma_e$
- $\gamma \vdash s \stackrel{\downarrow}{\sim} \sigma$
- $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$
- $\mathtt{Inject}_{\uparrow}(\sigma, E_p, \tau, \sigma_i)$ *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:* $\mathcal{D}_{\mathcal{H}}, \Delta, \sigma \vdash \mathrm{d.cs} \xrightarrow{comb} \sigma$

Lemma 17 (Rising Edge). *For all $sitpn$, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Def. 10, then* $\gamma, E_c, \tau \vdash s' \overset{\uparrow}{\sim} \sigma'$.

Proof. By definition of Post Rising Edge State Similarity, there are 7 points to prove.

1. $\forall p \in P, id_p \in Comps(\Delta)\ s'.t.\ \gamma(p) = id_p,\ s'.M(p) = \sigma'(id_p)(''s\_marking'')$.
2. $\forall t \in T_i, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
   $(upper(I_s(t)) = \infty \wedge s'.I(t) \leq lower(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(''s\_time\_counter''))$
   $\wedge (upper(I_s(t)) = \infty \wedge s'.I(t) > lower(I_s(t)) \Rightarrow \sigma'(id_t)(''s\_time\_counter'') = lower(I_s(t)))$
   $\wedge (upper(I_s(t)) \neq \infty \wedge s'.I(t) > upper(I_s(t)) \Rightarrow \sigma'(id_t)(''s\_time\_counter'') = upper(I_s(t)))$
   $\wedge (upper(I_s(t)) \neq \infty \wedge s'.I(t) \leq upper(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(''s\_time\_counter''))$.
3. $\forall t \in T_i, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t, s'.reset_t(t) = \sigma'(id_t)(''s\_reinit\_time\_counter'')$.
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. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t, t \in Sens(s'.M) \Leftrightarrow \sigma'(id_t)(''s\_enabled'') = \texttt{true}$.
7. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t, t \notin Sens(s'.M) \Leftrightarrow \sigma'(id_t)(''s\_enabled'') = \texttt{false}$.
8. $\forall t \in T, id_t \in Comps(\Delta)\ s.t.\ \gamma(t) = id_t,$
   $$\sigma'(id_t)(''s\_condition\_combination'') = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases}$$
   where $conds(t) = \{c \in \mathcal{C} \mid \mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1\}$.

Each point is proved by a separate lemma:

- Apply Lemma Rising Edge Equal Marking to solve 1.
- Apply Lemma Rising Edge Equal Time Counters lemma to solve 2.
- Apply Lemma Rising Edge Equal Reset Orders to solve 3.
- Apply Lemma Rising Edge Equal Action Executions to solve 4.
- Apply Lemma Rising Edge Equal Function Executions to solve 5.
- Apply Lemma Rising Edge Equal Sensitized to solve 6.
- Apply Lemma Rising Edge Equal Not Sensitized to solve 7.
- Apply Lemma Rising Edge Equal Condition Combination to solve 8.

□

1.5.1 Rising Edge and Marking

Lemma 18 (Rising Edge Equal Marking). *For all* $sitpn$, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' *that verify the hypotheses of Def. 10, then* $\forall p, id_p$ *s.t.* $\gamma(p) = id_p$ *and* $\sigma'(id_p) = \sigma'_p$, $s'.M(p) = \sigma'_p(\text{"}s\_marking\text{"})$.

Proof. Given a $p \in P$, let us show $\boxed{s'.M(p) = \sigma'(id_p)(\text{"}s\_marking\text{"}).}$
By definition of id_p, there exist gm_p, ipm_p, opm_p s.t. $\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$.By definition of the SITPN state transition relation on rising edge:

$$s'.M(p) = s.M(p) - \sum_{t \in Fired(s)} pre(p,t) + \sum_{t \in Fired(s)} post(t,p) \tag{1.5}$$

By property of the $\texttt{Inject}_\uparrow$, the $\mathcal{H}$-VHDL rising edge and the stabilize relations, and $\texttt{comp}(id_p,$ $\text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$:

$$\begin{aligned} \sigma'(id_p)(\text{"}sm\text{"}) = \sigma(id_p)(\text{"}sm\text{"}) - \sigma(id_p)(\text{"}s\_output\_token\_sum\text{"}) \\ +\sigma(id_p)(\text{"}s\_input\_token\_sum\text{"}) \end{aligned} \tag{1.6}$$

By the definition of Post Falling Edge State Similarity relation:

$$s.M(p) = \sigma(id_p)(\text{"}sm\text{"}) \tag{1.7}$$

$$\sum_{t \in Fired(s)} pre(p,t) = \sigma(id_p)(\text{"}sots\text{"}) \tag{1.8}$$

$$\sum_{t \in Fired(s)} post(t,p) = \sigma(id_p)(\text{"}sits\text{"}) \tag{1.9}$$

Rewriting the goal with 1.5, 1.6, 1.7, 1.8 and 1.9, tautology .

□

1.5.2 Rising edge and condition combination

Lemma 19 (Rising Edge Equal Condition Combination). *For all* $sitpn$, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' *that verify the hypotheses of Def. 10, then*
$\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,

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

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

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

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

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$.
By property of the $\mathcal{H}$-VHDL stabilize relation, and
$\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_t)(\text{"}scc\text{"}) = \prod_{i=0}^{\Delta(id_t)(\text{"}conditions\_number\text{"})-1} \sigma'(id_t)(\text{"}input\_conditions\text{"})[i] \tag{1.10}$$

Rewriting the goal with 1.10,

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

Case analysis on $conds(t)$ (2 CASES):

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

  $$\boxed{\prod_{i=0}^{\Delta(id_t)("cn")-1} \sigma'(id_t)("ic")[i] = \mathtt{true}.}$$

  By construction, $<\mathtt{conditions\_number} \Rightarrow 1> \in gm_t$ and $<\mathtt{input\_conditions}(0) \Rightarrow \mathtt{true}> \in ipm_t$.

  By property of the stabilize relation, $<\mathtt{conditions\_number} \Rightarrow 1> \in gm_t$ and $<\mathtt{input\_conditions}(0) \Rightarrow \mathtt{true}>$ ipm_t:

  $$\Delta(id_t)("cn") = 1 \tag{1.11}$$
  $$\sigma'(id_t)("ic")[0] = \mathtt{true} \tag{1.12}$$

  Rewriting the goal with 1.11 and 1.12, tautology.

- **CASE** $conds(t) \neq \emptyset$:
  By construction, $<\mathtt{conditions\_number} \Rightarrow |\mathtt{conds(t)}|> \in gm_t$, and by property of the stabilize relation:

  $$\Delta(id_t)("cn") = |conds(t)| \tag{1.13}$$

  Rewriting the goal with (1.13),

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

  Applying Theorem Big Operator Equality, there are two points to prove:

  1. $|conds(t)| = |conds(t)|$
  2. $\exists$ an injection $\iota \in [0, |conds(t)| - 1] \rightarrow conds(t)$ s.t.
     $$\forall i \in [0, |conds(t)| - 1],\ \sigma'(id_t)("ic")[i] = \begin{cases} E_c(\tau, \iota(i)) & if\ \mathbb{C}(t, \iota(i)) = 1 \\ \mathtt{not}(E_c(\tau, \iota(i))) & if\ \mathbb{C}(t, \iota(i)) = -1 \end{cases}$$

  By construction, there exists a bijection $\beta \in [0, |conds(t)| - 1] \rightarrow conds(t)$ such that for all $i \in [0, |conds(t)| - 1]$, there exists an $id_c \in Ins(\Delta)$ and:

  - $\gamma(\beta(i)) = id_c$
  - $\mathbb{C}(t, \beta(i)) = 1$ implies $<\mathtt{input\_conditions(i)} \Rightarrow \mathtt{id_c}> \in ipm_t$
  - $\mathbb{C}(t, \beta(i)) = -1$ implies $<\mathtt{input\_conditions(i)} \Rightarrow \mathtt{not\ id_c}> \in ipm_t$

Let us take such a bijection β to prove the goal. Then, given an $i \in [0, |conds(t)| - 1]$, let us show
$\boxed{\sigma'(id_t)(''ic'')[i] = \begin{cases} E_c(\tau, \beta(i)) & if\ \mathbb{C}(t, \beta(i)) = 1 \\ \texttt{not}(E_c(\tau, \beta(i))) & if\ \mathbb{C}(t, \beta(i)) = -1 \end{cases}}$
By definition of $\beta(i) \in conds(t)$:

$$\mathbb{C}(t, \beta(i)) = 1 \vee \mathbb{C}(t, \beta(i)) = -1 \tag{1.14}$$

Case analysis on (1.14):

- **CASE** $\mathbb{C}(t, \beta(i)) = 1$: $\boxed{\sigma'(id_t)(''ic'')[i] = E_c(\tau, \beta(i))}$

  By property of β, there exists $id_c \in Ins(\Delta)$ s.t. $\gamma(\beta(i)) = id_c$ and $<\texttt{input\_conditions(i)} \Rightarrow \texttt{id}_\texttt{c}> \in ipm_t$.
  By property of the stabilize relation and $<\texttt{input\_conditions(i)} \Rightarrow \texttt{id}_\texttt{c}> \in ipm_t$:

  $$\sigma'(id_t)(''ic'')[i] = \sigma'(id_c) \tag{1.15}$$

  By property of the $\mathcal{H}$-VHDL rising edge and stabilize relations, and $id_c \in Ins(\Delta)$:

  $$\sigma'(id_c) = \sigma_i(id_c) \tag{1.16}$$

  By property of the $\texttt{Inject}_\uparrow$ relation and $id_c \in Ins(\Delta)$:

  $$\sigma_i(id_c) = E_p(\tau, \uparrow)(id_c) \tag{1.17}$$

  By property of $\gamma \vdash E_p \stackrel{env}{=} E_c$:

  $$E_p(\tau, \uparrow)(id_c) = E_c(\tau, c) \tag{1.18}$$

  Rewriting the goal with (1.15), (1.16), (1.17), (1.18), tautology.

- **CASE** $\mathbb{C}(t, c) = -1$: $\boxed{\sigma'(id_t)(''ic'')[i] = \texttt{not}\ E_c(\tau, \beta(i))}$
  By property of β, there exists $id_c \in Ins(\Delta)$ s.t. $\gamma(\beta(i)) = id_c$ and $<\texttt{input\_conditions(i)} \Rightarrow \texttt{not id}_\texttt{c}> \in ipm_t$.
  By property of the stabilize relation and $<\texttt{input\_conditions(i)} \Rightarrow \texttt{not id}_\texttt{c}> \in ipm_t$:

  $$\sigma'(id_t)(''ic'')[i] = \texttt{not}\ \sigma'(id_c) \tag{1.19}$$

  Then, equations (1.16), (1.17) and (1.18) also hold this case.
  Rewriting the goal with (1.19), (1.16), (1.17) and (1.18), tautology.

□

1.5.3 Rising edge and time counters

Lemma 20 (Rising Edge Equal Time Counters). *For all $sitpn$, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Def. 10, then*
$\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,

$(upper(I_s(t)) = \infty \wedge s'.I(t) \leq lower(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"}))$
$\wedge(upper(I_s(t)) = \infty \wedge s'.I(t) > lower(I_s(t)) \Rightarrow \sigma'(id_t)(\text{"}s\_time\_counter\text{"}) = lower(I_s(t)))$
$\wedge(upper(I_s(t)) \neq \infty \wedge s'.I(t) > upper(I_s(t)) \Rightarrow \sigma'(id_t)(\text{"}s\_time\_counter\text{"}) = upper(I_s(t)))$
$\wedge(upper(I_s(t)) \neq \infty \wedge s'.I(t) \leq upper(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"}))$.

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

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

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$.

Then, there are 4 points to show:

1. $\boxed{upper(I_s(t)) = \infty \wedge s'.I(t) \leq lower(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"})}$

   Assuming $upper(I_s(t)) = \infty$ and $s'.I(t) \leq lower(I_s(t))$, let us show $\boxed{s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"}).}$

   By property of the $\texttt{Inject}_{\uparrow}$, $\mathcal{H}$-VHDL rising edge and stabilize relations, and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

   $$\sigma'(id_t)(\text{"}s\_time\_counter\text{"}) = \sigma(id_t)(\text{"}s\_time\_counter\text{"}) \tag{1.20}$$

   By property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

   $$s.I(t) = \sigma(id_t)(\text{"}s\_time\_counter\text{"}) \tag{1.21}$$

   Rewriting the goal with (1.20) and (1.21), tautology.

2. $\boxed{upper(I_s(t)) = \infty \wedge s'.I(t) > lower(I_s(t)) \Rightarrow \sigma'(id_t)(\text{"}s\_time\_counter\text{"}) = lower(I_s(t).}$

   Proved in the same fashion as 1.

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

   Proved in the same fashion as 1.

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

   Proved in the same fashion as 1.

□

1.5.4 Rising edge and reset orders

Lemma 21 (Rising Edge Equal Reset Orders). *For all sitpn, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_{\uparrow}$, σ' that verify the hypotheses of Def. 10, then*
$\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $s'.reset_t(t) = \sigma'(id_t)(\text{"}s\_reinit\_time\_counter\text{"})$

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

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$. By property of the $\mathcal{H}$-VHDL stabilize relation and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_t)(\text{"}srtc\text{"}) = \sum_{i=0}^{\Delta(id_t)(\text{"}input\_arcs\_number\text{"})-1} \sigma'(id_t)(\text{"}reinit\_time\text{"})[i] \tag{1.22}$$

Rewriting the goal with (1.22), $\boxed{s'.reset_t(t) = \sum_{i=0}^{\Delta(id_t)(\text{"}ian\text{"})-1} \sigma'(id_t)(\text{"}rt\text{"})[i].}$

Case analysis on $input(t)$ (2 CASES):

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

  By construction, $<\texttt{input\_arcs\_number} \Rightarrow \texttt{1}> \in gm_t$, and by property of the elaboration relation:

  $$\Delta(id_t)(\text{"}ian\text{"}) = 1 \tag{1.23}$$

  By construction, there exists an $id_{ft} \in Sigs(\Delta)$ s.t. $<\texttt{reinit\_time(0)} \Rightarrow \texttt{id}_{\texttt{ft}}> \in ipm_t$ and $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}}> \in opm_t$, and by property of the $\mathcal{H}$-VHDL stabilize relation and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

  $$\sigma'(id_t)(\text{"}rt\text{"})[0] = \sigma'(id_{ft}) \tag{1.24}$$
  $$\sigma'(id_{ft}) = \sigma'(id_t)(\text{"}fired\text{"}) \tag{1.25}$$
  $$\sigma'(id_t)(\text{"}fired\text{"}) = \sigma'(id_t)(\text{"}s\_fired\text{"}) \tag{1.26}$$
  $$\sigma'(id_t)(\text{"}s\_fired\text{"}) = \sigma'(id_t)(\text{"}s\_firable\text{"}).\sigma'(id_t)(\text{"}s\_priority\_combination\text{"}) \tag{1.27}$$

  Rewriting the goal with (1.24), (1.39), (1.26) and (1.27), $\boxed{s'.reset_t(t) = \sigma'(id_t)(\text{"}s\_firable\text{"}).\sigma'(id_t)(\text{"}s\_priority\_combination\text{"}).}$

  By property of the stabilize relation, and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

  $$\sigma'(id_t)(\text{"}spc\text{"}) = \prod_{i=0}^{\Delta(id_t)(\text{"}ian\text{"})-1} \sigma'(id_t)(\text{"}priority\_authorizations\text{"})[i] \tag{1.28}$$

  By construction, $<\texttt{priority\_authorizations(0)} \Rightarrow \texttt{true}> \in ipm_t$, and by property of the stabilize relation and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

  $$\sigma'(id_t)(\text{"}priority\_authorizations\text{"})[0] = true \tag{1.29}$$

  Rewriting the goal with (1.23), (1.28) and (1.29), and simplifying the equation, $\boxed{s'.reset_t(t) = \sigma'(id_t)(\text{"}s\_firable\text{"}).}$

  Case analysis on $t \in Fired(s)$ or $t \notin Fired(s)$:

  - **CASE** $t \in Fired(s)$:

By property of $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$:

$$s'.reset_t(t) = \texttt{true} \tag{1.30}$$

Rewriting the goal with (1.30), $\boxed{\sigma'(id_t)(\text{"}s\_firable\text{"}) = \texttt{true}.}$

By property of the stabilize, the $\mathcal{H}$-VHDL rising edge and the $\texttt{Inject}_\uparrow$ relations, and $\texttt{comp}(id_t,$ $\text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma(id_t)(\text{"}s\_firable\text{"}) = \sigma'(id_t)(\text{"}s\_firable\text{"}) \tag{1.31}$$

Rewriting the goal with (1.31), $\boxed{\sigma(id_t)(\text{"}s\_firable\text{"}) = \texttt{true}.}$

By property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

$$t \in Firable(s) \Leftrightarrow \sigma(id_t)(\text{"}sfa\text{"}) = \texttt{true} \tag{1.32}$$

Rewriting the goal with (1.32), $\boxed{t \in Firable(s).}$

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 $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$:

  $$s'.reset_t(t) = \texttt{false} \tag{1.33}$$

  Rewriting the goal with (1.33), $\boxed{\sigma'(id_t)(\text{"}s\_firable\text{"}) = \texttt{false}.}$

  By property of the stabilize, the $\mathcal{H}$-VHDL rising edge and the $\texttt{Inject}_\uparrow$ relations, and $\texttt{comp}(id_t,$ $\text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$, equation (1.31) holds.

  Rewriting the goal with (1.31), $\boxed{\sigma(id_t)(\text{"}s\_firable\text{"}) = false.}$

  By property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

  $$t \notin Firable(s) \Leftrightarrow \sigma(id_t)(\text{"}sfa\text{"}) = \texttt{false} \tag{1.34}$$

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

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

By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in gm_t$, and by property of the $\mathcal{H}$-VHDL elaboration relation:

$$\Delta(id_t)(\text{"}ian\text{"}) = |input(t)| \tag{1.35}$$

Rewriting the goal with (1.35), $\boxed{s'.reset_t(t) = \sum_{i=0}^{|input(t)|-1} \sigma'(id_t)(\text{"}rt\text{"})[i].}$

Case analysis on $t \in Fired(s)$ or $t \notin Fired(s)$:

- **CASE** $t \in Fired(s)$:

  By property of $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$, equation (1.30) holds.

  Rewriting the goal with (1.30), $\boxed{\sum_{i=0}^{|input(t)|-1} \sigma'(id_t)(''rt'')[i] = \texttt{true}.}$

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

  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 definition of id_p, there exist gm_p, ipm_p, opm_p s.t. $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$. 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. $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in opm_p$ and $<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in ipm_t$. Let us take such an i, j and id_{ji}, and let us use i to prove the goal: $\boxed{\sigma'(id_t)(''rt'')[i] = \texttt{true}.}$

  By property of the stabilize relation, $<\texttt{reinit\_transition\_time(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in opm_p$ and $<\texttt{reinit\_time(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in ipm_t$:

  $$\sigma'(id_t)(''rt'')[i] = \sigma'(id_{ji}) = \sigma'(id_p)(''rtt'')[j] \tag{1.36}$$

  Rewriting the goal with (1.36), $\boxed{\sigma'(id_p)(''rtt'')[j] = \texttt{true}.}$

  By property of the $\texttt{Inject}_\uparrow$, the $\mathcal{H}$-VHDL rising edge and the stabilize relations:

  $$\begin{aligned} \sigma'(id_p)(''rtt'')[j] = &((\sigma(id_p)(''oat'')[j] = \texttt{BASIC} + \sigma(id_p)(''oat'')[j] = \texttt{TEST}) \\ &.(\sigma(id_p)(''sm'') - \sigma(id_p)(''sots'') < \sigma(id_p)(''oaw'')[j]) \\ &.(\sigma(id_p)(''sots'') > 0)) \\ &+ \sigma(id_p)(''otf'')[j] \end{aligned} \tag{1.37}$$

  Rewriting the goal with (1.37),

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

  By construction, there exists $id_{ft} \in Sigs(\Delta)$ s.t. $<\texttt{output\_transitions\_fired(j)} \Rightarrow \texttt{id}_{\texttt{ft}}> \in ipm_p$ and $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}}> \in opm_t$. By property of state σ as being a stable state:

  $$\sigma(id_t)(''fired'') = \sigma(id_{ft}) = \sigma(id_p)(''otf'')[j] \tag{1.38}$$

Rewriting the goal with (1.38),

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

By property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

$$t \in Fired(s) \Leftrightarrow \sigma(id_t)(''fired'') = \mathtt{true} \tag{1.39}$$

Knowing that $t \in Fired(s)$, we can rewrite the goal with the right side of (1.39) and simplify the goal (i.e, $\forall b \in \mathbb{B},\ b + \mathtt{true} = \mathtt{true}$), 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. $pre(p, t) = (\omega, \mathtt{basic}) \vee pre(p, t) = (\omega, \mathtt{test})$
    3. $s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega$

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

    Assuming that p exists, and by property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

    $$s'.reset_t(t) = \mathtt{true} \tag{1.40}$$

    Rewriting the goal with (1.40), $\boxed{\sum_{i=0}^{|input(t)|-1} \sigma'(id_t)(''rt'')[i] = \mathtt{true}.}$

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

    By construction, there exists $id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$.

    By definition of id_p, there exist gm_p, ipm_p, opm_p s.t. $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$. 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. $<\mathtt{reinit\_transition\_time(j)} \Rightarrow \mathtt{id_{ji}}> \in opm_p$ and $<\mathtt{reinit\_time(i)} \Rightarrow \mathtt{id_{ji}}> \in ipm_t$. Let us take such an i, j and id_{ji}, and let us use i to prove the goal: $\boxed{\sigma'(id_t)(''rt'')[i] = \mathtt{true}.}$

    By property of the stabilize relation, $<\mathtt{reinit\_transition\_time(j)} \Rightarrow \mathtt{id_{ji}}> \in opm_p$ and $<\mathtt{reinit\_time(i)} \Rightarrow \mathtt{id_{ji}}> \in ipm_t$:

    $$\sigma'(id_t)(''rt'')[i] = \sigma'(id_{ji}) = \sigma'(id_p)(''rtt'')[j] \tag{1.41}$$

Rewriting the goal with (1.41), $\boxed{\sigma'(id_p)(''rtt'')[j] = \mathtt{true}.}$
By property of the $\mathtt{Inject}_{\uparrow}$, the $\mathcal{H}$-VHDL rising edge and the stabilize relations:

$$\begin{aligned}\sigma'(id_p)(''rtt'')[j] =&((\sigma(id_p)(''oat'')[j] = \mathtt{BASIC} + \sigma(id_p)(''oat'')[j] = \mathtt{TEST})\\ &.(\sigma(id_p)(''sm'') - \sigma(id_p)(''sots'') < \sigma(id_p)(''oaw'')[j])\\ &.(\sigma(id_p)(''sots'') > 0))\\ &+ \sigma(id_p)(''otf'')[j]\end{aligned} \tag{1.42}$$

Rewriting the goal with (1.42),

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

By construction, $<\mathtt{output\_arcs\_types(j)} \Rightarrow \mathtt{BASIC}> \in ipm_p$ and $<\mathtt{output\_arcs\_weights(j)} \Rightarrow \omega> \in ipm_p$.
By property of the stabilize relation and $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_p)(''oat'')[j] = \mathtt{BASIC} \tag{1.43}$$
$$\sigma'(id_p)(''oaw'')[j] = \omega \tag{1.44}$$

By property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

$$\sigma(id_p)(''sm'') = s.M(p) \tag{1.45}$$
$$\sigma(id_p)(''sots'') = \sum_{t_i \in Fired(s)} pre(p, t_i) \tag{1.46}$$

Rewriting the goal with (1.43), (1.44), (1.45) and (1.46), and simplifying the goal:

$$\boxed{(s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega \; . \sum_{t_i \in Fired(s)} pre(p, t_i) > 0)) + \sigma(id_t)(''fired'') = \mathtt{true}}$$

Thanks to the hypotheses 1 and 3:

$$s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega = \mathtt{true} \tag{1.47}$$
$$\sum_{t_i \in Fired(s)} pre(p, t_i) > 0 = \mathtt{true} \tag{1.48}$$
$$\tag{1.49}$$

Rewriting the goal with (1.47) and (1.48), and simplifying the goal, tautology.

* **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 $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

$$s'.reset_t(t) = \texttt{false} \tag{1.50}$$

Rewriting the goal with (1.50): $\boxed{\sum_{i=0}^{|input(t)|-1} \sigma'(id_t)(''rt'')[i] = \texttt{false}.}$

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

By construction, there exist a $p \in input(t)$, an $id_p \in Comps(\Delta)$, gm_p, ipm_p, opm_p, a $j \in [0, |output(p)| - 1]$, an $id_{ji} \in Sigs(\Delta)$ s.t. $\gamma(p) = id_p$ and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$ and $<\texttt{reinit\_transition\_time}(\texttt{j}) \Rightarrow \texttt{id}_{\texttt{ji}}> \in opm_p$ and $<\texttt{reinit\_time}(\texttt{i}) \Rightarrow \texttt{id}_{\texttt{ji}}> \in ipm_t$. Let us take such a p, id_p, gm_p, ipm_p, opm_p, j and id_{ji}.
By property of the stabilize relation, $<\texttt{reinit\_transition\_time}(\texttt{j}) \Rightarrow \texttt{id}_{\texttt{ji}}> \in opm_p$ and $<\texttt{reinit\_time}(\texttt{i}) \Rightarrow \texttt{id}_{\texttt{ji}}> \in ipm_t$:

$$\sigma'(id_t)(''rt'')[i] = \sigma'(id_{ji}) = \sigma'(id_p)(''rtt'')[j] \tag{1.51}$$

Rewriting the goal with (1.51): $\boxed{\sigma'(id_p)(''rtt'')[j] = \texttt{false}.}$

By property of the $\texttt{Inject}_{\uparrow}$, the $\mathcal{H}$-VHDL rising edge and the stabilize relations:

$$\begin{aligned}\sigma'(id_p)(''rtt'')[j] =& ((\sigma(id_p)(''oat'')[j] = \texttt{BASIC} + \sigma(id_p)(''oat'')[j] = \texttt{TEST})\\ &.(\sigma(id_p)(''sm'') - \sigma(id_p)(''sots'') < \sigma(id_p)(''oaw'')[j])\\ &.(\sigma(id_p)(''sots'') > 0))\\ &+ \sigma(id_p)(''otf'')[j]\end{aligned} \tag{1.52}$$

Rewriting the goal with (1.52),

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

By construction, there exists $id_{ft} \in Sigs(\Delta)$ s.t. $<\texttt{output\_transitions\_fired}(\texttt{j}) \Rightarrow \texttt{id}_{\texttt{ft}}> \in ipm_p$ and $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}}> \in opm_t$. By property of state σ as being a stable state:

$$\sigma(id_t)(''fired'') = \sigma(id_{ft}) = \sigma(id_p)(''otf'')[j] \tag{1.53}$$

Rewriting the goal with (1.53),

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

By property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

$$t \notin Fired(s) \Leftrightarrow \sigma(id_t)(''fired'') = \texttt{false} \tag{1.54}$$

Knowing that $t \notin Fired(s)$, we can rewrite the goal with the right side of (1.54) and simplify the goal (i.e, $\forall b \in \mathbb{B},\ b + \texttt{false} = b$):

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

Then, there are two cases:

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

   By property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

   $$\sum_{t_i \in Fired(s)} pre(p, t_i) = \sigma(id_p)(''sots'') \tag{1.55}$$

   Rewriting the goal with (1.55) and $\sum\limits_{t_i \in Fired(s)} pre(p, t_i) = 0$, simplifying the goal: tautology.

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

   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, $<\texttt{output\_arcs\_weights(j)} \Rightarrow \omega> \in ipm_p$.
   By property of stable state σ and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

   $$\sigma(id_p)(''oaw'')[j] \quad = \quad \omega \tag{1.56}$$

   By property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

   $$\sigma(id_p)(''sm'') \quad = \quad s.M(p) \tag{1.57}$$
   $$\sigma(id_p)(''sots'') \quad = \quad \sum_{t_i \in Fired(s)} pre(p, t_i) \tag{1.58}$$

By hypothesis, we know that $s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) \geq \omega$, and then we can deduce:

$$s.M(p) - \sum_{t_i \in Fired(s)} pre(p, t_i) < \omega = \texttt{false} \tag{1.59}$$

Rewriting the goal with (1.56), (1.57), (1.58), and (1.59), and simplifying the goal, tautology.

(b) **CASE** $pre(p,t) = (\omega, \texttt{inhib})$:
By construction, $<\texttt{output\_arcs\_types(j)} \Rightarrow \texttt{INHIB}> \in ipm_p$.
By property of stable state σ and $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma(id_p)("oat")[j] = \texttt{INHIB} \tag{1.60}$$

Rewriting the goal with (1.60), and simplifying the goal, tautology.

□

1.5.5 Rising edge and action executions

Lemma 22 (Rising Edge Equal Action Executions). *For all sitpn, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Def. 10, 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 $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$:

$$s.ex(a) = s'.ex(a) \tag{1.61}$$

By construction, id_a is an output port identifier of boolean type in the $\mathcal{H}$-VHDL design d assigned by the ‘‘action’’ process only during a falling edge phase.
By property of the $\mathcal{H}$-VHDL $\texttt{Inject}_\uparrow$, rising edge, stabilize relations, and the ‘‘action’’ process:

$$\sigma(id_a) = \sigma'(id_a) \tag{1.62}$$

Rewriting the goal with (1.61) and (1.62), $\boxed{s.ex(a) = \sigma(id_a).}$

By property of $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$, $s.ex(a) = \sigma(id_a)$. □

1.5.6 Rising edge and function executions

Lemma 23 (Rising Edge Equal Function Executions). *For all sitpn, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Def. 10, 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 $E_c, \tau \vdash s \xrightarrow{\uparrow} s'$:

$$s'.ex(f) = \sum_{t \in Fired(s)} \mathbb{F}(t, f) \tag{1.63}$$

By construction, the ‘‘function’’ process is a part of design d's behavior, i.e
$\mathtt{ps}("function", \emptyset, sl, ss) \in d.cs$.
By construction id_f is an output port of design d, and it is only assigned in the body of the ‘‘function’’ process. Let $trs(f)$ be the set of transitions associated to function f, i.e $trs(f) = \{t \in T \mid \mathbb{F}(t, f) = true\}$. Then, depending on $trs(f)$, there are two cases of assignment of output port id_f:

- **CASE** $trs(f) = \emptyset$:
  By construction, $\mathtt{id_f} \Leftarrow \mathtt{false} \in ss_\uparrow$ where $ss_\uparrow$ is the part of the ‘‘function’’ process body executed during the rising edge phase.

  By property of the $\mathcal{H}$-VHDL rising edge, the stabilize relations and $\mathtt{ps}("function", \emptyset, sl, ss) \in d.cs$:
  $$\sigma'(id_f) = false \tag{1.64}$$
  By property of $\sum\limits_{t \in Fired(s)} \mathbb{F}(t, f)$ and $trs(f) = \emptyset$:
  $$\sum_{t \in Fired(s)} \mathbb{F}(t, f) = \mathtt{false} \tag{1.65}$$
  Rewriting the goal with (1.63), (1.64) and (1.65), tautology.

- **CASE** $trs(f) \neq \emptyset$:
  By construction, $\mathtt{id_f} \Leftarrow \mathtt{id_{ft_0}} + \cdots + \mathtt{id_{ft_n}} \in ss_\uparrow$, where $id_{ft_i} \in Sigs(\Delta)$, $ss_\uparrow$ is the part of the ‘‘function’’ process body executed during the rising edge phase, and $n = |trs(f)| - 1$.

  By property of the $\mathtt{Inject}_\uparrow$, the $\mathcal{H}$-VHDL rising edge, the stabilize relations, and
  $\mathtt{ps}("function", \emptyset, sl, ss) \in d.cs$:
  $$\sigma'(id_f) = \sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}) \tag{1.66}$$
  Rewriting the goal with (1.63) and (1.66), $\boxed{\sum\limits_{t \in Fired(s)} \mathbb{F}(t, f) = \sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}).}$

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

  - **CASE** $\sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}) = \mathtt{true}$:
    Then, we can rewrite the goal as follows: $\boxed{\sum\limits_{t \in Fired(s)} \mathbb{F}(t, f) = \mathtt{true}.}$

    To prove the above goal, let us show $\boxed{\exists t \in Fired(s)\ s.t.\ \mathbb{F}(t, f) = \mathtt{true}.}$
    Knowing that $\sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}) = \mathtt{true}$, then $\exists id_{ft_i}\ s.t.\ \sigma(id_{ft_i}) = \mathtt{true}$. Let us take such an id_{ft_i}.
    By construction, for all id_{ft_i}, there exist a $t_i \in trs(f)$, an $id_{t_i} \in Comps(\Delta)$, gm_{t_i}, ipm_{t_i} and opm_{t_i} s.t. $\gamma(t_i) = id_{t_i}$ and $\mathtt{comp}(id_{t_i}, "transition", gm_{t_i}, ipm_{t_i}, opm_{t_i}) \in d.cs$ and $<\mathtt{fired} \Rightarrow \mathtt{id_{ft_i}}> \in opm_{t_i}$. Let us take such a t_i, id_{t_i}, gm_{t_i}, ipm_{t_i} and opm_{t_i}.
    By property of σ as being a stable design state, and $\mathtt{comp}(id_{t_i}, "transition", gm_{t_i}, ipm_{t_i}, opm_{t_i}) \in d.cs$:
    $$\sigma(id_{t_i})("fired") = \sigma(id_{ft_i}) \tag{1.67}$$

Thanks to (1.67) and $\sigma(id_{ft_i}) = \texttt{true}$, we can deduce that $\sigma(id_{t_i})(\text{"}fired\text{"}) = \texttt{true}$.

By property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$:

$$t_i \in Fired(s) \Leftrightarrow \sigma(id_{t_i})(\text{"}fired\text{"}) = \texttt{true} \tag{1.68}$$

Thanks to (1.68), we can deduce $t_i \in Fired(s)$.

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

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

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

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

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

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, for all $t \in T$ s.t. $\mathbb{F}(t,f) = \texttt{true}$, there exist an $id_t \in Comps(\Delta)$, gm_t, ipm_t, opm_t and $id_{ft_i} \in Sigs(\Delta)$ s.t. $\gamma(t) = id_t$ and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$ and $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}_\texttt{i}}> \in opm_t$. Let us take such a id_t, gm_t, ipm_t, opm_t and id_{ft_i}.
  By property of stable design state σ and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$, equation (1.67) holds.
  By property of $\gamma \vdash s \overset{\downarrow}{\sim} \sigma$, equation (1.68) holds.
  Thanks to (1.67) and (1.68), we can deduce that $\sigma(id_{ft_i}) = \texttt{true}$.
  Then, $\sigma(id_{ft_i}) = \texttt{true}$ contradicts $\sigma(id_{ft_0}) + \cdots + \sigma(id_{ft_n}) = \texttt{false}.$

□

1.5.7 Rising edge and sensitization

Lemma 24 (Rising Edge Equal Sensitized). *For all sitpn, d, γ, E_c, E_p, τ, Δ, σ_e, s, s′, σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Def. 10, then*
$\forall t \in T, id_t \in Comps(\Delta) \text{ s.t. } \gamma(t) = id_t, t \in Sens(s'.M) \Leftrightarrow \sigma'(id_t)(\text{"}s\_enabled\text{"}) = \texttt{true}.$

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

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$. Then, the proof is in two parts:

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

   By property of the stabilize relation and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_t)(\text{"}se\text{"}) = \prod_{i=0}^{\Delta(id_t)(\text{"}ian\text{"})-1} \sigma'(id_t)(\text{"}input\_arcs\_valid\text{"})[i] \tag{1.69}$$

Rewriting the goal with (1.69), $\boxed{\prod_{i=0}^{\Delta(id_t)("ian")-1} \sigma'(id_t)("iav")[i] = \texttt{true}.}$

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

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

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

- **CASE** $input(t) = \emptyset$:
  By construction, $<\texttt{input\_arcs\_number} \Rightarrow \texttt{1}> \in gm_t$ and
  $<\texttt{input\_arcs\_valid}(0) \Rightarrow \texttt{true}> \in ipm_t$.
  By property of the elaboration and stabilize relations and
  $\texttt{comp}(id_t, "transition", gm_t, ipm_t, opm_t) \in d.cs$:

$$\Delta(id_t)("ian") = 1 \tag{1.70}$$
$$\sigma'(id_t)("iav")[0] = \texttt{true} \tag{1.71}$$

  Thanks to (1.70), we can deduce that $i = 0$. Rewriting the goal with (1.71), tautology.

- **CASE** $input(t) \neq \emptyset$:
  By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in gm_t$.
  By property of the elaboration relation and $\texttt{comp}(id_t, "transition", gm_t, ipm_t, opm_t) \in d.cs$:

$$\Delta(id_t)("ian") = |input(t)| \tag{1.72}$$

  Thanks to (1.72), we know that $i \in [0, |input(t)| - 1]$.
  By construction, there exist a $p \in input(t)$, $id_p \in Comps(\Delta)$, gm_p, ipm_p, opm_p, $j \in [0, |output(p)| - 1]$ and $id_{ji} \in Sigs(\Delta)$ s.t. $\gamma(p) = id_p$ and
  $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$ and $<\texttt{output\_arcs\_valid(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in opm_p$
  and $<\texttt{input\_arcs\_valid(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in ipm_t$.
  By property of the stabilize relation, $\texttt{comp}(id_t, "transition", gm_t, ipm_t, opm_t) \in d.cs$ and $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_t)("iav")[i] = \sigma'(id_{ji}) = \sigma'(id_p)("oav")[j] \tag{1.73}$$

  Rewriting the goal with (1.73), $\boxed{\sigma'(id_p)("oav")[j] = \texttt{true}.}$

  By property of the stabilize relation and $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$:

$$\begin{aligned}\sigma'(id_p)("oav")[j] = &((\sigma'(id_p)("oat")[j] = \texttt{BASIC} + \sigma'(id_p)("oat")[j] = \texttt{TEST}) \\ &\ .\ \sigma'(id_p)("sm") \geq \sigma'(id_p)("oaw")[j]) \\ &+ (\sigma'(id_p)("oat")[j] = \texttt{INHIB}\ .\ \sigma'(id_p)("sm") < \sigma'(id_p)("oaw")[j])\end{aligned} \tag{1.74}$$

Rewriting the goal with (1.74),

$$\boxed{\begin{aligned}\mathtt{true} = & ((\sigma'(id_p)(''oat'')[j] = \mathtt{BASIC} + \sigma'(id_p)(''oat'')[j] = \mathtt{TEST}) \\ & \quad . \, \sigma'(id_p)(''sm'') \geq \sigma'(id_p)(''oaw'')[j]) \\ & + (\sigma'(id_p)(''oat'')[j] = \mathtt{INHIB} \, . \, \sigma'(id_p)(''sm'') < \sigma'(id_p)(''oaw'')[j])\end{aligned}}$$

Let us perform case analysis on $pre(p, t)$; there are 3 cases:

- **CASE** $pre(p, t) = (\omega, \mathtt{BASIC})$:

  By construction, $<\mathtt{output\_arcs\_types(j)} \Rightarrow \mathtt{BASIC}> \in ipm_p$ and $<\mathtt{output\_arcs\_weights(j)} \Rightarrow \omega> \in ipm_p$.
  By property of the stabilize relation and $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

  $$\sigma'(id_p)(''oat'')[j] = \mathtt{BASIC} \tag{1.75}$$
  $$\sigma'(id_p)(''oaw'')[j] = \omega \tag{1.76}$$

  Rewriting the goal with (1.75) and (1.76), and simplifying the goal:
  $\boxed{\sigma'(id_p)(''sm'') \geq \omega = \mathtt{true}.}$
  Appealing to Lemma Rising Edge Equal Marking:

  $$s'.M(p) = \sigma'(id_p)(''sm'') \tag{1.77}$$

  Rewriting the goal with (1.77): $\boxed{s'.M(p) \geq \omega = \mathtt{true}.}$

  By definition of $t \in Sens(s'.M)$, $s'.M(p) \geq \omega = \mathtt{true}.$ [1]

- **CASE** $pre(p, t) = (\omega, \mathtt{TEST})$: same as the preceding case.
- **CASE** $pre(p, t) = (\omega, \mathtt{INHIB})$:
  By construction, $<\mathtt{output\_arcs\_types(j)} \Rightarrow \mathtt{INHIB}> \in ipm_p$ and $<\mathtt{output\_arcs\_weights(j)} \Rightarrow \omega> \in ipm_p$.
  By property of the stabilize relation and $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

  $$\sigma'(id_p)(''oat'')[j] = \mathtt{INHIB} \tag{1.78}$$
  $$\sigma'(id_p)(''oaw'')[j] = \omega \tag{1.79}$$

  Rewriting the goal with (1.78) and (1.79), and simplifying the goal:
  $\boxed{\sigma'(id_p)(''sm'') < \omega = \mathtt{true}.}$
  Appealing to Lemma Rising Edge Equal Marking, equation (1.77) holds.
  Rewriting the goal with (1.77): $\boxed{s'.M(p) < \omega = \mathtt{true}.}$

  By definition of $t \in Sens(s'.M)$, $s'.M(p) < \omega = \mathtt{true}.$

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

[1] Here $\geq$ denotes a boolean operator, i.e $\geq \in \mathbb{N} \to \mathbb{N} \to \mathbb{B}$. As the $\geq \subseteq (\mathbb{N} \times \mathbb{B})$ relation is decidable for all pairs of natural numbers, we can interchange an expression $a \geq b = \mathtt{true}$ with $a \geq b$ where $a, b \in \mathbb{N}$.

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

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

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 $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$, equation (1.69) holds.
Rewriting $\sigma'(id_t)(\text{"}se\text{"}) = \texttt{true}$ with (1.69), $\prod_{i=0}^{\Delta(id_t)(\text{"}ian\text{"})-1} \sigma'(id_t)(\text{"}input\_arcs\_valid\text{"})[i] = \texttt{true}$.
Then, we can deduce that $\forall i \in [0, \Delta(id_t)(\text{"}ian\text{"}) - 1]$, $\sigma'(id_t)(\text{"}iav\text{"})[i] = \texttt{true}$.
By construction, there exist an $id_p \in Comps(\Delta)$, gm_p, ipm_p, opm_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
$\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$ and $<\texttt{output\_arcs\_valid(j)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in opm_p$ and $<\texttt{input\_arcs\_valid(i)} \Rightarrow \texttt{id}_{\texttt{ji}}> \in ipm_t$. Let us take such an $id_p \in Comps(\Delta)$, gm_p, ipm_p, opm_p, $i \in [0, |input(t)| - 1]$, $j \in [0, |output(p)| - 1]$ and $id_{ji} \in Sigs(\Delta)$.
By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in gm_t$.
By property of the elaboration relation and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$, equation (1.72) holds.
Thanks to (1.72), we can deduce that $\forall i \in [0, |input(t)| - 1]$, $\sigma'(id_t)(\text{"}iav\text{"})[i] = \texttt{true}$.
Having such an $i \in [0, |input(t)| - 1]$, we can deduce that $\sigma'(id_t)(\text{"}iav\text{"})[i] = \texttt{true}$.
By property of the stabilize relation, $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$ and $\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$, equation (1.73) holds.
Thanks to (1.73), we can deduce that $\sigma'(id_p)(\text{"}oav\text{"})[j] = \texttt{true}$.
By property of the stabilize relation and $\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$, equation (1.74) holds. Thanks to (1.74), we can deduce that:

$$\begin{aligned}\texttt{true} = &((\sigma'(id_p)(\text{"}oat\text{"})[j] = \texttt{BASIC} + \sigma'(id_p)(\text{"}oat\text{"})[j] = \texttt{TEST}) \\ &\quad . \ \sigma'(id_p)(\text{"}sm\text{"}) \geq \sigma'(id_p)(\text{"}oaw\text{"})[j]) \\ &\quad + (\sigma'(id_p)(\text{"}oat\text{"})[j] = \texttt{INHIB} \ . \ \sigma'(id_p)(\text{"}sm\text{"}) < \sigma'(id_p)(\text{"}oaw\text{"})[j])\end{aligned}$$

By construction, $<\texttt{output\_arcs\_types(j)} \Rightarrow \texttt{BASIC}> \in ipm_p$ and
$<\texttt{output\_arcs\_weights(j)} \Rightarrow \omega> \in ipm_p$.
By property of the stabilize relation and $\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$, equations (1.75) and (1.76) hold.
Thanks to (1.75) and (1.76), we can deduce that $\sigma'(id_p)(\text{"}sm\text{"}) \geq \omega = \texttt{true}$.
Appealing to Lemma Rising Edge Equal Marking, $s'.M(p) \geq \omega$.

(b) Assuming $pre(p,t) = (\omega, \texttt{inhib})$, let us show $\boxed{s'.M(p) < \omega.}$
The proceeding is the same as the preceding case. Here, we will start the proof where the two cases are diverging, i.e:
By construction, <`output_arcs_types(j)` $\Rightarrow$ `INHIB`> $\in ipm_p$ and <`output_arcs_weights(j)` $\Rightarrow \omega$> $\in ipm_p$.
By property of the stabilize relation and $\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$, equations (1.78) and (1.76) hold.
Thanks to (1.78) and (1.76), we can deduce that $\sigma'(id_p)(\text{"}sm\text{"}) < \omega = \texttt{true}$.
Appealing to Lemma Rising Edge Equal Marking, $s'.M(p) < \omega$.

□

Lemma 25 (Rising Edge Equal Not Sensitized). *For all sitpn, d, γ, E_c, E_p, τ, Δ, σ_e, s, s', σ, σ_i, $\sigma_\uparrow$, σ' that verify the hypotheses of Def. 10, then*
$\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t, t \notin Sens(s'.M) \Leftrightarrow \sigma'(id_t)(\text{"}s\_enabled\text{"}) = \texttt{false}$.

Proof. Proving the above lemma is trivial by appealing to Lemma Rising Edge Equal Sensitized and by reasoning on contrapositives. □

1.6 Falling Edge

Definition 11 (Falling Edge Hypotheses). *Given an sitpn $\in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$, $\Delta \in ElDesign(d, \mathcal{D}_\mathcal{H})$, $E_p \in (\mathbb{N} \times \{\uparrow,\downarrow\}) \to Ins(\Delta) \to value$, $\tau \in \mathbb{N}$, $s, s' \in S(sitpn)$, $\sigma_e, \sigma, \sigma_i, \sigma_\downarrow, \sigma' \in \Sigma(\Delta)$, assume that:*

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

Lemma 26 (Falling Edge). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\gamma \vdash s' \overset{\downarrow}{\sim} \sigma'$.*

Proof. By definition of Post Falling Edge State Similarity, there are 12 points to prove.

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

2. $\forall t \in T_i, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,
$(upper(I_s(t)) = \infty \wedge s'.I(t) \leq lower(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"}))$
$\wedge(upper(I_s(t)) = \infty \wedge s'.I(t) > lower(I_s(t)) \Rightarrow \sigma'(id_t)(\text{"}s\_time\_counter\text{"}) = lower(I_s(t)))$
$\wedge(upper(I_s(t)) \neq \infty \wedge s'.I(t) > upper(I_s(t)) \Rightarrow \sigma'(id_t)(\text{"}s\_time\_counter\text{"}) = upper(I_s(t)))$
$\wedge(upper(I_s(t)) \neq \infty \wedge s'.I(t) \leq upper(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"}))$.

3. $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $s'.reset_t(t) = \sigma'(id_t)(''s\_reinit\_time\_counter'')$.
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. $\gamma(f) = id_f$, $s'.ex(f) = \sigma'(id_f)$.
7. $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, t \in Firable(s') \Leftrightarrow \sigma'(id_t)(''s\_firable'') = \texttt{true}$.
8. $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, t \notin Firable(s') \Leftrightarrow \sigma'(id_t)(''s\_firable'') = \texttt{false}$.
9. $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, t \in Fired(s') \Leftrightarrow \sigma'(id_t)(''fired'') = \texttt{true}$.
10. $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, t \notin Fired(s') \Leftrightarrow \sigma'(id_t)(''fired'') = \texttt{false}$.
11. $\forall p \in P, id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$, $\sum\limits_{t \in Fired(s')} pre(p,t) = \sigma'(id_p)(''s\_output\_token\_sum'')$.
12. $\forall p \in P, id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$, $\sum\limits_{t \in Fired(s')} post(t,p) = \sigma'(id_p)(''s\_input\_token\_sum'')$.

Each point is proved by a separate lemma:

- Apply Lemma Falling Edge Equal Marking to solve 1.
- Apply Lemma Falling Edge Equal Time Counters to solve 2.
- Apply Lemma Falling Edge Equal Reset Orders to solve 3.
- Apply Lemma Falling Edge Equal Condition Values to solve 4.
- Apply Lemma Falling Edge Equal Action Executions to solve 5.
- Apply Lemma Falling Edge Equal Function Executions to solve 6.
- Apply Lemma Falling Edge Equal Firable to solve 7.
- Apply Lemma Falling Edge Equal Not Firable to solve 8.
- Apply Lemma Falling Edge Equal Fired to solve 9.
- Apply Lemma Falling Edge Equal Not Fired to solve 10.
- Apply Lemma Falling Edge Equal Output Token Sum to solve 11.
- Apply Lemma Falling Edge Equal Input Token Sum to solve 12.

□

1.6.1 Falling Edge and marking

Lemma 27 (Falling Edge Equal Marking). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\forall p \in P, id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$, $s'.M(p) = \sigma'(id_p)(''s\_marking'')$.*

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

By definition of $E_c, \tau \vdash sitpn, s \xrightarrow{\downarrow} s'$:

$$s.M(p) = s'.M(p) \tag{1.80}$$

By property of the $\mathtt{Inject}_{\downarrow}$ relation, the $\mathcal{H}$-VHDL falling edge relation, the stabilize relation and $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_p)(''s\_marking'') = \sigma(id_p)(''s\_marking'') \tag{1.81}$$

Rewriting the goal with (1.80) and (1.81): $\boxed{s.M(p) = \sigma(id_p)(''s\_marking'').}$

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

□

Lemma 28 (Falling Edge Equal Output Token Sum). *For all $sitpn$, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_{\downarrow}$, σ' that verify the hypotheses of Def. 11, then $\forall p, id_p$ s.t. $\gamma(p) = id_p$, $\sum\limits_{t \in Fired(s')} pre(p,t) = \sigma'(id_p)(''s\_output\_token\_sum'')$.*

Proof. Given a $p \in P$ and an $id_p \in Comps(\Delta)$, let us show $\boxed{\sum\limits_{t \in Fired(s')} pre(p,t) = \sigma'(id_p)(''s\_output\_token\_sum'').}$

By definition of id_p, there exist gm_p, ipm_p, opm_p s.t. $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$. By property of the stabilize relation and $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_p)(''sots'') = \sum_{i=0}^{\Delta(id_p)(''oan'')-1} \begin{cases} \sigma'(id_p)(''oaw'')[i] \text{ if } (\sigma'(id_p)(''otf'')[i] \\ \qquad . \, \sigma'(id_p)(''oat'')[i] = \mathtt{BASIC}) \\ 0 \text{ otherwise} \end{cases} \tag{1.82}$$

Rewriting the goal with (1.82):

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

Let us unfold the definition of the left sum term:

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

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

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

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

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

   By construction, $<\texttt{output\_arcs\_number} \Rightarrow \texttt{1}> \in gm_p$, $<\texttt{output\_arcs\_types(0)} \Rightarrow \texttt{BASIC}> \in ipm_p$, $<\texttt{output\_transitions\_fired(0)} \Rightarrow \texttt{true}> \in ipm_p$, and $<\texttt{output\_arcs\_weights(0)} \Rightarrow \texttt{0}> \in ipm_p$.
   By property of the elaboration relation and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

   $$\Delta(id_p)(''oan'') = 1 \tag{1.83}$$

   By property of the stabilize relation and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

   $$\sigma'(id_p)(''oat'')[0] = \texttt{BASIC} \tag{1.84}$$
   $$\sigma'(id_p)(''otf'')[0] = \texttt{true} \tag{1.85}$$
   $$\sigma'(id_p)(''oaw'')[0] = 0 \tag{1.86}$$

   By property of $output(p) = \emptyset$:

   $$\sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } pre(p,t) = (\omega, \texttt{basic}) \\ 0 \textit{ otherwise} \end{cases} = 0 \tag{1.87}$$

   Rewriting the goal with (1.83), (1.84), (1.85), (1.86) and (1.87), tautology.

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

   By construction, $<\texttt{output\_arcs\_number} \Rightarrow |output(p)|> \in gm_p$, and by property of the elaboration relation:

   $$\Delta(id_p)(''oan'') = |output(p)| \tag{1.88}$$

   Rewriting the goal with (1.88): $\boxed{\sum_{t \in Fired(s')} f(t) = \sum_{i=0}^{|output(p)|-1} g(i).}$

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

   - **BASE CASE:**

In that case, $0 > |output| - 1$ and $\sum_{i=0}^{|output(p)|-1} g(i) = 0$.

As $0 > |output| - 1$, then $|output(p)| = 0$, thus contradicting $output(p) \neq \emptyset$.

- **INDUCTION CASE**:

In that case, $0 \leq |output(p)| - 1$.

$$\forall F \subseteq Fired(s'), g(0) + \sum_{t \in F} f(t) = g(0) + \sum_{i=1}^{|output(p)|-1} g(i)$$

$$\boxed{\sum_{t \in Fired(s')} f(t) = g(0) + \sum_{i=1}^{|output(p)|-1} g(i)}$$

By definition of g:

$$g(0) = \begin{cases} \sigma'(id_p)(\text{"}oaw\text{"})[0] \texttt{ if } (\sigma'(id_p)(\text{"}otf\text{"})[0] \\ \qquad\qquad\qquad . \, \sigma'(id_p)(\text{"}oat\text{"})[0] = \texttt{BASIC}) \\ 0 \textit{ otherwise} \end{cases} \tag{1.89}$$

Let us perform case analysis on the value of $\sigma'(id_p)(\text{"}otf\text{"})[0] \,.\, \sigma'(id_p)(\text{"}oat\text{"})[0] = \texttt{BASIC}$; there are two cases:

(a) $(\sigma'(id_p)(\text{"}otf\text{"})[0] \,.\, \sigma'(id_p)(\text{"}oat\text{"})[0] = \texttt{BASIC}) = \texttt{false}$:
In that case, $g(0) = 0$, and then we can apply the induction hypothesis with $F = Fired(s')$ to solve the goal: $\sum_{t \in Fired(s')} f(t) = \sum_{i=1}^{|output(p)|-1} g(i)$.

(b) $(\sigma'(id_p)(\text{"}otf\text{"})[0] \,.\, \sigma'(id_p)(\text{"}oat\text{"})[0] = \texttt{BASIC}) = \texttt{true}$:
In that case, $g(0) = \sigma'(id_p)(\text{"}oaw\text{"})[0]$, $\sigma'(id_p)(\text{"}otf\text{"})[0] = \texttt{true}$ and $\sigma'(id_p)(\text{"}oat\text{"})[0] = \texttt{BASIC}$.
By construction, there exist a $t \in output(t)$, $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$. Let us take such a $t \in output(p)$.
By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$.
As $t \in output(p)$, there exist $\omega \in \mathbb{N}^*$ and $a \in \{\texttt{BASIC}, \texttt{TEST}, \texttt{INHIB}\}$ s.t. $pre(p,t) = (\omega, a)$.
Let us take an ω and a s.t. $pre(p,t) = (\omega, a)$.
By construction, $<\texttt{output\_arcs\_types}(0) \Rightarrow a> \in ipm_p$, $<\texttt{output\_arcs\_weights}(0) \Rightarrow \omega> \in ipm_p$, and there exists $id_{ft} \in Sigs(\Delta)$ s.t. $<\texttt{fired} \Rightarrow id_{ft}> \in opm_t$ and $<\texttt{output\_transitions\_fired}(0) \Rightarrow \texttt{id}_\texttt{ft}> \in ipm_p$
By property of the stabilize relation, $\sigma'(id_p)(\text{"}oat\text{"})[0] = \texttt{BASIC}$ and $<\texttt{output\_arcs\_types}(0) \Rightarrow \texttt{a}> \in ipm_p$:

$$pre(p,t) = (\omega, \texttt{basic}) \tag{1.90}$$

By property of the stabilize relation, $<\texttt{fired} \Rightarrow \texttt{id}_\texttt{ft}> \in opm_t$, $<\texttt{output\_transitions\_fired}(0) \Rightarrow \texttt{id}_\texttt{ft}> \in ipm_p$ and $\sigma'(id_p)(\text{"}otf\text{"})[0] = \texttt{true}$:

$$\sigma'(id_t)(\text{"}fired\text{"}) = \texttt{true} \tag{1.91}$$

Appealing to Lemma 39, we know $t \in Fired(s')$.
As $t \in Fired(s')$, we can rewrite the left sum term of the goal as follows:

$$\boxed{f(t) + \sum_{t' \in Fired(s') \setminus \{t\}} f(t') = g(0) + \sum_{i=1}^{|output(p)|-1} g(i)}$$

We know that $g(0) = \sigma'(id_p)("oaw")[0]$, and by property of the stabilize relation and $<\mathtt{output\_arcs\_weights}(0) \Rightarrow \omega> \in ipm_p$:

$$\sigma'(id_p)("oaw")[0] = \omega \tag{1.92}$$

Rewriting the goal with (1.92):

$$\boxed{f(t) + \sum_{t' \in Fired(s') \setminus \{t\}} f(t') = \omega + \sum_{i=1}^{|output(p)|-1} g(i)}$$

By definition of f, and as $pre(p,t) = (\omega, \mathtt{basic})$, then $f(t) = \omega$; thus, rewriting the goal:

$$\boxed{\omega + \sum_{t' \in Fired(s') \setminus \{t\}} f(t') = \omega + \sum_{i=1}^{|output(p)|-1} g(i)}$$

Then, knowing that $g(0) = \omega$, we can apply the induction hypothesis with $F = Fired(s') \setminus \{t\}$: $g(0) + \sum_{t' \in Fired(s') \setminus \{t\}} f(t') = g(0) + \sum_{i=1}^{|output(p)|-1} g(i)$.

$\square$

Lemma 29 (Falling Edge Equal Input Token Sum)**.** *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, 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

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

By definition of id_p, there exist gm_p, ipm_p, opm_p s.t. $\mathtt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$.
By property of the stabilize relation and $\mathtt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_p)("sits") = \sum_{i=0}^{\Delta(id_p)("ian")-1} \begin{cases} \sigma'(id_p)("iaw")[i] \text{ } \mathtt{if} \text{ } \sigma'(id_p)("itf")[i] \\ 0 \text{ } otherwise \end{cases} \tag{1.93}$$

Rewriting the goal with (1.93):

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

Let us unfold the definition of the left sum term:

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

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

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

   By construction, $<\texttt{input\_arcs\_number} \Rightarrow \texttt{1}> \in gm_p$, $<\texttt{input\_transitions\_fired}(\texttt{0}) \Rightarrow \texttt{true}> \in ipm_p$, and $<\texttt{input\_arcs\_weights}(\texttt{0}) \Rightarrow \texttt{0}> \in ipm_p$.
   By property of the elaboration relation and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

   $$\Delta(id_p)(''ian'') = 1 \tag{1.94}$$

   By property of the stabilize relation and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

   $$\sigma'(id_p)(''itf'')[0] = \texttt{true} \tag{1.95}$$
   $$\sigma'(id_p)(''iaw'')[0] = 0 \tag{1.96}$$

   By property of $input(p) = \emptyset$:

   $$\sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } post(t,p) = \omega \\ 0 \textit{ otherwise} \end{cases} = 0 \tag{1.97}$$

   Rewriting the goal with (1.94), (1.95), (1.96), and (1.97), and simplifying the goal, tautology.

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

   By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(p)|> \in gm_p$, and by property of the elaboration relation:

   $$\Delta(id_p)(''ian'') = |input(p)| \tag{1.98}$$

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

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

Rewriting the goal with (1.98): $\boxed{\sum_{t\in Fired(s')} f(t) = \sum_{i=0}^{|input(p)|-1} g(i).}$

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

- **BASE CASE**:

  In that case, $0 > |input(p)| - 1$ and $\sum_{i=0}^{|input(p)|-1} g(i) = 0$.

  As $0 > |input(p)| - 1$, then $|input(p)| = 0$, thus contradicting $input(p) \neq \emptyset$.

- **INDUCTION CASE**:

  In that case, $0 \leq |input(p)| - 1$.

  $$\forall F \subseteq Fired(s'),\ g(0) + \sum_{t\in F} f(t) = g(0) + \sum_{i=1}^{|input(p)|-1} g(i)$$

  $$\boxed{\sum_{t\in Fired(s')} f(t) = g(0) + \sum_{i=1}^{|input(p)|-1} g(i)}$$

  By definition of g:

  $$g(0) = \begin{cases} \sigma'(id_p)("iaw")[0] \texttt{ if } \sigma'(id_p)("itf")[0] \\ 0\ otherwise \end{cases} \tag{1.99}$$

  Let us perform case analysis on the value of $\sigma'(id_p)("itf")[0]$; there are two cases:

  (a) $\sigma'(id_p)("itf")[0] = \texttt{false}$:

  In that case, $g(0) = 0$, and then we can apply the induction hypothesis with $F = Fired(s')$ to solve the goal: $\sum_{t\in Fired(s')} f(t) = \sum_{i=1}^{|input(p)|-1} g(i).$

  (b) $\sigma'(id_p)("itf")[0] = \texttt{true}$:

  In that case, $g(0) = \sigma'(id_p)("iaw")[0]$ and $\sigma'(id_p)("itf")[0] = \texttt{true}$.

  By construction, there exist a $t \in input(t)$, $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$. Let us take such a $t \in input(p)$.

  By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, "transition", gm_t, ipm_t, opm_t) \in d.cs$.

  As $t \in input(p)$, there exist $\omega \in \mathbb{N}^*$ s.t. $post(t,p) = \omega$. Let us take an ω s.t. $post(t,p) = \omega$.

  By construction, $<\texttt{input\_arcs\_weights}(0) \Rightarrow \omega> \in ipm_p$, and there exists $id_{ft} \in Sigs(\Delta)$ s.t. $<\texttt{fired} \Rightarrow id_{ft}> \in opm_t$ and $<\texttt{input\_transitions\_fired}(0) \Rightarrow \texttt{id}_{\texttt{ft}}> \in ipm_p$

  By property of the stabilize relation and $<\texttt{input\_arcs\_types}(0) \Rightarrow \texttt{a}> \in ipm_p$:

  $$post(t,p) = \omega \tag{1.100}$$

  By property of the stabilize relation, $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}}> \in opm_t$, $<\texttt{input\_transitions\_fired}(0) \Rightarrow \texttt{id}_{\texttt{ft}}> \in ipm_p$ and $\sigma'(id_p)("itf")[0] = \texttt{true}$:

  $$\sigma'(id_t)("fired") = \texttt{true} \tag{1.101}$$

Appealing to Lemma 39 and (1.101), we know $t \in Fired(s')$.
As $t \in Fired(s')$, we can rewrite the left sum term of the goal as follows:

$$\boxed{f(t) + \sum_{t' \in Fired(s') \setminus \{t\}} f(t') = g(0) + \sum_{i=1}^{|input(p)|-1} g(i)}$$

We know that $g(0) = \sigma'(id_p)(\text{"}iaw\text{"})[0]$, and by property of the stabilize relation and $<\texttt{input\_arcs\_weights}(0) \Rightarrow \omega> \in ipm_p$:

$$\sigma'(id_p)(\text{"}iaw\text{"})[0] = \omega \tag{1.102}$$

Rewriting the goal with (1.102):

$$\boxed{f(t) + \sum_{t' \in Fired(s') \setminus \{t\}} f(t') = \omega + \sum_{i=1}^{|input(p)|-1} g(i)}$$

By definition of f, and as $post(t, p) = \omega$, then $f(t) = \omega$; thus, rewriting the goal:

$$\boxed{\omega + \sum_{t' \in Fired(s') \setminus \{t\}} f(t') = \omega + \sum_{i=1}^{|input(p)|-1} g(i)}$$

Then, knowing that $g(0) = \omega$, we can apply the induction hypothesis with $F = Fired(s') \setminus \{t\}$: $g(0) + \sum_{t' \in Fired(s') \setminus \{t\}} f(t') = g(0) + \sum_{i=1}^{|input(p)|-1} g(i)$.

□

1.6.2 Falling edge and time counters

Lemma 30 (Falling Edge Equal Time Counters). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s′, σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,*
$(upper(I_s(t)) = \infty \wedge s'.I(t) \leq lower(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"}))$
$\wedge (upper(I_s(t)) = \infty \wedge s'.I(t) > lower(I_s(t)) \Rightarrow \sigma'(id_t)(\text{"}s\_time\_counter\text{"}) = lower(I_s(t)))$
$\wedge (upper(I_s(t)) \neq \infty \wedge s'.I(t) > upper(I_s(t)) \Rightarrow \sigma'(id_t)(\text{"}s\_time\_counter\text{"}) = upper(I_s(t)))$
$\wedge (upper(I_s(t)) \neq \infty \wedge s'.I(t) \leq upper(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"}))$.

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

$$\boxed{\begin{aligned}&(upper(I_s(t)) = \infty \wedge s'.I(t) \leq lower(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"}))\\ &\wedge (upper(I_s(t)) = \infty \wedge s'.I(t) > lower(I_s(t)) \Rightarrow \sigma'(id_t)(\text{"}s\_time\_counter\text{"}) = lower(I_s(t)))\\ &\wedge (upper(I_s(t)) \neq \infty \wedge s'.I(t) > upper(I_s(t)) \Rightarrow \sigma'(id_t)(\text{"}s\_time\_counter\text{"}) = upper(I_s(t)))\\ &\wedge (upper(I_s(t)) \neq \infty \wedge s'.I(t) \leq upper(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"}))\end{aligned}}$$

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$.

By property of the elaboration, $\texttt{Inject}_\downarrow$, $\mathcal{H}$-VHDL rising edge and stabilize relations, and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

$$\begin{aligned}\sigma(id_t)(\text{"}se\text{"}) = \texttt{true} \wedge \Delta(id_t)(\text{"}tt\text{"}) \neq \texttt{NOT\_TEMPORAL} \wedge \sigma(id_t)(\text{"}srtc\text{"}) = \texttt{false}\\ \wedge \sigma(id_t)(\text{"}stc\text{"}) < \Delta(id_t)(\text{"}mtc\text{"}) \Rightarrow \sigma'(id_t)(\text{"}stc\text{"}) = \sigma(id_t)(\text{"}stc\text{"}) + 1\end{aligned} \tag{1.103}$$

$$\begin{aligned}\sigma(id_t)(''se'') = \texttt{true} \wedge \Delta(id_t)(''tt'') \neq \texttt{NOT\_TEMPORAL} \wedge \sigma(id_t)(''srtc'') = \texttt{false} \\ \wedge \sigma(id_t)(''stc'') \geq \Delta(id_t)(''mtc'') \Rightarrow \sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'')\end{aligned} \tag{1.104}$$

$$\begin{aligned}\sigma(id_t)(''se'') = \texttt{true} \wedge \Delta(id_t)(''tt'') \neq \texttt{NOT\_TEMPORAL} \\ \wedge \sigma(id_t)(''srtc'') = \texttt{true} \Rightarrow \sigma'(id_t)(''stc'') = 1\end{aligned} \tag{1.105}$$

$$\sigma(id_t)(''se'') = \texttt{false} \vee \Delta(id_t)(''tt'') = \texttt{NOT\_TEMPORAL} \Rightarrow \sigma'(id_t)(''stc'') = 0 \tag{1.106}$$

Then, there are 4 points to show:

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

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

   Case analysis on $t \in Sens(s.M)$; there are two cases:

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

      By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(''se'') = \texttt{false}$ (1.107).
      Appealing to (1.106) and (1.107), we have $\sigma'(id_t)(''stc'') = 0$ (1.108).
      By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = 0$ (1.109).
      Rewriting the goal with (1.108) and (1.109): tautology.

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

      By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(''se'') = \texttt{true}$ (1.110).
      By construction, and as $upper(I_s(t)) = \infty$, `<transition_type ⇒ TEMP_A_INF>` $\in gm_t$. By property of the elaboration relation, we have $\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_INF}$ (1.111).
      Case analysis on $s.reset_t(t)$; there are two cases:

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

         By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, $\sigma(id_t)(''srtc'') = \texttt{true}$ (1.112).
         Appealing to (1.105), (1.110), (1.111) and (1.112), we have $\sigma'(id_t)(''stc'') = 1$ (1.113).
         By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = 1$ (1.114).
         Rewriting the goal with (1.113) and (1.114): tautology.

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

         By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(''srtc'') = \texttt{false}$ (1.115).
         As $upper(I_s(t)) = \infty$, 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, `<maximal_time_counter ⇒` a`>` $\in gm_t$, and by property of the elaboration relation, we have $\Delta(id_t)(''mtc'') = a$ (1.116).
         By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, and knowing that $t \in Sens(s.M)$, $s.reset_t(t) = \texttt{false}$ and $upper(I_s(t)) = \infty$:

$$s'.I(t) = s.I(t) + 1 \tag{1.117}$$

Rewriting the goal with (1.117): $\boxed{s.I(t) + 1 = \sigma'(id_t)(''stc'').}$
We assumed that $s'.I(t) \leq lower(I_s(t))$, and as $s'.I(t) = s.I(t) + 1$, then $s.I(t) + 1 \leq lower(I_s(t))$, then $s.I(t) < lower(I_s(t))$, then $s.I(t) < a$ since $a = lower(I_s(t))$.
By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, and knowing that $s.I(t) < lower(I_s(t))$ and $upper(I_s(t)) = \infty$:

$$s.I(t) = \sigma(id_t)(''stc'') \tag{1.118}$$

Appealing to (1.116), (1.118) and $s.I(t) < a$:

$$\sigma(id_t)(''stc'') < \Delta(id_t)(''mtc'') \tag{1.119}$$

Appealing to (1.103), (1.119), (1.115) and (1.110):

$$\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'') + 1 \tag{1.120}$$

Rewriting the goal with (1.120) and (1.118): tautology.

2. $\boxed{upper(I_s(t)) = \infty \land s'.I(t) > lower(I_s(t)) \Rightarrow \sigma'(id_t)(''s\_time\_counter'') = lower(I_s(t).}$
Assuming that $upper(I_s(t)) = \infty$ and $s'.I(t) > lower(I_s(t))$, let us show $\boxed{\sigma'(id_t)(''s\_time\_counter'') = lower(I_s(t)).}$
As $upper(I_s(t)) = \infty$, 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, $<\texttt{maximal\_time\_counter} \Rightarrow a> \in gm_t$, and $<\texttt{transition\_type} \Rightarrow \texttt{TEMP\_A\_INF}> \in gm_t$ by property of the elaboration relation:

$$\Delta(id_t)(''mtc'') = a \tag{1.121}$$
$$\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_INF} \tag{1.122}$$

Case analysis on $t \in Sens(s.M)$:

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

(b) $t \in Sens(s.M)$:
By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$ and $t \in Sens(s.M)$:

$$\sigma(id_t)(''se'') = \texttt{true} \tag{1.123}$$

Case analysis on $s.reset_t(t)$; there are two cases:

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

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

By property of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$ and $s.reset_t(t) = \texttt{false}$:

$$\sigma(id_t)(\text{"}srtc\text{"}) = \texttt{false} \tag{1.124}$$

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, and knowing that $s'.I(t) > lower(I_s(t))$:

$$\begin{aligned} s'.I(t) = s.I(t) + 1 &\Rightarrow s.I(t) + 1 > lower(I_s(t)) \\ &\Rightarrow s.I(t) \geq lower(I_s(t)) \end{aligned} \tag{1.125}$$

Case analysis on $s.I(t) \geq lower(I_s(t))$:

A. $s.I(t) > lower(I_s(t))$: $\boxed{\sigma'(id_t)(\text{"}stc\text{"}) = lower(I_s(t)).}$

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

$$\sigma(id_t)(\text{"}stc\text{"}) = lower(I_s(t)) \tag{1.126}$$

Appealing to (1.104):

$$\sigma'(id_t)(\text{"}stc\text{"}) = \sigma(id_t)(\text{"}stc\text{"}) \tag{1.127}$$

Rewriting the goal with (1.126) and (1.127): tautology.

B. $s.I(t) = lower(I_s(t))$: $\boxed{\sigma'(id_t)(\text{"}stc\text{"}) = lower(I_s(t)).}$

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

$$s.I(t) = \sigma(id_t)(\text{"}stc\text{"}) \tag{1.128}$$

Appealing to (1.104):

$$\sigma'(id_t)(\text{"}stc\text{"}) = \sigma(id_t)(\text{"}stc\text{"}) \tag{1.129}$$

Rewriting the goal with (1.129), (1.128) and $s.I(t) = lower(I_s(t))$: tautology.

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

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

As $upper(I_s(t)) \neq \infty$, 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, there exists $tt \in \{\texttt{TEMP\_A\_A}, \texttt{TEMP\_A\_B}\}$ s.t. $<\texttt{maximal\_time\_counter} \Rightarrow b> \in gm_t$, and $<\texttt{transition\_type} \Rightarrow tt> \in gm_t$; by property of the elaboration relation:

$$\Delta(id_t)(\text{"}mtc\text{"}) \;=\; b = upper(I_s(t)) \tag{1.130}$$

$$\Delta(id_t)(\text{"}tt\text{"}) \;\neq\; \texttt{NOT\_TEMP} \tag{1.131}$$

Case analysis on $t \in Sens(s.M)$:

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

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

Contradicts $s'.I(t) > upper(I_s(t))$.

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

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$ and $t \in Sens(s.M)$:

$$\sigma(id_t)(''se'') = \texttt{true} \tag{1.132}$$

Case analysis on $s.reset_t(t)$; there are two cases:

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

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

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

By property of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$ and $s.reset_t(t) = \texttt{false}$:

$$\sigma(id_t)(''srtc'') = \texttt{false} \tag{1.133}$$

Case analysis on $s.I(t) > upper(I_s(t))$ or $s.I(t) \leq upper(I_s(t))$:

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

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

$$s'.I(t) = s.I(t) \tag{1.134}$$

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

$$\sigma(id_t)(''stc'') = upper(I_s(t)) \tag{1.135}$$

Appealing to (1.104), we have $\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'')$.
Rewriting the goal with $\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'')$ and (1.135): tautology.

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

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

$$s.I(t) = \sigma(id_t)(''stc'') \tag{1.136}$$

Case analysis on $s.I(t) \leq upper(I_s(t))$; there are two cases:

- $s.I(t) = upper(I_s(t))$:

  Appealing to (1.130), (1.136) and $s.I(t) = upper(I_s(t))$:

  $$\Delta(id_t)(''mtc'') \leq \sigma(id_t)(''stc'') \tag{1.137}$$

  Appealing to (1.137) and (1.104):

  $$\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'') \tag{1.138}$$

Rewriting the goal with (1.138), (1.136) and $s.I(t) = upper(I_s(t))$: tautology.

- $s.I(t) < upper(I_s(t))$:

  By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$:

  $$s'.I(t) = s.I(t) + 1 \tag{1.139}$$

  From (1.139) and $s.I(t) < upper(I_s(t))$, we can deduce $s'.I(t) \leq upper(I_s(t))$; contradicts $s'.I(t) > upper(I_s(t))$.

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

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

As $upper(I_s(t)) \neq \infty$, 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, there exists $tt \in \{\texttt{TEMP\_A\_A}, \texttt{TEMP\_A\_B}\}$ s.t. $<\texttt{maximal\_time\_counter} \Rightarrow b> \in gm_t$, and $<\texttt{transition\_type} \Rightarrow tt> \in gm_t$; by property of the elaboration relation:

$$\Delta(id_t)(''mtc'') = b = upper(I_s(t)) \tag{1.140}$$
$$\Delta(id_t)(''tt'') \neq \texttt{NOT\_TEMP} \tag{1.141}$$

Case analysis on $t \in Sens(s.M)$:

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

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(''se'') = \texttt{false}$ (1.142).
Appealing (1.106) and (1.142), we have $\sigma'(id_t)(''stc'') = 0$ (1.143).

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = 0$ (1.144).
Rewriting the goal with (1.143) and (1.144): tautology.

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

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(''se'') = \texttt{true}$ (1.145).
Case analysis on $s.reset_t(t)$:

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

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(''srtc'') = \texttt{true}$ (1.146).
Appealing to (1.105), (1.141), (1.145) and (1.146), we have $\sigma'(id_t)(''stc'') = 1$ (1.147).

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = 1$ (1.148).
Rewriting the goal with (1.147) and (1.148), tautology.

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

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(''srtc'') = \texttt{false}$ (1.149).
Case analysis on $s.I(t) > upper(I_s(t))$ or $s.I(t) \leq upper(I_s(t))$:

A. $s.I(t) > upper(I_s(t))$:

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

B. $s.I(t) \leq upper(I_s(t))$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $s.I(t) = \sigma(id_t)(''stc'')$ (1.150).

- $s.I(t) < upper(I_s(t))$:
  From $s.I(t) < upper(I_s(t))$, (1.150) and (1.140), we can deduce $\sigma(id_t)(''stc'') < \Delta(id_t)(''mtc'')$ (1.151).
  From (1.103), (1.145), (1.141), (1.149) and (1.151), we can deduce:

$$\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'') + 1 \tag{1.152}$$

  By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$:

$$s'.I(t) = s.I(t) + 1 \tag{1.153}$$

  Rewriting the goal with (1.152) and (1.153), tautology.

- $s.I(t) = upper(I_s(t))$:
  By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we know that $s'.I(t) = s.I(t) + 1$. We assumed that $s'.I(t) \leq upper(I_s(t))$; thus, $s.I(t) + 1 \leq upper(I_s(t))$.
  Contradicts $s.I(t) = upper(I_s(t))$.

□

1.6.3 Falling edge and reset orders

Lemma 31 (Falling Edge Equal Reset Orders). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $s'.reset_t(t) = \sigma'(id_t)(''s\_reinit\_time\_counter'')$.*

Proof. Given a $t \in T_i$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show $\boxed{s'.reset_t(t) = \sigma'(id_t)(''srtc'').}$
By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$.
By property of the stabilize relation and $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_t)(''srtc'') = \sum_{i=0}^{\Delta(id_t)(''ian'')-1} \sigma'(id_t)(''rt'')[i] \tag{1.154}$$

□

1.6.4 Falling edge and condition values

Lemma 32 (Falling Edge Equal Condition Values). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, 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 $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.cond(c) = E_c(\tau, c)$ (1.155).
By property of the $\texttt{Inject}_\downarrow$, the $\mathcal{H}$-VHDL falling edge, the stabilize relations and $id_c \in Ins(\Delta)$, we have $\sigma'(id_c) = E_p(\tau, \downarrow)(id_c)$ (1.156).

Rewriting the goal with (1.155) and (1.156): $\boxed{E_c(\tau, c) = E_p(\tau, \downarrow)(id_c)}$

By definition of $\gamma \vdash E_p \overset{env}{=} E_c$: $E_c(\tau, c) = E_p(\tau, \downarrow)(id_c)$.

□

1.6.5 Falling and action executions

Lemma 33 (Falling Edge Equal Action Executions). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, 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 $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$:

$$s'.ex(a) = \sum_{p \in marked(s.M)} \mathbb{A}(p, a) \tag{1.157}$$

By construction, the ‘‘`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. $\texttt{ps}("action", \emptyset, sl, ss) \in d.cs$.
By construction id_a is only assigned in the body of the ‘‘`action`’’ process. 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 the falling edge phase.

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

  $$\sigma'(id_a) = false \tag{1.158}$$

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

  $$\sum_{p \in marked(s.M)} \mathbb{A}(p, a) = \texttt{false} \tag{1.159}$$

  Rewriting the goal with (1.157), (1.158) and (1.159), tautology.

- **CASE** $pls(a) \neq \emptyset$:
  By construction, $\texttt{id}_\texttt{a} \Leftarrow \texttt{id}_{\texttt{mp}_0} + \cdots + \texttt{id}_{\texttt{mp}_\texttt{n}} \in ss_{a\downarrow}$, 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 $\texttt{Inject}_\downarrow$, the $\mathcal{H}$-VHDL falling edge, the stabilize relations, and $\texttt{ps}("action", \emptyset, sl, ss) \in d.cs$:

  $$\sigma'(id_a) = \sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}) \tag{1.160}$$

  Rewriting the goal with (1.157) and (1.160), $\boxed{\sum_{p \in marked(s.M)} \mathbb{A}(p, a) = \sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}).}$

  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, for all id_{mp_i}, there exist a $p_i \in pls(a)$, an $id_{p_i} \in Comps(\Delta)$, gm_{p_i}, ipm_{p_i} and opm_{p_i} s.t. $\gamma(p_i) = id_{p_i}$ and $\texttt{comp}(id_{p_i}, "place", gm_{p_i}, ipm_{p_i}, opm_{p_i}) \in d.cs$ and $<\texttt{marked} \Rightarrow \texttt{id}_{\texttt{mp}_\texttt{i}}> \in opm_{p_i}$. Let us take such a p_i, id_{p_i}, gm_{p_i}, ipm_{p_i} and opm_{p_i}.
  By property of stable σ, and $\texttt{comp}(id_{p_i}, "place", gm_{p_i}, ipm_{p_i}, opm_{p_i}) \in d.cs$:

$$\sigma(id_{mp_i}) = \sigma(id_{p_i})("marked") \tag{1.161}$$
$$\sigma(id_{p_i})("marked") = \sigma(id_{p_i})("sm") > 0 \tag{1.162}$$

  From (1.161), (1.162) and $\sigma(id_{mp_i}) = \texttt{true}$, we can deduce that $\sigma(id_{p_i})("marked") = \texttt{true}$ and $(\sigma(id_{p_i})("sm") > 0) = \texttt{true}$.
  By property of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$:

$$s.M(p_i) = \sigma(id_{p_i})("sm") \tag{1.163}$$

  From (1.163) and $(\sigma(id_{p_i})("sm") > 0) = \texttt{true}$, we can deduce $p_i \in marked(s.M)$, i.e $s.M(p_i) > 0$.
  Let us use p_i to prove the goal: $\boxed{\mathbb{A}(p,a) = \texttt{true}.}$
  By definition of $p_i \in pls(a)$, $\mathbb{A}(p,a) = \texttt{true}.$
- **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, for all $p \in P$ s.t. $\mathbb{A}(p,a) = \texttt{true}$, there exist an $id_p \in Comps(\Delta)$, gm_{tp}, ipm_p, opm_p and $id_{mp_i} \in Sigs(\Delta)$ s.t. $\gamma(p) = id_p$ and $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$ and $<\texttt{marked} \Rightarrow \texttt{id}_{\texttt{mp}_\texttt{i}}> \in opm_p$. Let us take such a id_p, gm_p, ipm_p, opm_p and id_{mp_i}.
    By property of stable σ and $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma(id_{mp_i}) = \sigma(id_p)("marked") \tag{1.164}$$
$$\sigma(id_p)("marked") = \sigma(id_p)("sm") > 0 \tag{1.165}$$

    From $\sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}) = \texttt{false}$, we can deduce $\sigma(id_p)("marked") = \texttt{false}$, and thus that $(\sigma(id_p)("sm") > 0) = \texttt{false}$.

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $s.M(p) = \sigma(id_p)("sm")$, 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, $s.M(p) > 0$).

$\square$

1.6.6 Falling edge and function executions

Lemma 34 (Falling Edge Equal Function Executions). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, 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 $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$:

$$s.ex(f) = s'.ex(f) \tag{1.166}$$

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 a rising edge phase.
By property of the $\mathcal{H}$-VHDL $\texttt{Inject}_\uparrow$, rising edge, stabilize relations, and the ‘‘`function`’’ process:

$$\sigma(id_f) = \sigma'(id_f) \tag{1.167}$$

Rewriting the goal with (1.166) and (1.167), $\boxed{s.ex(f) = \sigma(id_f).}$

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, $s.ex(f) = \sigma(id_f)$. $\square$

1.6.7 Falling edge and firable transitions

Lemma 35 (Falling Edge Equal Firable). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $t \in Firable(s') \Leftrightarrow \sigma'(id_t)("s\_firable") = \texttt{true}$.*

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

The proof is in two parts:

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

   Apply Lemma Falling Edge Equal Firable 1 to solve the goal.

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

   Apply Lemma Falling Edge Equal Firable 2 to solve the goal.

$\square$

Lemma 36 (Falling Edge Equal Firable 1). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $t \in Firable(s') \Rightarrow \sigma'(id_t)("s\_firable") = \texttt{true}$.*

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

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, "transition", gm_t, ipm_t, opm_t) \in d.cs$. By property of the $Inject_{\downarrow}$, the $\mathcal{H}$-VHDL falling edge, the stabilize relations and $\texttt{comp}(id_t, "transition", gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_t)("sfa") = \sigma(id_t)("se") \,.\, \sigma(id_t)("scc") \,.\, \texttt{checktc}(\Delta(id_t), \sigma(id_t)) \tag{1.168}$$

Let us define term $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$ as follows:

$$\begin{aligned}
\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \Big(&\texttt{not}\ \sigma(id_t)("srtc") \,. \\
&\big[(\Delta(id_t)("tt") = \texttt{TEMP_A_B} \,.\, (\sigma(id_t)("stc") \geq \sigma(id_t)("A") - 1) \\
&\qquad\qquad\qquad\qquad\qquad .\, (\sigma(id_t)("stc") \leq \sigma(id_t)("B") - 1)) \\
&+ (\Delta(id_t)("tt") = \texttt{TEMP_A_A} \,.\, (\sigma(id_t)("stc") = \sigma(id_t)("A") - 1)) \\
&+ (\Delta(id_t)("tt") = \texttt{TEMP_A_INF} \,.\, (\sigma(id_t)("stc") \geq \sigma(id_t)("A") - 1))\big]\Big) \\
+ &\,(\sigma(id_t)("srtc") \,.\, \Delta(id_t)("tt") \neq \texttt{NOT_TEMP} \,.\, \sigma(id_t)("A") = 1) \\
+ &\,\Delta(id_t)("tt") = \texttt{NOT_TEMP}
\end{aligned} \tag{1.169}$$

Rewriting the goal with (1.168): $\boxed{\sigma(id_t)("se") \,.\, \sigma(id_t)("scc") \,.\, \texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}.}$
Then, there are three points to prove:

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

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

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

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

$$\sigma(id_t)("scc") = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} \tag{1.170}$$

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

   Rewriting the goal with (1.170): $\boxed{\prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} = \texttt{true}.}$

   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**: *true* = *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, and 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 $\mathbb{C}(t,c) = 1 \lor \mathbb{C}(t,c) = -1$:

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

By definition of $t \in Firable(s')$, we can deduce that $s'.cond(c) = \texttt{true}$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, 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 $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, 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 $t \notin T_i \lor s'.I(t) \in I_s(t)$. Let us perform case analysis on $t \notin T_i \lor s'.I(t) \in I_s(t)$:

(a) $t \notin T_i$:

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

(b) $s'.I(t) \in I_s(t)$:

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. $<\texttt{transition\_type} \Rightarrow tt> \in gm_t$. By property of the elaboration relation, we have $\Delta(id_t)(''tt'') = tt$, and thus, we know $\Delta(id_t)(''tt'') \neq$

NOT_TEMP. Therefore, we can simplfy the term $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$ as follows:

$$\begin{aligned}\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \Big(&\texttt{not}\ \sigma(id_t)(''srtc'')\ .\\ &\big[\big(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B}\ .\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1)\\ &\qquad .\ (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1)\big)\\ &+(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_A}\ .\\ &\quad(\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1))\\ &+(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_INF}\ .\\ &\quad(\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1))\big]\Big)\\ &+\big(\sigma(id_t)(''srtc'')\ .\ \sigma(id_t)(''A'') = 1\big)\end{aligned} \tag{1.171}$$

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $s.reset_t(t) = \sigma(id_t)(''srtc'')$.
Let us perform case analysis on the value $s.reset_t(t)$:

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

   Then, from $s.reset_t(t) = \sigma(id_t)(''srtc'')$, we can deduce that $\sigma(id_t)(''srtc'') = \texttt{true}$. From $\sigma(id_t)(''srtc'') = \texttt{true}$, we can simplify the term $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$ as follows:

$$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \big(\sigma(id_t)(''A'') = 1\big) \tag{1.172}$$

   Rewriting the goal with (1.172), and simplifying the goal: $\boxed{\sigma(id_t)(''A'') = 1.}$

   By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, 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, $<\texttt{time\_A\_value} \Rightarrow a> \in ipm_t$, and by property of stable σ, we have $\sigma(id_t)(''A'') = a = 1$.

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

   Then, from $s.reset_t(t) = \sigma(id_t)(''srtc'')$, we can deduce that $\sigma(id_t)(''srtc'') = \texttt{false}$. From $\sigma(id_t)(''srtc'') = \texttt{false}$, we can simplify the term $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$ as follows:

$$\begin{aligned}&\texttt{checktc}(\Delta(id_t), \sigma(id_t))\\ &=\\ &\big(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B}\ .\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1)\\ &\qquad .\ (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1)\big)\\ &+(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_A}\ .\ (\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1))\\ &+(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_INF}\ .\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1))\end{aligned} \tag{1.173}$$

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

- $I_s(t) = [a, b]$ where $a, b \in \mathbb{N}^*$; then, either $a = b$ or $a \neq b$:
  - $a = b$:
    Then, we have $I_s(t) = [a, a]$, and by construction <`transition_type` $\Rightarrow$ `TEMP_A_A`> $\in gm_t$. By property of the elaboration relation, we have $\Delta(id_t)(''tt'') =$ `TEMP_A_A`; thus we can simplify the term `checktc` as follows:

    $$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = (\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1) \tag{1.174}$$

    Rewriting the goal with (1.174), and simplifying the goal:
    $\boxed{\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1.}$
    From $s'.I(t) \in [a, a]$, we can deduce that $s'.I(t) = a$. Let us perform case analysis on $s.I(t) < upper(I_s(t))$ or $s.I(t) \geq upper(I_s(t))$:
    * $s.I(t) < upper(I_s(t))$:
      By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $s.I(t) = \sigma(id_t)(''stc'')$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, 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, <`time_A_value` $\Rightarrow a$> $\in ipm_t$, and by property of stable σ, we have $\sigma(id_t)(''A'') = a$.
      Rewriting the goal with $\sigma(id_t)(''A'') = a$ and $s.I(t) = \sigma(id_t)(''stc'')$:
      $\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1.$
    * $s.I(t) \geq upper(I_s(t))$:
      In the case where $s.I(t) > upper(I_s(t))$, then $s.I(t) > a$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s.I(t) = s'.I(t) = a$. Then, $a > a$ is a contradiction.

      In the case where $s.I(t) = upper(I_s(t))$, then $s.I(t) = a$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, 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.
  - $a \neq b$:
    Then, we have $I_s(t) = [a, b]$, and by construction <`transition_type` $\Rightarrow$ `TEMP_A_B`> $\in gm_t$. By property of the elaboration relation, we have $\Delta(id_t)(''tt'') =$ `TEMP_A_B`; thus we can simplify the term `checktc` as follows:

    $$\begin{gathered}\texttt{checktc}(\Delta(id_t), \sigma(id_t)) \\ = \\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1) \,.\, (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1)\end{gathered} \tag{1.175}$$

    Rewriting the goal with (1.175), and simplifying the goal:
    $\boxed{(\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1) \wedge (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1).}$

    Let us perform case analysis on $s.I(t) < upper(I_s(t))$ or $s.I(t) \geq upper(I_s(t))$:
    * $s.I(t) < upper(I_s(t))$:
      By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $s.I(t) = \sigma(id_t)(''stc'')$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, 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` $\Rightarrow a> \in ipm_t$ and `<time_B_value` $\Rightarrow b> \in ipm_t$, and by property of stable σ, we have $\sigma(id_t)(''A'') = a$ and $\sigma(id_t)(''B'') = b$.
Rewriting the goal with $\sigma(id_t)(''A'') = a$, $\sigma(id_t)(''B'') = b$ and $s.I(t) = \sigma(id_t)(''stc'')$:
$a - 1 \leq s.I(t) \wedge s.I(t) \leq b - 1$.

* $s.I(t) \geq upper(I_s(t))$:
  In the case where $s.I(t) > upper(I_s(t))$, then $s.I(t) > b$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s.I(t) = s'.I(t) = b$. Then, $b > b$ is a contradiction.

  In the case where $s.I(t) = upper(I_s(t))$, then $s.I(t) = b$. By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, 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.

- $I_s(t) = [a, \infty]$ where $a \in \mathbb{N}^*$:
  By construction `<transition_type` $\Rightarrow$ `TEMP_A_INF>` $\in gm_t$. By property of the elaboration relation, we have $\Delta(id_t)(''tt'') =$ `TEMP_A_INF`; thus we can simplify the term `checktc` as follows:

$$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1)) \tag{1.176}$$

  Rewriting the goal with (1.176), and simplifying the goal:
  $\boxed{\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1.}$
  From $s'.I(t) \in [a, \infty]$, we can deduce $a \leq s'.I(t)$. Then, let us perform case analysis on $s.I(t) \leq lower(I_s(t))$ or $s.I(t) > lower(I_s(t))$:
  - $s.I(t) \leq lower(I_s(t))$:
    By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $s.I(t) = \sigma(id_t)(''stc'')$.
    By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = s.I(t) + 1$:
    $\Rightarrow a \leq s'.I(t)$
    $\Rightarrow a \leq s.I(t) + 1$
    $\Rightarrow a - 1 \leq s.I(t)$
    By construction, `<time_A_value` $\Rightarrow a> \in ipm_t$, and by property of stable σ, we have $\sigma(id_t)(''A'') = a$.
    Rewriting the goal with $\sigma(id_t)(''A'') = a$ and $s.I(t) = \sigma(id_t)(''stc'')$:
    $a - 1 \leq s.I(t)$.
  - $s.I(t) > lower(I_s(t))$:
    By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(''stc'') = lower(I_s(t)) = a$.
    By construction, `<time_A_value` $\Rightarrow a> \in ipm_t$, and by property of stable σ, we have $\sigma(id_t)(''A'') = a$.
    Rewriting the goal with $\sigma(id_t)(''stc'') = a$ and $\sigma(id_t)(''A'') = a$: $a - 1 \leq a$.

$\square$

Lemma 37 (Falling Edge Equal Firable 2). *For all $sitpn$, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $\sigma'(id_t)(''s\_firable'') = \mathtt{true} \Rightarrow t \in Firable(s')$.*

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

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\mathtt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$. By property of the $Inject_\downarrow$, the $\mathcal{H}$-VHDL falling edge, the stabilize relations and $\mathtt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_t)(''sfa'') = \sigma(id_t)(''se'')\ .\ \sigma(id_t)(''scc'')\ .\ \mathtt{checktc}(\Delta(id_t), \sigma(id_t)) = \mathtt{true} \tag{1.177}$$

From (1.177), we can deduce:

$$\sigma(id_t)(''se'') = \mathtt{true} \tag{1.178}$$

$$\sigma(id_t)(''scc'') = \mathtt{true} \tag{1.179}$$

$$\mathtt{checktc}(\Delta(id_t), \sigma(id_t)) = \mathtt{true} \tag{1.180}$$

Term $\mathtt{checktc}(\Delta(id_t), \sigma(id_t))$ 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. $\boxed{t \notin T_i \vee s'.I(t) \in I_s(t)}$
3. $\boxed{\forall c \in \mathcal{C},\ \mathbb{C}(t,c) = 1 \Rightarrow s'.cond(c) = \mathtt{true} \text{ and } \mathbb{C}(t,c) = -1 \Rightarrow s'.cond(c) = \mathtt{false}}$

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 $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(''se'') = \mathtt{true} \Leftrightarrow t \in Sens(s.M)$.

   $t \in Sens(s.M)$.

2. $\boxed{\forall c \in \mathcal{C},\ \mathbb{C}(t,c) = 1 \Rightarrow s'.cond(c) = \mathtt{true} \text{ and } \mathbb{C}(t,c) = -1 \Rightarrow s'.cond(c) = \mathtt{false}}$

   Given a $c \in \mathcal{C}$, there are two points to prove:

   (a) $\boxed{\mathbb{C}(t,c) = 1 \Rightarrow s'.cond(c) = \mathtt{true}.}$

   (b) $\boxed{\mathbb{C}(t,c) = -1 \Rightarrow s'.cond(c) = \mathtt{false}.}$

   Let us prove these two points:

(a) Assuming that $\mathbb{C}(t,c) = 1$, let us show $\boxed{s'.cond(c) = \mathtt{true}.}$

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

$$\sigma(id_t)(\text{"}scc\text{"}) = \prod_{c \in conds(t)} \begin{cases} E_c(\tau,c) & if\ \mathbb{C}(t,c) = 1 \\ \mathtt{not}(E_c(\tau,c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} = \mathtt{true} \tag{1.181}$$

where $conds(t) = \{c \in \mathcal{C} \mid \mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1\}$.
As $c \in conds(t)$ and $\mathbb{C}(t,c) = 1$, and by definition of the product expression, we have:

$$E_c(\tau,c)\ .\prod_{c' \in conds(t) \setminus \{c\}} \begin{cases} E_c(\tau,c') & if\ \mathbb{C}(t,c') = 1 \\ \mathtt{not}(E_c(\tau,c')) & if\ \mathbb{C}(t,c') = -1 \end{cases} = \mathtt{true} \tag{1.182}$$

From (1.182), we can deduce that $E_c(\tau,c) = \mathtt{true}$.

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

Rewriting the goal with $s'.cond(c) = E_c(\tau,c)$ and $E_c(\tau,c) = \mathtt{true}$: tautology.

(b) Assuming that $\mathbb{C}(t,c) = -1$, let us show $\boxed{s'.cond(c) = \mathtt{false}.}$

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

$$\sigma(id_t)(\text{"}scc\text{"}) = \prod_{c \in conds(t)} \begin{cases} E_c(\tau,c) & if\ \mathbb{C}(t,c) = 1 \\ \mathtt{not}(E_c(\tau,c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} = \mathtt{true} \tag{1.183}$$

where $conds(t) = \{c \in \mathcal{C} \mid \mathbb{C}(t,c) = 1 \vee \mathbb{C}(t,c) = -1\}$.
As $c \in conds(t)$ and $\mathbb{C}(t,c) = -1$, and by definition of the product expression, we have:

$$\mathtt{not}\ E_c(\tau,c)\ .\prod_{c' \in conds(t) \setminus \{c\}} \begin{cases} E_c(\tau,c') & if\ \mathbb{C}(t,c') = 1 \\ \mathtt{not}(E_c(\tau,c')) & if\ \mathbb{C}(t,c') = -1 \end{cases} = \mathtt{true} \tag{1.184}$$

From (1.184), we can deduce that $E_c(\tau,c) = \mathtt{false}$.

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

Rewriting the goal with $s'.cond(c) = E_c(\tau,c)$ and $E_c(\tau,c) = \mathtt{false}$: tautology.

3. $\boxed{t \notin T_i \vee s'.I(t) \in I_s(t)}$

Reasoning on $\mathtt{checktc}(\Delta(id_t), \sigma(id_t)) = \mathtt{true}$, there are 3 cases:

(a) $\left(\mathtt{not}\ \sigma(id_t)(\text{"}srtc\text{"})\ .\ [\dots]\right) = \mathtt{true}$[2]

(b) $\left(\sigma(id_t)(\text{"}srtc\text{"})\ .\ \Delta(id_t)(\text{"}tt\text{"}) \neq \mathtt{NOT\_TEMP}\ .\ \sigma(id_t)(\text{"}A\text{"}) = 1\right) = \mathtt{true}$

(c) $\left(\Delta(id_t)(\text{"}tt\text{"}) = \mathtt{NOT\_TEMP}\right) = \mathtt{true}$

(a) $\left(\mathtt{not}\ \sigma(id_t)(\text{"}srtc\text{"})\ .\ [\dots]\right) = \mathtt{true}$:

[2] See equation (1.169) for the full definition

Then, we can deduce not $\sigma(id_t)(''srtc'') = \texttt{true}$ and $[\dots] = \texttt{true}$. From not $\sigma(id_t)(''srtc'') = \texttt{true}$, we can deduce $\sigma(id_t)(''srtc'') = \texttt{false}$, and from $[\dots] = \texttt{true}$, we have three other cases:

i. $(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B} \,.\, (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1) \,.\, (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1)) = \texttt{true}$

ii. $(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_A} \,.\, (\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1)) = \texttt{true}$

iii. $(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_INF} \,.\, (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1)) = \texttt{true}$

Let us prove the goal is these three contexts:

i. $(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B} \,.\, (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1) \,.\, (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1)) = \texttt{true}$:
   Then, converting boolean equalities into intuitionistic predicates, we have:
   - $\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B}$
   - $\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1$
   - $\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1$

   By property of the elaboration relation, and $\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B}$, 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 $\sigma(id_t)(''A'') = a$ and $\sigma(id_t)(''B'') = b$.
   Rewriting the goal with $\sigma(id_t)(''A'') = a$ and $\sigma(id_t)(''B'') = b$, and by definition of $\in$: $\boxed{\sigma(id_t)(''A'') \leq s'.I(t) \leq \sigma(id_t)(''B'').}$
   Now, let us perform case analysis on $s.I(t) \leq upper(I_s(t))$ or $s.I(t) > upper(I_s(t))$:
   - $s.I(t) \leq upper(I_s(t))$:

     By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $s.I(t) = \sigma(id_t)(''stc'')$.
     From $\sigma(id_t)(''se'') = \texttt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)(''srtc'') = \texttt{false}$, we can deduce $s.reset_t(t) = \texttt{false}$. Then, by definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = s.I(t) + 1$.
     $\Rightarrow \boxed{\sigma(id_t)(''A'') \leq s.I(t) + 1 \leq \sigma(id_t)(''B'')}$ (by $s'.I(t) = s.I(t) + 1$)
     $\Rightarrow \boxed{\sigma(id_t)(''A'') \leq \sigma(id_t)(''stc'') + 1 \leq \sigma(id_t)(''B'')}$ (by $s.I(t) = \sigma(id_t)(''stc'')$)
     $\Rightarrow \sigma(id_t)(''A'') - 1 \leq \sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1$
   - $s.I(t) > upper(I_s(t))$:

     By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(''stc'') = upper(I_s(t)) = b$.
     Then, from $\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1$, $\sigma(id_t)(''stc'') = upper(I_s(t)) = b$ and $\sigma(id_t)(''B'') = b$, we can deduce the following contradiction:
     $\sigma(id_t)(''B'') \leq \sigma(id_t)(''B'') - 1.$

ii. $(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_A} \,.\, (\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1)) = \texttt{true}$:
   Then, converting boolean equalities into intuitionistic predicates, we have:
   - $\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_A}$
   - $\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1$

   By property of the elaboration relation, and $\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_A}$, 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 ⇒a>, and by property of stable σ, we have $\sigma(id_t)("A") = a$.
Rewriting the goal with $\sigma(id_t)("A") = a$, unfolding the definition of $\in$, and simplifying the goal: $\boxed{s'.I(t) = \sigma(id_t)("A").}$
Now, let us perform case analysis on $s.I(t) \leq upper(I_s(t))$ or $s.I(t) > upper(I_s(t))$:

- $s.I(t) \leq upper(I_s(t))$:

  By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $s.I(t) = \sigma(id_t)("stc")$.
  From $\sigma(id_t)("se") = \texttt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)("srtc") = \texttt{false}$, we can deduce $s.reset_t(t) = \texttt{false}$. Then, by definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = s.I(t) + 1$.
  $\Rightarrow \boxed{s.I(t) + 1 = \sigma(id_t)("A")}$ (by $s'.I(t) = s.I(t) + 1$)
  $\Rightarrow \boxed{\sigma(id_t)("stc") + 1 = \sigma(id_t)("A")}$ (by $s.I(t) = \sigma(id_t)("stc")$)
  $\Rightarrow \sigma(id_t)("stc") = \sigma(id_t)("A") - 1$

- $s.I(t) > upper(I_s(t))$:

  By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)("stc") = upper(I_s(t)) = a$.
  Then, from $\sigma(id_t)("stc") = \sigma(id_t)("A") - 1$, $\sigma(id_t)("stc") = upper(I_s(t)) = a$, $\sigma(id_t)("A") = a$, and $a \in \mathbb{N}^*$, we can deduce the following contradiction:
  $\sigma(id_t)("A") = \sigma(id_t)("A") - 1$.

iii. $(\Delta(id_t)("tt") = \texttt{TEMP\_A\_INF}\ .\ (\sigma(id_t)("stc") \geq \sigma(id_t)("A") - 1)) = \texttt{true}$:
Then, converting boolean equalities into intuitionistic predicates, we have:

- $\Delta(id_t)("tt") = \texttt{TEMP\_A\_INF}$
- $\sigma(id_t)("stc") \geq \sigma(id_t)("A") - 1$

By property of the elaboration relation, and $\Delta(id_t)("tt") = \texttt{TEMP\_A\_INF}$, 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, <time_A_value ⇒a>, and by property of stable σ, we have $\sigma(id_t)("A") = a$.
Rewriting the goal with $\sigma(id_t)("A") = a$, unfolding the definition of $\in$, and simplifying the goal: $\boxed{\sigma(id_t)("A") \leq s'.I(t).}$
Now, let us perform case analysis on $s.I(t) \leq lower(I_s(t))$ or $s.I(t) > lower(I_s(t))$:

- $s.I(t) \leq lower(I_s(t))$:

  By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $s.I(t) = \sigma(id_t)("stc")$.
  From $\sigma(id_t)("se") = \texttt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)("srtc") = \texttt{false}$, we can deduce $s.reset_t(t) = \texttt{false}$. Then, by definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = s.I(t) + 1$.
  $\Rightarrow \boxed{\sigma(id_t)("A") \leq s.I(t) + 1}$ (by $s'.I(t) = s.I(t) + 1$)
  $\Rightarrow \boxed{\sigma(id_t)("A") \leq \sigma(id_t)("stc") + 1}$ (by $s.I(t) = \sigma(id_t)("stc")$)
  $\Rightarrow \sigma(id_t)("A") - 1 \leq \sigma(id_t)("stc")$

- $s.I(t) > lower(I_s(t))$:

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(\text{"}stc\text{"}) = lower(I_s(t)) = a$.
From $\sigma(id_t)(\text{"}se\text{"}) = \texttt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)(\text{"}srtc\text{"}) = \texttt{false}$, we can deduce $s.reset_t(t) = \texttt{false}$. Then, by definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = s.I(t) + 1$.

$\Rightarrow \boxed{\sigma(id_t)(\text{"}A\text{"}) \leq s.I(t) + 1}$ (by $s'.I(t) = s.I(t) + 1$)

$\Rightarrow \boxed{a \leq s.I(t) + 1}$ (by $\sigma(id_t)(\text{"}A\text{"}) = a$)

$\Rightarrow \boxed{a < s.I(t)}$

$\Rightarrow lower(I_s(t)) < s.I(t)$

(b) $\big(\sigma(id_t)(\text{"}srtc\text{"}) \;.\; \Delta(id_t)(\text{"}tt\text{"}) \neq \texttt{NOT\_TEMP} \;.\; \sigma(id_t)(\text{"}A\text{"}) = 1\big) = \texttt{true}$

Then, converting boolean equalities into intuitionistic predicates, we have:

- $\sigma(id_t)(\text{"}srtc\text{"}) = \texttt{true}$
- $\Delta(id_t)(\text{"}tt\text{"}) \neq \texttt{NOT\_TEMP}$
- $\sigma(id_t)(\text{"}A\text{"}) = 1$

By property of the elaboration relation, and $\Delta(id_t)(\text{"}tt\text{"}) \neq \texttt{NOT\_TEMP}$, 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, $\texttt{<time\_A\_value} \Rightarrow a\texttt{>} \in ipm_t$, and by property of stable σ, we have $\sigma(id_t)(\text{"}A\text{"}) = a$. Thus, we can deduce $a = 1$ and $I_s(t) = [1, ni]$.

By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, from $\sigma(id_t)(\text{"}se\text{"}) = \texttt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)(\text{"}srtc\text{"}) = \texttt{true}$, we can deduce $s.reset_t(t) = \texttt{true}$.

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, $t \in Sens(s.M)$ 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) $\big(\Delta(id_t)(\text{"}tt\text{"}) = \texttt{NOT\_TEMP}\big) = \texttt{true}$

Let us show $\boxed{t \notin T_i.}$

By property of the elaboration relation and $\Delta(id_t)(\text{"}tt\text{"}) = \texttt{NOT\_TEMP}$, we have $t \notin T_i$.

□

Lemma 38 (Falling Edge Equal Not Firable). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then* $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \in Firable(s') \Leftrightarrow \sigma'(id_t)(\text{"}s\_firable\text{"}) = \texttt{true}$.

Proof. Proving the above lemma is trivial by appealing to Lemma Falling Edge Equal Firable and by reasoning on contrapositives. □

1.7 A detailled proof: equivalence of fired transitions

Lemma 39 (Falling Edge Equal Fired). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then* $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $t \in Fired(s') \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = \texttt{true}$.

Proof. Given a $t \in T$ and an id_t s.t. $\gamma(t) = id_t$, let us show $\boxed{t \in Fired(s') \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = \texttt{true}.}$ The proof is in two parts:

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

   By definition of $t \in Fired(s')$, there exists $fset \subseteq T$ s.t. $IsFiredSet(s', fset) \ \wedge \ t \in fset$.

   Let us take such an $fset$, and apply Lemma Falling Edge Equal Fired Set to solve the goal.

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

   By definition of $t \in Fired(s')$, let us show that $\boxed{\exists fset \subseteq T \text{ s.t. } IsFiredSet(s', fset) \ \wedge \ t \in fset}$

   well-
   PN

   Assuming that $sitpn$ is a well-defined $SITPN$ (see Section), we can always find an $fset \subseteq T$ such that $\forall s \in S(sitpn)$, $IsFiredSet(s, fset)$ is derivable. Let us take an $fset \subseteq T$ s.t. $IsFiredSet(s', fset)$, and use it to prove the goal by applying Lemma Falling Edge Equal Fired Set.

□

Lemma 40 (Falling Edge Equal Not Fired). *For all $sitpn$, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\forall t, id_t$ s.t. $\gamma(t) = id_t$, $t \notin Fired(s') \Leftrightarrow \sigma_t'(\text{"}fired\text{"}) = \texttt{false}$.*

Proof. Proving the above lemma is trivial by appealing to Lemma Falling Edge Equal Fired and by reasoning on contrapositives. □

Lemma 41 (Falling Edge Equal Fired Set). *For all $sitpn$, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\forall t \in T$, $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $\forall fset \subseteq T$, s.t. $IsFiredSet(s', fset)$, $t \in fset \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = true$.*

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 $\boxed{t \in fset \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = true.}$

By definition of $IsFiredSet(s', fset)$, we have $IsFiredSetAux(s', \emptyset, T, fset)$.
Then, we can appeal to Lemma Falling Edge Equal Fired Set Aux to solve the goal, but first we must prove the following *extra hypothesis* (i.e, one of the premise of Lemma Falling Edge Equal Fired Set Aux):

$$\boxed{\begin{array}{l}\forall t' \in T, id_{t'} \in Comps(\Delta) \text{ s.t. } \gamma(t') = id_{t'},\\ (t' \in \emptyset \Rightarrow \sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true} \Rightarrow t' \in \emptyset \ \vee \ t' \in T).\end{array}}$$

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. $\boxed{t' \in \emptyset \Rightarrow \sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true}}$

2. $\boxed{\sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true} \Rightarrow t' \in \emptyset \ \vee \ t' \in T}$

Let us show these two points:

1. Assuming $t' \in \emptyset$, let us show $\boxed{\sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true}.}$

   $t' \in \emptyset$ is a contradiction.

2. Assuming $\sigma'(id_{t'})(''fired'') = \texttt{true}$, let us show $\boxed{t' \in \emptyset \lor t' \in T.}$

   By definition, $t' \in T$.

$\square$

Lemma 42 (Falling Edge Equal Fired Set Aux). *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $\forall fired \subseteq T$, $T_s \subseteq T$, $fset \subseteq T$, assume that:*

- $IsFiredSetAux(s', fired, T_s, fset)$
- *EH (Extra. Hypothesis):*
  $\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
  $(t' \in fired \Rightarrow \sigma'(id_{t'})(''fired'') = \texttt{true}) \land (\sigma'(id_{t'})(''fired'') = \texttt{true} \Rightarrow t' \in fired \lor t' \in T_s)$.

then $t \in fset \Leftrightarrow \sigma'(id_t)(''fired'') = \texttt{true}$.

Proof. Given a $t \in T$, an $id_t \in Comps(\Delta)$, a $fired, T_s, fset \subseteq T$, and assuming $IsFiredSetAux(s', fired, T_s, fset)$ and EH, let us show $\boxed{t \in fset \Leftrightarrow \sigma'(id_t)(''fired'') = \texttt{true}.}$
Let us reason by induction on $IsFiredSetAux(s', fired, T_s, fset)$.

- **BASE CASE**: $\boxed{t \in fired \Leftrightarrow \sigma'(id_t)(''fired'') = \texttt{true}.}$

  In that case, $fired = fset$ and $T_s = \emptyset$, EH looks like this:
  $\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
  $(t' \in fired \Rightarrow \sigma'(id_{t'})(''fired'') = \texttt{true}) \land (\sigma'(id_{t'})(''fired'') = \texttt{true} \Rightarrow t' \in fired \lor t' \in \emptyset)$.

  From EH, we can deduce $t \in fired \Leftrightarrow \sigma'(id_t)(''fired'') = \texttt{true}$.

- **INDUCTION CASE**: $\boxed{t \in fset \Leftrightarrow \sigma'(id_t)(''fired'') = \texttt{true}.}$

  In that case, we have:

  – $IsTopPrioritySet(T_s, tp)$
  – $ElectFired(s', fired, tp, fired')$
  – $FiredAux(s', fired', T_s \setminus tp, fset)$

  $(\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
  $(t' \in fired' \Rightarrow \sigma'(id_{t'})(''fired'') = \texttt{true}) \land(\sigma'(id_{t'})(''fired'') = \texttt{true} \Rightarrow t' \in fired' \lor t' \in T_s \setminus tp)) \Rightarrow$
  $t \in fset \Leftrightarrow \sigma'_t(''fired'') = true.$

  Applying the induction hypothesis, then, the new goal is:

  $\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
  $(t' \in fired' \Rightarrow \sigma'(id_{t'})(''fired'') = \texttt{true})$
  $\land(\sigma'(id_{t'})(''fired'') = \texttt{true} \Rightarrow t' \in fired' \lor t' \in T_s \setminus tp)$

Apply Lemma Elect Fired Equal Fired to solve the goal.

□

Lemma 43 (Elect Fired Equal Fired)**.** *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_{\downarrow}$, σ' that verify the hypotheses of Def. 11, then $\forall fired, fired', T_s, tp,\ fset \subseteq T$, assume that:*

- $IsTopPrioritySet(T_s, tp)$
- $ElectFired(s', fired, tp, fired')$
- $FiredAux(s', fired', T_s \setminus tp, fset)$
- *EH (Extra. Hypothesis):*
  $\forall t' \in T, id_{t'} \in Comps(\Delta)$ *s.t.* $\gamma(t') = id_{t'}$,
  $(t' \in fired \Rightarrow \sigma'(id_{t'})(''fired'') = \mathtt{true}) \wedge (\sigma'(id_{t'})(''fired'') = \mathtt{true} \Rightarrow t' \in fired\ \vee\ t' \in T_s)$

then $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$,
$(t \in fired' \Rightarrow \sigma'(id_t)(''fired'') = \mathtt{true}) \wedge (\sigma'(id_t)(''fired'') = \mathtt{true} \Rightarrow t \in fired' \vee t \in T_s \setminus tp)$.

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

$$\boxed{(t \in fired' \Rightarrow \sigma'(id_t)(''fired'') = \mathtt{true}) \wedge (\sigma'(id_t)(''fired'') = \mathtt{true} \Rightarrow t \in fired' \vee t \in T_s \setminus tp).}$$

Let us reason by induction on $ElectFired(s', fired, tp, fired')$; there are three cases:

1. **BASE CASE**: $tp = \emptyset$ and $fired = fired'$.
2. **INDUCTIVE CASE**: $tp = \{t_0\} \cup tp_0$ and t_0 is elected to be fired.
3. **INDUCTIVE CASE**: $tp = \{t_0\} \cup tp_0$ and t_0 is not elected to be fired.

Let us prove the goal in these three contexts:

1. **BASE CASE**:

   $$\boxed{(t \in fired \Rightarrow \sigma'(id_t)(''fired'') = \mathtt{true}) \wedge (\sigma'(id_t)(''fired'') = \mathtt{true} \Rightarrow t \in fired \vee t \in T_s).}$$

   Apply EH to solve the goal.

2. **INDUCTIVE CASE**: $tp = \{t_0\} \cup tp_0$ and t_0 is elected to be fired.

   In that case, we have:

   - $IsTopPrioritySet(T_s, \{t_0\} \cup tp_0)$
   - $ElectFired(s', fired \cup \{t_0\}, tp_0, fired')$
   - $IsFiredSetAux(s', fired', T_s \setminus \{t_0\} \cup tp_0, fset)$
   - $t_0 \in Firable(s')$
   - $t_0 \in Sens(s'.M - \sum_{t_i \in Pr(t, fired)} pre(t_i))$ where $Pr(t, fired) = \{t' \mid t' \succ t \wedge t' \in fired\}$
   - EH: $\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
     $(t' \in fired \Rightarrow \sigma'(id_{t'})(''f'') = \mathtt{true}) \wedge (\sigma'(id_{t'})(''f'') = \mathtt{true} \Rightarrow t' \in fired\ \vee\ t' \in T_s)$

$\forall T'_s \subseteq T,$
$IsTopPrioritySet(T'_s, tp_0) \Rightarrow$
$IsFiredSetAux(s', fired', T'_s \setminus tp_0, fset) \Rightarrow$
$(\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'},$
$(t' \in fired \cup \{t_0\} \Rightarrow \sigma'_{t'}("f") = \texttt{true}) \wedge (\sigma'(id_{t'})("f") = \texttt{true} \Rightarrow t' \in fired \cup \{t_0\} \ \vee \ t' \in T'_s)) \Rightarrow$
$\forall t \in T,\ id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t,$
$\left(t \in fired' \Rightarrow \sigma'(id_t)("f") = \texttt{true}\right) \wedge \left(\sigma'(id_t)("f") = \texttt{true} \Rightarrow t \in fired' \vee t \in T'_s \setminus tp_0\right)$

$\forall t \in T,\ id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t,$
$\left(t \in fired' \Rightarrow \sigma'_t("f") = \texttt{true}\right) \wedge \left(\sigma'_t("f") = \texttt{true} \Rightarrow t \in fired' \vee t \in T_s \setminus \{t_0\} \cup tp_0\right)$

To solve the goal, we can apply the induction hypothesis with $T'_s = T_s \setminus \{t_0\}$; then, there are three points to prove:

(a) $\boxed{IsTopPrioritySet(T_s \setminus \{t_0\}, tp_0)}$

(b) $\boxed{IsFiredSetAux(s', fired', (T_s \setminus \{t_0\}) \setminus tp_0, fset)}$

(c) $\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'},$
$(t' \in fired \cup \{t_0\} \Rightarrow \sigma'_{t'}("f") = \texttt{true}) \wedge (\sigma'(id_{t'})("f") = \texttt{true} \Rightarrow t' \in fired \cup \{t_0\} \ \vee \ t' \in T_s \setminus \{t_0\})$

Let us prove these three points:

(a) $\boxed{IsTopPrioritySet(T_s \setminus \{t_0\}, tp_0)}$

Not provable yet.

(b) $\boxed{IsFiredSetAux(s', fired', (T_s \setminus \{t_0\}) \setminus tp_0, fset)}$.
We know that $(T_s \setminus \{t_0\}) \setminus tp_0 = T_s \setminus (\{t_0\} \cup tp_0)$, and thus
$IsFiredSetAux(s', fired', T_s \setminus (\{t_0\} \cup tp_0), fset)$ is an assumption.

(c) $\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'},$
$(t' \in fired \cup \{t_0\} \Rightarrow \sigma'(id_{t'})("f") = \texttt{true}) \wedge (\sigma'(id_{t'})("f") = \texttt{true} \Rightarrow t' \in fired \cup \{t_0\} \ \vee \ t' \in T_s \setminus \{t_0\})$

Given a $t' \in T$ and an $id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$, let us show
$(t' \in fired \cup \{t_0\} \Rightarrow \sigma'(id_{t'})("f") = \texttt{true})$
$\wedge (\sigma'(id_{t'})("f") = \texttt{true} \Rightarrow t' \in fired \cup \{t_0\} \ \vee \ t' \in T_s \setminus \{t_0\})$.
The proof is in two parts.

i. Assuming that $t' \in fired \cup \{t_0\}$, let us show $\boxed{\sigma'(id_{t'})("f") = \texttt{true}.}$
Case analysis on $t' \in fired \cup \{t_0\}$; there are two cases:
- $t' \in fired$
- $t' = t_0$

Let us prove the goal in these two contexts.

- **CASE** $t' \in fired$: Thanks to EH, we can deduce $\sigma'_{t'}("f") = \texttt{true}$.
- **CASE** $t' = t_0$:
  By definition of $id_{t'}$, there exist a $gm_{t'}$, $ipm_{t'}$, $opm_{t'}$ s.t. $\texttt{comp}(id_{t'}, "transition", gm_{t'}, ipm_{t'}, opm_{t'}) \in d.cs$.
  By property of the stabilize relation and $\texttt{comp}(id_{t'}, "transition", gm_{t'}, ipm_{t'}, opm_{t'}) \in d.cs$:
  $$\sigma(id_{t'})("f") = \sigma(id_{t'})("sfa") \,.\, \sigma(id_{t'})("spc") \tag{1.185}$$
  Rewriting the goal with (1.185): $\boxed{\sigma(id_{t'})("sfa") \,.\, \sigma(id_{t'})("spc") = \texttt{true}.}$
  Then, we can show that:
  - $\sigma(id_{t'})("sfa") = \texttt{true}$ by applying Lemma Falling Edge Equal Firable
  - $\sigma(id_{t'})("spc") = \texttt{true}$ by applying Lemma Stabilize Compute Priority Combination After Falling Edge.

ii. Assuming that $\sigma'(id_{t'})("f") = \texttt{true}$, let us show $\boxed{t' \in fired \cup \{t_0\} \ \vee \ t' \in T_s \setminus \{t_0\}.}$
From $\sigma'(id_{t'})("f") = \texttt{true}$ and EH, we can deduce that $t' \in fired \vee t' \in T_s$.
Case analysis on $t' \in fired \vee t' \in T_s$.

- **CASE** $t' \in fired$: then, it is trivial to show $t' \in fired \cup \{t_0\}$.
- **CASE** $t' \in T_s$: We know that $t_0 \in T_s$. Therefore, either $t' \in T_s \setminus \{t_0\}$, or $t' = t_0$, and then, $t' \in fired \cup \{t_0\}$.

3. **INDUCTIVE CASE**: $tp = \{t_0\} \cup tp_0$ and t_0 is not elected to be fired.

- $IsTopPrioritySet(T_s, \{t_0\} \cup tp_0)$
- $ElectFired(s', fired, tp_0, fired')$
- $IsFiredSetAux(s', fired', T_s \setminus \{t_0\} \cup tp_0, fset)$
- $\neg(t_0 \in Firable(s') \wedge t_0 \in Sens(s'.M - \sum_{t_i \in Pr(t, fired)} pre(t_i)))$
- EH:
  $\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
  $(t' \in fired \Rightarrow \sigma'(id_{t'})("f") = \texttt{true}) \wedge (\sigma'(id_{t'})("f") = \texttt{true} \Rightarrow t' \in fired \ \vee \ t' \in T_s)$

$\forall T'_s \subseteq T$,
$IsTopPrioritySet(T'_s, tp_0) \Rightarrow$
$IsFiredSetAux(s', fired', T'_s \setminus tp_0, fset) \Rightarrow$
$(\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in fired \Rightarrow \sigma'(id_{t'})("f") = \texttt{true}) \wedge (\sigma'(id_{t'})("f") = \texttt{true} \Rightarrow t' \in fired \ \vee \ t' \in T'_s)) \Rightarrow$
$\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,
$(t \in fired' \Rightarrow \sigma'(id_t)("f") = \texttt{true}) \wedge (\sigma'(id_t)("f") = \texttt{true} \Rightarrow t \in fired' \vee t \in T'_s \setminus tp_0)$

$\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,
$(t \in fired' \Rightarrow \sigma'(id_t)("f") = \texttt{true}) \wedge (\sigma'(id_t)("f") = \texttt{true} \Rightarrow t \in fired' \vee t \in T_s \setminus \{t_0\} \cup tp_0)$.

Then, we can apply the induction hypothesis with $T_s' = T_s \setminus \{t_0\}$, then, there are three points to prove:

(a) $\boxed{IsTopPrioritySet(T_s \setminus \{t_0\}, tp_0)}$

(b) $\boxed{IsFiredSetAux(s', fired', (T_s \setminus \{t_0\}) \setminus tp_0, fset)}$

(c) $\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in fired \Rightarrow \sigma'(id_{t'})(''f'') = \texttt{true}) \wedge (\sigma'(id_{t'})(''f'') = \texttt{true} \Rightarrow t' \in fired \vee t' \in T_s \setminus \{t_0\})$

Let us prove these three points:

(a) $\boxed{IsTopPrioritySet(T_s \setminus \{t_0\}, tp_0)}$

Not provable yet.

(b) $\boxed{IsFiredSetAux(s', fired', (T_s \setminus \{t_0\}) \setminus tp_0, fset)}$
We know that $(T_s \setminus \{t_0\}) \setminus tp_0 = T_s \setminus (\{t_0\} \cup tp_0)$, and thus
$IsFiredSetAux(s', fired', T_s \setminus (\{t_0\} \cup tp_0), fset)$ is an assumption.

(c) $\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
$(t' \in fired \Rightarrow \sigma'(id_{t'})(''f'') = \texttt{true}) \wedge (\sigma'(id_{t'})(''f'') = \texttt{true} \Rightarrow t' \in fired \vee t' \in T_s \setminus \{t_0\})$
Given a $t' \in T$ and an $id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$, let us show

$$\boxed{(t' \in fired \Rightarrow \sigma'(id_{t'})(''f'') = \texttt{true}) \wedge (\sigma'(id_{t'})(''f'') = \texttt{true} \Rightarrow t' \in fired \vee t' \in T_s \setminus \{t_0\})}$$

The proof is in two parts:

i. Assuming that $t' \in fired$, let us show $\boxed{\sigma'(id_{t'})(''f'') = \texttt{true}}$.
From $t' \in fired$ and EH, $\sigma'(id_{t'})(''f'') = \texttt{true}$.

ii. Assuming that $\sigma'(id_{t'})(''f'') = \texttt{true}$, let us show $\boxed{t' \in fired \vee t' \in T_s \setminus \{t_0\}}$.
Thanks to $\sigma'(id_{t'})(''f'') = \texttt{true}$ and EH, we know that: $t' \in fired \vee t' \in T_s$.
Case analysis on $t' \in fired \vee t' \in T_s$; there are two cases:

- **CASE** $t' \in fired$.
- **CASE** $t' \in T_s$:
  From $IsTopPrioritySet(T_s, \{t_0\} \cup tp_0)$, we can deduce that $t_0 \in T_s$. Therefore, either $t' \in T_s \setminus \{t_0\}$ or $t' = t_0$.
  In the case where $t' = t_0$, we need to show a contradiction by proving
  $t' \in Firable(s')$ and $t' \in Sens(s'.M - \sum_{t_i \in Pr(t, fired)} pre(t_i))$ based on $\sigma'(id_{t'})(''f'') = \texttt{true}$.
  By definition of $id_{t'}$, there exist a $gm_{t'}$, $ipm_{t'}$, $opm_{t'}$ s.t. $\texttt{comp}(id_{t'}, ''transition'', gm_{t'}, ipm_{t'}, opm_{t'}) \in d.cs$.

By property of the stabilize relation and $\texttt{comp}(id_{t'}, \text{"}transition\text{"}, gm_{t'}, ipm_{t'}, opm_{t'}) \in d.cs$:

$$\sigma(id_{t'})(\text{"}f\text{"}) = \sigma(id_{t'})(\text{"}sfa\text{"}) \,.\, \sigma(id_{t'})(\text{"}spc\text{"}) = \texttt{true} \tag{1.186}$$

From $\sigma(id_{t'})(\text{"}sfa\text{"}) = \texttt{true}$, and appealing to Lemma Falling Edge Equal Firable, we can deduce $t' \in Firable(s')$.
From $\sigma(id_{t'})(\text{"}spc\text{"}) = \texttt{true}$, and appealing to Lemma Stabilize Compute Priority Combination After Falling Edge, we can deduce $t' \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i))$.

Then, as $t' = t_0$, $\neg(t_0 \in Firable(s') \wedge t_0 \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i)))$ is a contradiction.

□

Lemma 44 (Stabilize Compute Priority Combination After Falling Edge)**.** *For all sitpn, d, γ, Δ, σ_e, E_c, E_p, τ, s, s', σ, σ_i, $\sigma_\downarrow$, σ' that verify the hypotheses of Def. 11, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,*
$\forall fired, fired', T_s, tp, fset \subseteq T$ assume that:

- $IsTopPrioritySet(T_s, \{t\} \cup tp)$
- $ElectFired(s', fired, tp, fired')$
- $FiredAux(s', fired', T_s \setminus \{t\} \cup tp, fset)$
- *EH:* $\forall t' \in T, id_{t'} \in Comps(\Delta)$ *s.t.* $\gamma(t') = id_{t'}$,
  $(t' \in fired \Rightarrow \sigma'(id_{t'})(\text{"}f\text{"}) = \texttt{true}) \wedge (\sigma'(id_{t'})(\text{"}f\text{"}) = \texttt{true} \Rightarrow t' \in fired \vee t' \in T_s)$.
- $t \in Firable(s')$

then $t \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i)) \Leftrightarrow \sigma'(id_t)(\text{"}spc\text{"}) = \texttt{true}$

Proof. Given a $t \in T$ and an $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, a $fired$, $fired'$, T_s, tp, $fset \subseteq T$ and assuming all the above hypotheses, let us show

$$\boxed{t \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i)) \Leftrightarrow \sigma'(id_t)(\text{"}spc\text{"}) = \texttt{true}.}$$

By definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$.
By property of the stabilize relation and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_t)(\text{"}spc\text{"}) = \prod_{i=0}^{\Delta(id_t)(\text{"}ian\text{"})-1} \sigma'(id_t)(\text{"}pauths\text{"})[i] \tag{1.187}$$

Rewriting the goal with (1.187):

$$\boxed{t \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i)) \Leftrightarrow \prod_{i=0}^{\Delta(id_t)(\text{"}ian\text{"})-1} \sigma'(id_t)(\text{"}pauths\text{"})[i] = \texttt{true}.}$$

Then, the proof is in two parts:

1. $t \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i)) \Rightarrow \prod_{i=0}^{\Delta(id_t)(\text{"}ian\text{"})-1} \sigma'(id_t)(\text{"}pauths\text{"})[i] = \texttt{true}$

2. $\prod_{i=0}^{\Delta(id_t)("ian")-1} \sigma'(id_t)("pauths")[i] = \mathtt{true} \Rightarrow t \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i))$

Let us prove both sides of the equivalence:

1. Assuming that $t \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i))$, let us show $\boxed{\prod_{i=0}^{\Delta(id_t)("ian")-1} \sigma'(id_t)("pauths")[i] = \mathtt{true}.}$

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

   - **CASE** $input(t) = \emptyset$:
     By construction, $<$`input_arcs_number` $\Rightarrow$1$>$ $\in gm_t$ and $<$`priority_authorizations(0)` $\Rightarrow$ `true`$>$ $\in ipm_t$.
     By property of the elaboration relation, we have $\Delta(id_t)("ian") = 1$, and by property of the stabilize relation, we have $\sigma'(id_t)("pauths")[0] = \mathtt{true}$.
     Rewriting the goal with $\Delta(id_t)("ian") = 1$ and $\sigma'(id_t)("pauths")[0] = \mathtt{true}$, and simplifying the goal: tautology.

   - **CASE** $input(t) \neq \emptyset$:
     Then, let us show an equivalent goal:
     $\boxed{\forall i \in [0, \Delta(id_t)("ian") - 1],\ \sigma'(id_t)("pauths")[i] = \mathtt{true}.}$
     Given an $i \in [0, \Delta(id_t)("ian") - 1]$, let us show $\boxed{\sigma'(id_t)("pauths")[i] = \mathtt{true}.}$
     By construction, $<$`input_arcs_number` $\Rightarrow |input(t)|>$ $\in gm_t$.
     By property of the elaboration relation, we have $\Delta(id_t)("ian") = |input(t)|$. 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 gm_p, ipm_p, opm_p s.t. $\mathtt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$, and there exist a $j \in [0, |output(p)|]$ and an $id_{ji} \in Sigs(\Delta)$ s.t.
     $<$`input_arcs_valid(i)` $\Rightarrow$ `id`$_{\mathtt{ji}}$$>$ $\in ipm_t$ and $<$`output_arcs_valid(j)` $\Rightarrow$ `id`$_{\mathtt{ji}}$$>$ $\in opm_t$.
     Let us take such a $p \in input(t)$, $id_p \in Comps(\Delta)$, gm_p, ipm_p, opm_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** $pre(p,t) = (\omega, \mathtt{test})$ or $pre(p,t) = (\omega, \mathtt{inhib})$:
       By construction, $<$`priority_authorizations(i)` $\Rightarrow$ `true`$>$ $\in ipm_t$, and by property of the stabilize relation: $\sigma'(id_t)("pauths")[i] = \mathtt{true}$.

     - **CASE** $pre(p,t) = (\omega, \mathtt{basic})$:
       Let us define $output_c(p) = \{t \in T \mid \exists \omega,\ pre(p,t) = (\omega, \mathtt{basic})\}$, 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, $<$`priority_authorizations(i)` $\Rightarrow$ `true`$>$ $\in ipm_t$, and by property of the stabilize relation: $\sigma'(id_t)("pauths")[i] = \mathtt{true}$.

* **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.
  $<$`priority_authorizations(i)` $\Rightarrow$ `id'ji`$>$ $\in ipm_t$ and
  $<$`priority_authorizations(j)` $\Rightarrow$ `id'ji`$>$ $\in opm_p$.
  By property of the stabilize relation, $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$ and $\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_t)(\text{"}pauths\text{"})[i] = \sigma'(id'_{ji}) = \sigma'(id_p)(\text{"}pauths\text{"})[j] \tag{1.188}$$

  Rewriting the goal with (1.188): $\boxed{\sigma'(id_p)(\text{"}pauths\text{"})[j] = \texttt{true}.}$
  By property of the stabilize relation and $\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_p)(\text{"}pauths\text{"})[j] = (\sigma'(id_p)(\text{"}sm\text{"}) \geq \texttt{rsum} + \sigma'(id_p)(\text{"}oaw\text{"})[j]) \tag{1.189}$$

  Let us define the `rsum` term as follows:

$$\texttt{rsum} = \sum_{i=0}^{j-1} \begin{cases} \sigma'(id_p)(\text{"}oaw\text{"})[i] \texttt{ if } \sigma'(id_p)(\text{"}otf\text{"})[i]. \\ \qquad\qquad \sigma'(id_p)(\text{"}oat\text{"})[i] = \texttt{basic} \\ 0 \; otherwise \end{cases} \tag{1.190}$$

  Rewriting the goal with (1.189): $\boxed{\sigma'(id_p)(\text{"}sm\text{"}) \geq \texttt{rsum} + \sigma'(id_p)(\text{"}oaw\text{"})[j]}$
  By definition of $t \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i))$, we have $s'.M(p) \geq \sum_{t_i \in Pr(t,fired)} pre(p, t_i) + \omega$.
  Then, there are three points to prove:
  (a) $\boxed{s'.M(p) = \sigma'(id_p)(\text{"}sm\text{"})}$
  (b) $\boxed{\omega = \sigma'(id_p)(\text{"}oaw\text{"})[j]}$
  (c) $\boxed{\sum_{t_i \in Pr(t,fired)} pre(p, t_i) = \texttt{rsum}}$
  Let us prove these three points:
  (a) $\boxed{s'.M(p) = \sigma'(id_p)(\text{"}sm\text{"})}$
  Appealing to Lemma Falling Edge Equal Marking: $s'.M(p) = \sigma'(id_p)(\text{"}sm\text{"})$.
  (b) $\boxed{\omega = \sigma'(id_p)(\text{"}oaw\text{"})[j]}$
  By construction, and as $pre(p, t) = (\omega, \texttt{basic})$, we have
  $<$`output_arcs_weights(j)` $\Rightarrow \omega>$ $\in ipm_p$.
  By property of the stabilize relation and $\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$:
  $\omega = \sigma'(id_p)(\text{"}oaw\text{"})[j]$.
  (c) $\boxed{\sum_{t_i \in Pr(t,fired)} pre(p, t_i) = \texttt{rsum}}$
  Let us replace the left and right term of the equality by their full definition:

$$\sum_{t_i \in Pr(t,fired)} \begin{cases} \omega \texttt{ if } pre(p,t_i) = (\omega, \texttt{basic}) \\ 0 \ otherwise \end{cases}$$
$$=$$
$$\sum_{i=0}^{j-1} \begin{cases} \sigma'(id_p)(''oaw'')[i] \texttt{ if } \sigma'(id_p)(''otf'')[i]. \\ \qquad\qquad\qquad \sigma'(id_p)(''oat'')[i] = \texttt{basic} \\ 0 \ otherwise \end{cases}$$

Let us define $f(t_i) = \begin{cases} \omega \texttt{ if } pre(p,t_i) = (\omega, \texttt{basic}) \\ 0 \ otherwise \end{cases}$ and
$g(i) = \begin{cases} \sigma'(id_p)(''oaw'')[i] \texttt{ if } \sigma'(id_p)(''otf'')[i]. \\ \qquad\qquad\qquad \sigma'(id_p)(''oat'')[i] = \texttt{basic} \\ 0 \ otherwise \end{cases}$

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

BASE CASE: then, we have $i > j - 1$, and then $j = 0$.

$$\sum_{t_i \in Pr(t,fired)} \begin{cases} \omega \texttt{ if } pre(p,t_i) = (\omega, \texttt{basic}) \\ 0 \ otherwise \end{cases} = 0$$

We know that the priority relation is a strict total order over the transitions of set $output_c(p)$. This ordering is reflected in the ordering of the indexes of output port `priority_authorizations` of place component instances. Thus, in the `priority_-authorizations` output port of a place component instance, the element of index 0 is connected to the transition of $output_c(t)$ with the highest firing priority. We know that component id_t is connected to `priority_authorizations(0)` in the output port map of component id_p. By construction, transition t is the transition of $output_c(p)$ with the highest firing priority, i.e, $\nexists t' \in output_c(p)$ s.t. $t' \succ t$.

The following part of the proof is the result of induction over term $\sum_{t_i \in Pr(t,fired)} f(t_i)$. Induction is not detailled here.

For all transition $t_i \in Pr(t, fired)$, either t_i is not in $output_c(p)$, and thus t_i has no effect in the value of the sum term $\sum_{t_i \in Pr(t,fired)} f(t_i)$; or, $t_i \in output_c(p)$. Then, by definition of $t_i \in Pr(t, fired)$, $t_i \succ t$, which is contradiction with $\nexists t' \in output_c(p)$ s.t. $t' \succ t$.

INDUCTIVE CASE: then, $0 \leq j - 1$, and thus $j > 0$.

For all $Pr' \subseteq T$, $g(0) + \sum\limits_{t_i \in Pr'} f(t_i) = g(0) + \sum\limits_{i=1}^{j-1} g(i)$

$$\sum_{t_i \in Pr(t,fired)} f(t_i) = g(0) + \sum_{i=1}^{j-1} g(i).$$

By definition of $g(0)$:

$$\sum_{t_i \in Pr(t,fired)} f(t_i) = \begin{cases} \sigma'(id_p)(''oaw'')[0] \texttt{ if } \sigma'(id_p)(''otf'')[0]. \\ \qquad\qquad\qquad\qquad \sigma'(id_p)(''oat'')[0] = \texttt{basic} \\ 0 \ otherwise \end{cases} + \sum_{i=1}^{j-1} g(i).$$

Case analysis on the value of $\sigma'(id_p)(''otf'')[0] \;.\; \sigma'(id_p)(''oat'')[0] = \texttt{basic}$:

In the case where $\left(\sigma'(id_p)(''otf'')[0] \;.\; \sigma'(id_p)(''oat'')[0] = \texttt{basic}\right) = \texttt{false}$, then $g(0) = 0$, and we can use the induction hypothesis with $Pr' = Pr(t, fired)$ to prove the goal.

In the case where $\left(\sigma'(id_p)(''otf'')[0] \;.\; \sigma'(id_p)(''oat'')[0] = \texttt{basic}\right) = \texttt{true}$, then $g(0) = \sigma'(id_p)(''oaw'')[0]$:

$$\sum_{t_i \in Pr(t,fired)} f(t_i) = \sigma'(id_p)(''oaw'')[0] + \sum_{i=1}^{j-1} g(i).$$

By construction, and knowing that $j > 0$ and that the priority relation is a strict total order over the set $output_c(p)$, there exist a $t_0 \in output_c(p)$ s.t. $t_0 \succ t$. Moreover, there exist an $id_{t_0} \in Comps(\Delta)$ s.t. $\gamma(t_0) = id_{t_0}$, and by definition of id_{t_0}, there exist gm_{t_0}, ipm_{t_0} and opm_{t_0} s.t. $\texttt{comp}(id_{t_0}, ''transition'', gm_{t_0}, ipm_{t_0}, opm_{t_0}) \in d.cs$. Finally, there exist an $id_{ft_0} \in Sigs(\Delta)$ s.t. $<\texttt{fired} \Rightarrow \texttt{id}_{\texttt{ft}_0}> \in opm_{t_0}$ and $<\texttt{output\_transitions\_fired(0)} \Rightarrow \texttt{id}_{\texttt{ft}_0}> \in ipm_p$.
By property of the stabilize relation, $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$ and $\texttt{comp}(id_{t_0}, ''transition'', gm_{t_0}, ipm_{t_0}, opm_{t_0}) \in d.cs$:

$$\sigma'(id_{t_0})(''f'') = \sigma'(id_{ft_0}) = \sigma'(id_p)(''otf'')[0] = \texttt{true} \qquad (1.191)$$

From EH and $\sigma'(id_{t_0})(''f'') = \texttt{true}$, we have either $t_0 \in fired$ or $t_0 \in T_s$.

❑ In the case where $t_0 \in fired$, then, by definition of $\sum$:

$$f(t_0) + \sum_{t_i \in Pr(t,fired) \setminus \{t_0\}} f(t_i) = \sigma'(id_p)(''oaw'')[0] + \sum_{i=1}^{j-1} g(i).$$

By definition of $t_0 \in output_c(p)$, there exists $\omega \in \mathbb{N}^*$ s.t. $pre(p, t_0) = (\omega, \texttt{basic})$. Thus, we have $f(t_0) = \omega$

By construction, $<$`output_arcs_weights`$(0) \Rightarrow \omega>$, and by property of the stabilize relation, we have $\sigma'(id_p)(''oaw'')[0] = \omega$. Thus, we can deduce that $g(0) = \omega$, and then we can rewrite the goal in order to apply the induction hypothesis with $Pr' = Pr(t, fired) \setminus \{t_0\}$.

❏ In the case where $t_0 \in T_s$:
As t is a top-priority transition in set T_s, there exists no transition $t' \in T_s$ s.t. $t' \succ t$.
Contradicts $t_0 \succ t$.

2. Assuming that $\prod_{i=0}^{\Delta(id_t)(''ian'')-1} \sigma'(id_t)(''pauths'')[i] = \mathtt{true}$, let us show

$$\boxed{t \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i)).}$$

By definition of $t \in Sens(s'.M - \sum_{t_i \in Pr(t,fired)} pre(t_i))$:

$$\boxed{\begin{aligned}&\forall p \in P, \omega \in \mathbb{N}^*,\\&\big((pre(p,t) = (\omega, \mathtt{basic}) \vee pre(p,t) = (\omega, \mathtt{test})) \Rightarrow s'.M(p) - \sum_{t_i \in Pr(t,fired)} pre(p,t_i) \geq \omega\big)\\&\wedge \big(pre(p,t) = (\omega, \mathtt{inhib}) \Rightarrow s'.M(p) - \sum_{t_i \in Pr(t,fired)} pre(p,t_i) < \omega\big)\end{aligned}}$$

Given a $p \in P$ and an $\omega \in \mathbb{N}^*$, let us show

$$\boxed{\begin{aligned}&\big((pre(p,t) = (\omega, \mathtt{basic}) \vee pre(p,t) = (\omega, \mathtt{test})) \Rightarrow s'.M(p) - \sum_{t_i \in Pr(t,fired)} pre(p,t_i) \geq \omega\big)\\&\wedge \big(pre(p,t) = (\omega, \mathtt{inhib}) \Rightarrow s'.M(p) - \sum_{t_i \in Pr(t,fired)} pre(p,t_i) < \omega\big)\end{aligned}}$$

By construction, there exists an $id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p$. By definition of id_p, there exist gm_p, ipm_p, opm_p s.t. $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$.

There are three different cases:

(a) Assuming that $pre(p,t) = (\omega, \mathtt{test})$, let us show $\boxed{s'.M(p) - \sum_{t_i \in Pr(t,fired)} pre(p,t_i) \geq \omega.}$

Then, assuming that the priority relation is well-defined, there exists no transition t_i connected by a `basic` arc to p that verified $t_i \succ t$. 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 output of p.
Then, we can deduce that $\sum_{t_i \in Pr(t,fired)} pre(p,t_i) = 0$.

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 $pre(p,t) = (\omega, \mathtt{inhib})$, let us show $\boxed{s'.M(p) - \sum_{t_i \in Pr(t,fired)} pre(p,t_i) < \omega.}$

Use the same strategy as above.

(c) Assuming that $pre(p,t) = (\omega, \texttt{basic})$, let us show $\boxed{s'.M(p) - \sum_{t_i \in Pr(t,fired)} pre(p,t_i) \geq \omega.}$

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 $\texttt{basic}$ arc that verifies $t_i \succ t$.
Then, we can deduce $\sum_{t_i \in Pr(t,fired)} pre(p,t_i) = 0$.
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 definition of id_t, there exist gm_t, ipm_t, opm_t s.t. $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$.
By construction, there exist $j \in [0, |input(t)| - 1]$, $k \in [0, |output(t)| - 1]$, and $id_{kj} \in Sigs(\Delta)$ s.t. $<\texttt{priority\_authorizations(j)} \Rightarrow \texttt{id}_{\texttt{kj}}> \in ipm_t$ and $<\texttt{priority\_authorizations(k)} \Rightarrow \texttt{id}_{\texttt{kj}}> \in opm_p$. Let us take such an j, k and id_{kj}.
From $\prod_{i=0}^{\Delta(id_t)(\text{"}ian\text{"})-1} \sigma'(id_t)(\text{"}pauths\text{"})[i] = \texttt{true}$, we can deduce that for all $i \in [0, \Delta(id_t)(\text{"}ian\text{"}) - 1]$, $\sigma'(id_t)(\text{"}pauths\text{"})[i] = \texttt{true}$.
By construction, $<\texttt{input\_arcs\_number} \Rightarrow |input(t)|> \in gm_t$, and by property of the elaboration relation, we have $\Delta(id_t)(\text{"}ian\text{"}) = |input(t)|$. Then, from $j \in [0, |input(t)| - 1]$, we can deduce $j \in [0, \Delta(id_t)(\text{"}ian\text{"}) - 1]$. And, from $\forall i \in [0, \Delta(id_t)(\text{"}ian\text{"}) - 1]$, $\sigma'(id_t)(\text{"}pauths\text{"})[i] = \texttt{true}$, we can deduce $\sigma'(id_t)(\text{"}pauths\text{"})[j] = \texttt{true}$.
By property of the stabilize relation, $\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$ and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_p)(\text{"}pauths\text{"})[k] = \sigma'(id_{kj})\sigma'(id_t)(\text{"}pauths\text{"})[j] = \texttt{true} \tag{1.192}$$

By property of the stabilize relation and $\texttt{comp}(id_p, \text{"}place\text{"}, gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_p)(\text{"}pauths\text{"})[k] = (\sigma'(id_p)(\text{"}sm\text{"}) \geq \texttt{rsum} + \sigma'(id_p)(\text{"}oaw\text{"})[k]) \tag{1.193}$$

Let us define the $\texttt{rsum}$ term as follows:

$$\texttt{rsum} = \sum_{i=0}^{k-1} \begin{cases} \sigma'(id_p)(\text{"}oaw\text{"})[i] \text{ if } \sigma'(id_p)(\text{"}otf\text{"})[i]. \\ \qquad\qquad \sigma'(id_p)(\text{"}oat\text{"})[i] = \texttt{basic} \\ 0 \text{ otherwise} \end{cases} \tag{1.194}$$

From (1.192) and (1.193), we can deduce that $\sigma'(id_p)(\text{"}sm\text{"}) \geq \texttt{rsum} + \sigma'(id_p)(\text{"}oaw\text{"})[k]$.
Then, there are three points to prove:

A. $\boxed{s'.M(p) = \sigma'(id_p)(\text{"}sm\text{"})}$

B. $\boxed{\omega = \sigma'(id_p)(\text{"}oaw\text{"})[k]}$

C. $\boxed{\sum_{t_i \in Pr(t,fired)} pre(p,t_i) = \texttt{rsum}}$

See 1 for the remainder of the proof.

□

Appendix A

Reminder on natural semantics

Appendix B

Reminder on induction principles

- Present all the material that will be used in the proof, and that needs clarifying for people who do not come from the field (e.g, automaticians and electronicians)
  - structural induction
  - induction on relations
  - ...