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 1, 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 $P_{id} = \{id \mid \texttt{comp}(id, \text{"}place\text{"}, gm, ipm, opm) \in behavior\}$
- $TMap \in T \rightarrow T_{id}$ where $T_{id} = \{id \mid \texttt{comp}(id, \text{"}transition\text{"}, gm, ipm, opm) \in behavior\}$
- $\mathcal{C}_{id} \subseteq \{id \mid (\texttt{in}, id, t) \in ports \wedge id \notin \{\text{"}clk\text{"}, \text{"}rst\text{"}\}\}$
- $\mathcal{A}_{id} \subseteq \{id \mid (\texttt{out}, id, t) \in ports\}$
- $\mathcal{F}_{id} \subseteq \{id \mid (\texttt{out}, id, t) \in ports\}$
- $CMap \in \mathcal{C} \rightarrow \mathcal{C}_{id}$
- $AMap \in \mathcal{A} \rightarrow \mathcal{A}_{id}$
- $FMap \in \mathcal{F} \rightarrow \mathcal{F}_{id}$

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. $\gamma(p) = id_p,\ s.M(p) = \sigma(id_p)(\text{"s\_marking"})$.
2. $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,
$\big(upper(I_s(t)) = \infty \land s.I(t) \leq lower(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(\text{"s\_time\_counter"})\big)$
$\land\big(upper(I_s(t)) = \infty \land s.I(t) > lower(I_s(t)) \Rightarrow \sigma(id_t)(\text{"s\_time\_counter"}) = lower(I_s(t))\big)$
$\land\big(upper(I_s(t)) \neq \infty \land s.I(t) > upper(I_s(t)) \Rightarrow \sigma(id_t)(\text{"s\_time\_counter"}) = upper(I_s(t))\big)$
$\land\big(upper(I_s(t)) \neq \infty \land s.I(t) \leq upper(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(\text{"s\_time\_counter"})\big)$.
3. $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, s.reset_t(t) = \sigma(id_t)(\text{"s\_reinit\_time\_counter"})$.
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. $\gamma(p) = id_p,\ s.M(p) = \sigma(id_p)(\text{"s\_marking"})$.
2. $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,
$\big(upper(I_s(t)) = \infty \land s.I(t) \leq lower(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(\text{"s\_time\_counter"})\big)$
$\land\big(upper(I_s(t)) = \infty \land s.I(t) > lower(I_s(t)) \Rightarrow \sigma(id_t)(\text{"s\_time\_counter"}) = lower(I_s(t))\big)$
$\land\big(upper(I_s(t)) \neq \infty \land s.I(t) > upper(I_s(t)) \Rightarrow \sigma(id_t)(\text{"s\_time\_counter"}) = upper(I_s(t))\big)$
$\land\big(upper(I_s(t)) \neq \infty \land s.I(t) \leq upper(I_s(t)) \Rightarrow s.I(t) = \sigma(id_t)(\text{"s\_time\_counter"})\big)$.
3. $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, s.reset_t(t) = \sigma(id_t)(\text{"s\_reinit\_time\_counter"})$.
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. $\gamma(t) = id_t, t \in Sens(s.M) \Leftrightarrow \sigma(id_t)(\text{"s\_enabled"}) = \texttt{true}$.

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:*

$$\textsc{SimTraceNil}\ \frac{}{\gamma \vdash [\,] \sim [\,]} \qquad \textsc{SimTraceCons}\ \frac{\gamma \vdash s \sim \sigma \qquad \gamma \vdash \theta_s \sim \theta_\sigma}{\gamma \vdash (s :: \theta_s) \sim (\sigma :: \theta_\sigma)}$$

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,* $* \in A \rightarrow A \rightarrow A$ *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* $f \in B \rightarrow A$ *and* $g \in C \rightarrow A$*, assume that:*

- *there exists an injection* $\iota \in B \rightarrow C$ *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 $\iota' \in B \setminus \{b\} \rightarrow C'$ 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 $\iota \in B \rightarrow C$ 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. $\boxed{\exists \text{ an injection } \iota' \in B \setminus \{b\} \rightarrow C \setminus \{\iota(b)\} \text{ s.t. } \forall b' \in B \setminus \{b\}, f(b') = g(\iota'(b'))}$

Let us define a $\iota' \in B \setminus \{b\} \to C \setminus \{\iota(b)\}$ 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

1.3 Initial States

1.4 First Rising Edge

1.5 Rising Edge

1.6 Falling Edge

Definition 8 (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* $\gamma \vdash E_p \overset{env}{=} E_c$ *and* $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash \mathrm{d} \xrightarrow{elab} \Delta, \sigma_e$
- $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$
- $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$
- $\mathtt{Inject}_\downarrow(\sigma, E_p, \tau, \sigma_i)$ *and* $\Delta, \sigma_i \vdash \mathrm{d.cs} \xrightarrow{\downarrow} \sigma_\downarrow$ *and* $\Delta, \sigma_\downarrow \vdash \mathrm{d.cs} \xrightarrow{\rightsquigarrow} \sigma'$
- *State σ is a stable design state:* $\mathcal{D}_{\mathcal{H}}, \Delta, \sigma \vdash \mathrm{d.cs} \xrightarrow{comb} \sigma$

Lemma 1 (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. 8, 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. $\gamma(t) = id_t, s'.reset_t(t) = \sigma'(id_t)(\text{"}s\_reinit\_time\_counter\text{"})$.
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. $\gamma(t) = id_t, t \in Firable(s') \Leftrightarrow \sigma'(id_t)(\text{"}s\_firable\text{"}) = \texttt{true}$.
8. $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, t \notin Firable(s') \Leftrightarrow \sigma'(id_t)(\text{"}s\_firable\text{"}) = \texttt{false}$.
9. $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, t \in Fired(s') \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = \texttt{true}$.
10. $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t, t \notin Fired(s') \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = \texttt{false}$.
11. $\forall p \in P, id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p,\ \sum\limits_{t \in Fired(s')} pre(p,t) = \sigma'(id_p)(\text{"}s\_output\_token\_sum\text{"})$.
12. $\forall p \in P, id_p \in Comps(\Delta)$ s.t. $\gamma(p) = id_p,\ \sum\limits_{t \in Fired(s')} post(t,p) = \sigma'(id_p)(\text{"}s\_input\_token\_sum\text{"})$.

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 2 (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. 8, 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 $E_c, \tau \vdash sitpn, s \xrightarrow{\downarrow} s'$:

$$s.M(p) = s'.M(p) \tag{1.5}$$

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

$$\sigma'(id_p)("s\_marking") = \sigma(id_p)("s\_marking") \tag{1.6}$$

Rewriting the goal with (1.5) and (1.6): $\boxed{s.M(p) = \sigma(id_p)("s\_marking").}$

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

□

Lemma 3 (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. 8, 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. $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$. By property of the stabilize relation and $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_p)("sots") = \sum_{i=0}^{\Delta(id_p)("oan")-1} \begin{cases} \sigma'(id_p)("oaw")[i] \texttt{ if } (\sigma'(id_p)("otf")[i] \\ \qquad . \sigma'(id_p)("oat")[i] = \texttt{BASIC}) \\ 0 \textit{ otherwise} \end{cases} \tag{1.7}$$

Rewriting the goal with (1.7):

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

Let us unfold the definition of the left sum term:

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

To ease the reading, let us define functions $f \in Fired(s') \to \mathbb{N}$ and $g \in [0, |output(p)| - 1] \to \mathbb{N}$ s.t.

$$f(t) = \begin{cases} \omega \texttt{ if } pre(p,t) = (\omega, \texttt{basic}) \\ 0 \; otherwise \end{cases} \quad \text{and } g(i) = \begin{cases} \sigma'(id_p)("oaw")[i] \texttt{ if } (\sigma'(id_p)("otf")[i] \\ \qquad\qquad . \; \sigma'(id_p)("oat")[i] = \texttt{BASIC}) \\ 0 \; otherwise \end{cases}$$

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

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

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

   By construction, $<$`output_arcs_number` $\Rightarrow$ `1`$> \in gm_p$, $<$`output_arcs_types(0)` $\Rightarrow$ `BASIC`$> \in ipm_p$, $<$`output_transitions_fired(0)` $\Rightarrow$ `true`$> \in ipm_p$, and $<$`output_arcs_weights(0)` $\Rightarrow$ `0`$> \in ipm_p$.
   By property of the elaboration relation and $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$:

$$\Delta(id_p)("oan") = 1 \tag{1.8}$$

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

$$\sigma'(id_p)("oat")[0] = \texttt{BASIC} \tag{1.9}$$
$$\sigma'(id_p)("otf")[0] = \texttt{true} \tag{1.10}$$
$$\sigma'(id_p)("oaw")[0] = 0 \tag{1.11}$$

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

$$\sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } pre(p,t) = (\omega, \texttt{basic}) \\ 0 \; otherwise \end{cases} = 0 \tag{1.12}$$

   Rewriting the goal with (1.8), (1.9), (1.10), (1.11) and (1.12), tautology.

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

   By construction, $<$`output_arcs_number` $\Rightarrow |output(p)|> \in gm_p$, and by property of the elaboration relation:

$$\Delta(id_p)("oan") = |output(p)| \tag{1.13}$$

Rewriting the goal with (1.13): $\boxed{\sum_{t \in Fired(s')} f(t) = \sum_{i=0}^{|output(p)|-1} g(i).}$

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:

  $$g(0) = \begin{cases} \sigma'(id_p)(''oaw'')[0] \texttt{ if } (\sigma'(id_p)(''otf'')[0] \\ \qquad\qquad\qquad\qquad . \sigma'(id_p)(''oat'')[0] = \texttt{BASIC}) \\ 0 \textit{ otherwise} \end{cases} \tag{1.14}$$

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

  (a) $(\sigma'(id_p)(''otf'')[0]$. $\sigma'(id_p)(''oat'')[0] = \texttt{BASIC}) = \texttt{false}$:
  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) $(\sigma'(id_p)(''otf'')[0]$. $\sigma'(id_p)(''oat'')[0] = \texttt{BASIC}) = \texttt{true}$:
  In that case, $g(0) = \sigma'(id_p)(''oaw'')[0]$, $\sigma'(id_p)(''otf'')[0] = \texttt{true}$ and $\sigma'(id_p)(''oat'')[0] = \texttt{BASIC}$.
  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. $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$.
  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, $\sigma'(id_p)(''oat'')[0] = \texttt{BASIC}$ and $<\texttt{output\_arcs\_types}(0) \Rightarrow \texttt{a}> \in ipm_p$:

$$pre(p,t) = (\omega, \texttt{basic}) \tag{1.15}$$

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

$$\sigma'(id_t)(''fired'') = \texttt{true} \tag{1.16}$$

Appealing to Lemma 14, 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 $<\texttt{output\_arcs\_weights}(0) \Rightarrow \omega> \in ipm_p$:

$$\sigma'(id_p)(''oaw'')[0] = \omega \tag{1.17}$$

Rewriting the goal with (1.17):

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

By definition of f, and as $pre(p,t) = (\omega, \texttt{basic})$, 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 4 (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. 8, 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. $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$.
By property of the stabilize relation and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_p)(''sits'') = \sum_{i=0}^{\Delta(id_p)(''ian'')-1} \begin{cases} \sigma'(id_p)(''iaw'')[i] \text{ if } \sigma'(id_p)(''itf'')[i] \\ 0 \text{ otherwise} \end{cases} \tag{1.18}$$

Rewriting the goal with (1.18):

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

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

$$\Delta(id_p)("ian") = 1 \tag{1.19}$$

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

$$\sigma'(id_p)("itf")[0] = \texttt{true} \tag{1.20}$$
$$\sigma'(id_p)("iaw")[0] = 0 \tag{1.21}$$

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

$$\sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } post(t,p) = \omega \\ 0 \textit{ otherwise} \end{cases} = 0 \tag{1.22}$$

   Rewriting the goal with (1.19), (1.20), (1.21), and (1.22), 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:

$$\Delta(id_p)("ian") = |input(p)| \tag{1.23}$$

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

Rewriting the goal with (1.23): $\boxed{\sum_{t \in Fired(s')} f(t) = \sum_{i=0}^{|input(p)|-1} g(i).}$

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:

  $$g(0) = \begin{cases} \sigma'(id_p)("iaw")[0] \text{ if } \sigma'(id_p)("itf")[0] \\ 0 \; otherwise \end{cases} \tag{1.24}$$

  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 $<\texttt{input\_arcs\_types}(0) \Rightarrow \texttt{a}> \in ipm_p$:

  $$post(t,p) = \omega \tag{1.25}$$

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

$$\sigma'(id_t)(''fired'') = \texttt{true} \tag{1.26}$$

Appealing to Lemma 14 and (1.26), 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 $g(0) = \sigma'(id_p)(''iaw'')[0]$, and by property of the stabilize relation and $<\texttt{input\_arcs\_weights}(0) \Rightarrow \omega> \in ipm_p$:

$$\sigma'(id_p)(''iaw'')[0] = \omega \tag{1.27}$$

Rewriting the goal with (1.27):

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

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 5 (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. 8, then $\forall t \in T_i, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$,*
$(upper(I_s(t)) = \infty \wedge s'.I(t) \leq lower(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(''s\_time\_counter''))$
$\wedge(upper(I_s(t)) = \infty \wedge s'.I(t) > lower(I_s(t)) \Rightarrow \sigma'(id_t)(''s\_time\_counter'') = lower(I_s(t)))$
$\wedge(upper(I_s(t)) \neq \infty \wedge s'.I(t) > upper(I_s(t)) \Rightarrow \sigma'(id_t)(''s\_time\_counter'') = upper(I_s(t)))$
$\wedge(upper(I_s(t)) \neq \infty \wedge s'.I(t) \leq upper(I_s(t)) \Rightarrow s'.I(t) = \sigma'(id_t)(''s\_time\_counter''))$.

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

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

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

$$\begin{aligned}\sigma(id_t)(\text{"}se\text{"}) = \mathtt{true} \wedge \Delta(id_t)(\text{"}tt\text{"}) \neq \mathtt{NOT\_TEMPORAL} \wedge \sigma(id_t)(\text{"}srtc\text{"}) = \mathtt{false}\\ \wedge \sigma(id_t)(\text{"}stc\text{"}) < \Delta(id_t)(\text{"}mtc\text{"}) \Rightarrow \sigma'(id_t)(\text{"}stc\text{"}) = \sigma(id_t)(\text{"}stc\text{"}) + 1\end{aligned} \tag{1.28}$$

$$\begin{aligned}\sigma(id_t)(\text{"}se\text{"}) = \mathtt{true} \wedge \Delta(id_t)(\text{"}tt\text{"}) \neq \mathtt{NOT\_TEMPORAL} \wedge \sigma(id_t)(\text{"}srtc\text{"}) = \mathtt{false}\\ \wedge \sigma(id_t)(\text{"}stc\text{"}) \geq \Delta(id_t)(\text{"}mtc\text{"}) \Rightarrow \sigma'(id_t)(\text{"}stc\text{"}) = \sigma(id_t)(\text{"}stc\text{"})\end{aligned} \tag{1.29}$$

$$\begin{aligned}\sigma(id_t)(\text{"}se\text{"}) = \mathtt{true} \wedge \Delta(id_t)(\text{"}tt\text{"}) \neq \mathtt{NOT\_TEMPORAL}\\ \wedge \sigma(id_t)(\text{"}srtc\text{"}) = \mathtt{true} \Rightarrow \sigma'(id_t)(\text{"}stc\text{"}) = 1\end{aligned} \tag{1.30}$$

$$\sigma(id_t)(\text{"}se\text{"}) = \mathtt{false} \vee \Delta(id_t)(\text{"}tt\text{"}) = \mathtt{NOT\_TEMPORAL} \Rightarrow \sigma'(id_t)(\text{"}stc\text{"}) = 0 \tag{1.31}$$

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 $\boxed{s'.I(t) = \sigma'(id_t)(\text{"}s\_time\_counter\text{"}).}$

   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 $\sigma(id_t)(\text{"}se\text{"}) = \mathtt{false}$ (1.32).
   Appealing to (1.31) and (1.32), we have $\sigma'(id_t)(\text{"}stc\text{"}) = 0$ (1.33).
   By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = 0$ (1.34).
   Rewriting the goal with (1.33) and (1.34): tautology.

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

   By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(\text{"}se\text{"}) = \mathtt{true}$ (1.35).
   By construction, and as $upper(I_s(t)) = \infty$, $<\mathtt{transition\_type} \Rightarrow \mathtt{TEMP\_A\_INF}> \in gm_t$. By property of the elaboration relation, we have $\Delta(id_t)(\text{"}tt\text{"}) = \mathtt{TEMP\_A\_INF}$ (1.36).
   Case analysis on $s.reset_t(t)$; there are two cases:

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

   By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, $\sigma(id_t)(\text{"}srtc\text{"}) = \mathtt{true}$ (1.37).
   Appealing to (1.30), (1.35), (1.36) and (1.37), we have $\sigma'(id_t)(\text{"}stc\text{"}) = 1$ (1.38).
   By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = 1$ (1.39).
   Rewriting the goal with (1.38) and (1.39): tautology.

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

   By definition of $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$, we have $\sigma(id_t)(\text{"}srtc\text{"}) = \mathtt{false}$ (1.40).

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 by property of the elaboration relation, we have $\Delta(id_t)(''mtc'') = a$ (1.41).

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 $upper(I_s(t)) = \infty$:

$$s'.I(t) = s.I(t) + 1 \tag{1.42}$$

Rewriting the goal with (1.42): $\boxed{s.I(t) + 1 = \sigma'(id_t)(''stc'').}$

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 $upper(I_s(t)) = \infty$:

$$s.I(t) = \sigma(id_t)(''stc'') \tag{1.43}$$

Appealing to (1.41), (1.43) and $s.I(t) < a$:

$$\sigma(id_t)(''stc'') < \Delta(id_t)(''mtc'') \tag{1.44}$$

Appealing to (1.28), (1.44), (1.40) and (1.35):

$$\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'') + 1 \tag{1.45}$$

Rewriting the goal with (1.45) and (1.43): tautology.

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

   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:

$$\begin{aligned} \Delta(id_t)(''mtc'') &= a & (1.46)\\ \Delta(id_t)(''tt'') &= \texttt{TEMP\_A\_INF} & (1.47) \end{aligned}$$

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

$$\sigma(id_t)(''se'') = \texttt{true} \tag{1.48}$$

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

$$\sigma(id_t)(\text{"srtc"}) = \texttt{false} \tag{1.49}$$

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

$$\begin{aligned} s'.I(t) = s.I(t) + 1 &\Rightarrow s.I(t) + 1 > lower(I_s(t)) \\ &\Rightarrow s.I(t) \geq lower(I_s(t)) \end{aligned} \tag{1.50}$$

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

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

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

$$\sigma(id_t)(\text{"stc"}) = lower(I_s(t)) \tag{1.51}$$

Appealing to (1.29):

$$\sigma'(id_t)(\text{"stc"}) = \sigma(id_t)(\text{"stc"}) \tag{1.52}$$

Rewriting the goal with (1.51) and (1.52): tautology.

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

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

$$s.I(t) = \sigma(id_t)(\text{"stc"}) \tag{1.53}$$

Appealing to (1.29):

$$\sigma'(id_t)(\text{"stc"}) = \sigma(id_t)(\text{"stc"}) \tag{1.54}$$

Rewriting the goal with (1.54), (1.53) and $s.I(t) = lower(I_s(t))$: tautology.

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

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

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:

$$\Delta(id_t)(\text{"mtc"}) = b = upper(I_s(t)) \tag{1.55}$$

$$\Delta(id_t)(\text{"tt"}) \neq \texttt{NOT\_TEMP} \tag{1.56}$$

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

$$\sigma(id_t)(\text{"}se\text{"}) = \texttt{true} \tag{1.57}$$

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

$$\sigma(id_t)(\text{"}srtc\text{"}) = \texttt{false} \tag{1.58}$$

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

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

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

$$s'.I(t) = s.I(t) \tag{1.59}$$

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

$$\sigma(id_t)(\text{"}stc\text{"}) = upper(I_s(t)) \tag{1.60}$$

Appealing to (1.29), we have $\sigma'(id_t)(\text{"}stc\text{"}) = \sigma(id_t)(\text{"}stc\text{"})$.
Rewriting the goal with $\sigma'(id_t)(\text{"}stc\text{"}) = \sigma(id_t)(\text{"}stc\text{"})$ and (1.60): tautology.

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

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

$$s.I(t) = \sigma(id_t)(\text{"}stc\text{"}) \tag{1.61}$$

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

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

  Appealing to (1.55), (1.61) and $s.I(t) = upper(I_s(t))$:

$$\Delta(id_t)(\text{"}mtc\text{"}) \leq \sigma(id_t)(\text{"}stc\text{"}) \tag{1.62}$$

Appealing to (1.62) and (1.29):

$$\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'') \tag{1.63}$$

Rewriting the goal with (1.63), (1.61) and $s.I(t) = upper(I_s(t))$: tautology.

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

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

  $$s'.I(t) = s.I(t) + 1 \tag{1.64}$$

  From (1.64) 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:

$$\Delta(id_t)(''mtc'') = b = upper(I_s(t)) \tag{1.65}$$
$$\Delta(id_t)(''tt'') \neq \texttt{NOT\_TEMP} \tag{1.66}$$

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

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

By definition of $\gamma, E_c, \tau \vdash s \xrightarrow{\uparrow} \sigma$, we have $\sigma(id_t)(''se'') = \texttt{false}$ (1.67).
Appealing (1.31) and (1.67), we have $\sigma'(id_t)(''stc'') = 0$ (1.68).

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = 0$ (1.69).
Rewriting the goal with (1.68) and (1.69): tautology.

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

By definition of $\gamma, E_c, \tau \vdash s \xrightarrow{\uparrow} \sigma$, we have $\sigma(id_t)(''se'') = \texttt{true}$ (1.70).
Case analysis on $s.reset_t(t)$:

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

By definition of $\gamma, E_c, \tau \vdash s \xrightarrow{\uparrow} \sigma$, we have $\sigma(id_t)(''srtc'') = \texttt{true}$ (1.71).
Appealing to (1.30), (1.66), (1.70) and (1.71), we have $\sigma'(id_t)(''stc'') = 1$ (1.72).

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, we have $s'.I(t) = 1$ (1.73).
Rewriting the goal with (1.72) and (1.73), tautology.

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

By definition of $\gamma, E_c, \tau \vdash s \xrightarrow{\uparrow} \sigma$, we have $\sigma(id_t)(''srtc'') = \texttt{false}$ (1.74).
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 $s.I(t) = \sigma(id_t)(''stc'')$ (1.75).

- $s.I(t) < upper(I_s(t))$:
  From $s.I(t) < upper(I_s(t))$, (1.75) and (1.65), we can deduce $\sigma(id_t)(''stc'') < \Delta(id_t)(''mtc'')$ (1.76).
  From (1.28), (1.70), (1.66), (1.74) and (1.76), we can deduce:

$$\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'') + 1 \tag{1.77}$$

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

$$s'.I(t) = s.I(t) + 1 \tag{1.78}$$

  Rewriting the goal with (1.77) and (1.78), 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 6 (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. 8, 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 $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_t)(''srtc'') = \sum_{i=0}^{\Delta(id_t)(''ian'')-1} \sigma'(id_t)(''rt'')[i] \tag{1.79}$$

□

1.6.4 Falling edge and condition values

Lemma 7 (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. 8, 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 $s'.cond(c) = E_c(\tau, c)$ (1.80).
By property of the $\texttt{Inject}_{\downarrow}$, the $\mathcal{H}$-VHDL falling edge, the stabilize relations and $id_c \in Ins(\Delta)$, we have $\sigma'(id_c) = E_p(\tau, \downarrow)(id_c)$ (1.81).

Rewriting the goal with (1.80) and (1.81): $\boxed{E_c(\tau, c) = E_p(\tau, \downarrow)(id_c)}$

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 8 (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. 8, 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 $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$:

$$s'.ex(a) = \sum_{p \in marked(s.M)} \mathbb{A}(p, a) \tag{1.82}$$

By construction, the ‘‘$\texttt{action}$’’ 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 ‘‘$\texttt{action}$’’ 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 ‘‘$\texttt{action}$’’ process body executed during the falling edge phase.

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

  $$\sigma'(id_a) = false \tag{1.83}$$

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

  $$\sum_{p \in marked(s.M)} \mathbb{A}(p, a) = \texttt{false} \tag{1.84}$$

  Rewriting the goal with (1.82), (1.83) and (1.84), 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 ‘‘$\texttt{action}$’’ 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 $\texttt{ps}(''action'', \emptyset, sl, ss) \in d.cs$:

  $$\sigma'(id_a) = \sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}) \tag{1.85}$$

Rewriting the goal with (1.82) and (1.85), $\boxed{\sum_{p \in marked(s.M)} \mathbb{A}(p,a) = \sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}).}$

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

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

  To prove the above goal, let us show $\boxed{\exists p \in marked(s.M) \ s.t. \ \mathbb{A}(p,a) = \mathtt{true}.}$
  From $\sigma(id_{mp_0}) + \cdots + \sigma(id_{mp_n}) = \mathtt{true}$, we can deduce that $\exists id_{mp_i} \ s.t. \ \sigma(id_{mp_i}) = \mathtt{true}$. Let us take an id_{mp_i} s.t. $\sigma(id_{mp_i}) = \mathtt{true}$.
  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 $\mathtt{comp}(id_{p_i}, ''place'', gm_{p_i}, ipm_{p_i}, opm_{p_i}) \in d.cs$ and $<\mathtt{marked} \Rightarrow \mathtt{id_{mp_i}}> \in opm_{p_i}$. 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 $\mathtt{comp}(id_{p_i}, ''place'', gm_{p_i}, ipm_{p_i}, opm_{p_i}) \in d.cs$:

$$\sigma(id_{mp_i}) = \sigma(id_{p_i})(''marked'') \tag{1.86}$$
$$\sigma(id_{p_i})(''marked'') = \sigma(id_{p_i})(''sm'') > 0 \tag{1.87}$$

  From (1.86), (1.87) and $\sigma(id_{mp_i}) = \mathtt{true}$, we can deduce that $\sigma(id_{p_i})(''marked'') = \mathtt{true}$ and $(\sigma(id_{p_i})(''sm'') > 0) = \mathtt{true}$.

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

$$s.M(p_i) = \sigma(id_{p_i})(''sm'') \tag{1.88}$$

  From (1.88) and $(\sigma(id_{p_i})(''sm'') > 0) = \mathtt{true}$, 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: $\boxed{\mathbb{A}(p,a) = \mathtt{true}.}$

  By definition of $p_i \in pls(a)$, $\mathbb{A}(p,a) = \mathtt{true}$.

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

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

  Given a $p \in marked(s.M)$, let us show $\boxed{\mathbb{A}(p,a) = \mathtt{false}.}$

  Let us perform case analysis on $\mathbb{A}(p,a)$; there are 2 cases:

  * **CASE** $\mathbb{A}(p,a) = \mathtt{false}$.
  * **CASE** $\mathbb{A}(p,a) = \mathtt{true}$:
    By construction, for all $p \in P$ s.t. $\mathbb{A}(p,a) = \mathtt{true}$, 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 $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$ and $<\mathtt{marked} \Rightarrow \mathtt{id_{mp_i}}> \in opm_p$. Let us take such a id_p, gm_p, ipm_p, opm_p and id_{mp_i}.
    By property of stable σ and $\mathtt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma(id_{mp_i}) = \sigma(id_p)(''marked'') \tag{1.89}$$
$$\sigma(id_p)(''marked'') = \sigma(id_p)(''sm'') > 0 \tag{1.90}$$

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

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

□

1.6.6 Falling edge and function executions

Lemma 9 (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. 8, 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 $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$:

$$s.ex(f) = s'.ex(f) \tag{1.91}$$

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:

$$\sigma(id_f) = \sigma'(id_f) \tag{1.92}$$

Rewriting the goal with (1.91) and (1.92), $\boxed{s.ex(f) = \sigma(id_f).}$

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

1.6.7 Falling edge and firable transitions

Lemma 10 (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. 8, 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. Given a $t \in T$ and $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, let us show that $\boxed{t \in Firable(s') \Leftrightarrow \sigma'(id_t)(\text{"}s\_firable\text{"}) = \texttt{true}.}$

The proof is in two parts:

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

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

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

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

□

Lemma 11 (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. 8, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $t \in Firable(s') \Rightarrow \sigma'(id_t)(\text{"}s\_firable\text{"}) = \texttt{true}$.*

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

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

$$\sigma'(id_t)(\text{"}sfa\text{"}) = \sigma(id_t)(\text{"}se\text{"}) \,.\, \sigma(id_t)(\text{"}scc\text{"}) \,.\, \mathtt{checktc}(\Delta(id_t), \sigma(id_t)) \tag{1.93}$$

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

$$\begin{aligned}
\mathtt{checktc}(\Delta(id_t), \sigma(id_t)) = \Big(& \mathtt{not}\ \sigma(id_t)(\text{"}srtc\text{"}) \,. \\
& \big[\big(\Delta(id_t)(\text{"}tt\text{"}) = \mathtt{TEMP_A_B} \,.\, (\sigma(id_t)(\text{"}stc\text{"}) \geq \sigma(id_t)(\text{"}A\text{"}) - 1) \\
& \qquad\qquad\qquad\qquad .\, (\sigma(id_t)(\text{"}stc\text{"}) \leq \sigma(id_t)(\text{"}B\text{"}) - 1)\big) \\
& + (\Delta(id_t)(\text{"}tt\text{"}) = \mathtt{TEMP_A_A} \,.\, (\sigma(id_t)(\text{"}stc\text{"}) = \sigma(id_t)(\text{"}A\text{"}) - 1)) \\
& + (\Delta(id_t)(\text{"}tt\text{"}) = \mathtt{TEMP_A_INF} \,.\, (\sigma(id_t)(\text{"}stc\text{"}) \geq \sigma(id_t)(\text{"}A\text{"}) - 1))\big]\Big) \\
& + \big(\sigma(id_t)(\text{"}srtc\text{"}) \,.\, \Delta(id_t)(\text{"}tt\text{"}) \neq \mathtt{NOT_TEMP} \,.\, \sigma(id_t)(\text{"}A\text{"}) = 1\big) \\
& + \Delta(id_t)(\text{"}tt\text{"}) = \mathtt{NOT_TEMP}
\end{aligned} \tag{1.94}$$

Rewriting the goal with (1.93): $\boxed{\sigma(id_t)(\text{"}se\text{"}) \,.\, \sigma(id_t)(\text{"}scc\text{"}) \,.\, \mathtt{checktc}(\Delta(id_t), \sigma(id_t)) = \mathtt{true}.}$ Then, there are three points to prove:

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

   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 $\sigma(id_t)(\text{"}se\text{"}) = \mathtt{true}.$

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

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

$$\sigma(id_t)(\text{"}scc\text{"}) = \prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \mathtt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} \tag{1.95}$$

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

   Rewriting the goal with (1.95): $\boxed{\prod_{c \in conds(t)} \begin{cases} E_c(\tau, c) & if\ \mathbb{C}(t,c) = 1 \\ \mathtt{not}(E_c(\tau, c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} = \mathtt{true}.}$

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

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

- **BASE CASE**: *true* $=$ *true*.
- **INDUCTION CASE**:

$$\prod_{c' \in conds(t) \backslash \{c\}} f(c') = \mathtt{true}$$

$\boxed{f(c) \,.\, \prod_{c' \in conds(t) \backslash \{c\}} f(c') = \mathtt{true}.}$

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

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

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

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

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

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

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

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

(a) $t \notin T_i$:

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

(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 $tt \in \{\mathtt{TEMP\_A\_B}, \mathtt{TEMP\_A\_A}, \mathtt{TEMP\_A\_INF}\}$ s.t. $<\mathtt{transition\_type} \Rightarrow tt> \in gm_t$. 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:

$$\begin{aligned}\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \Big(&\texttt{not}\ \sigma(id_t)(''srtc'')\ .\\ &\big[(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B}\ .\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1)\\ &\qquad .\ (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1))\\ &+(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_A}\ .\\ &\quad (\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1))\\ &+(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_INF}\ .\\ &\quad (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1))\big]\Big)\\ &+ \big(\sigma(id_t)(''srtc'')\ .\ \sigma(id_t)(''A'') = 1\big)\end{aligned} \tag{1.96}$$

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:

$$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \big(\sigma(id_t)(''A'') = 1\big) \tag{1.97}$$

Rewriting the goal with (1.97), 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:

$$\begin{gathered}\texttt{checktc}(\Delta(id_t), \sigma(id_t))\\ =\\ \begin{aligned}&(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B}\ .\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1)\\ &\qquad .\ (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1))\\ &+(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_A}\ .\ (\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1))\\ &+(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_INF}\ .\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1))\end{aligned}\end{gathered} \tag{1.98}$$

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 $<\texttt{transition\_type} \Rightarrow \texttt{TEMP\_A\_A}> \in gm_t$. By property of the elaboration relation, we have
    $\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_A}$; thus we can simplify the term `checktc` as follows:

$$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = (\sigma(id_t)(''stc'') = \sigma(id_t)(''A'') - 1) \tag{1.99}$$

    Rewriting the goal with (1.99), 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, $<\texttt{time\_A\_value} \Rightarrow a> \in ipm_t$, 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 $<\texttt{transition\_type} \Rightarrow \texttt{TEMP\_A\_B}> \in gm_t$. By property of the elaboration relation, we have
    $\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B}$; thus we can simplify the term `checktc` as follows:

$$\begin{gathered}\texttt{checktc}(\Delta(id_t), \sigma(id_t)) \\ = \\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1) \,.\, (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1)\end{gathered} \tag{1.100}$$

    Rewriting the goal with (1.100), 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:

$$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1) \tag{1.101}$$

  Rewriting the goal with (1.101), 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 12 (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. 8, then $\forall t \in T, id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $\sigma'(id_t)(''s\_firable'') = \texttt{true} \Rightarrow t \in Firable(s')$.*

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

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 $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_t)(''sfa'') = \sigma(id_t)(''se'')\ .\ \sigma(id_t)(''scc'')\ .\ \texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true} \tag{1.102}$$

From (1.102), we can deduce:

$$\sigma(id_t)(''se'') = \texttt{true} \tag{1.103}$$

$$\sigma(id_t)(''scc'') = \texttt{true} \tag{1.104}$$

$$\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true} \tag{1.105}$$

Term $\texttt{checktc}(\Delta(id_t), \sigma(id_t))$ as the same definition as in Lemma Falling Edge Equal Firable 1. By definition of $t \in Firable(s')$, there are three points to prove:

1. $\boxed{t \in Sens(s'.M)}$
2. $\boxed{t \notin T_i \vee s'.I(t) \in I_s(t)}$
3. $\boxed{\forall c \in \mathcal{C},\ \mathbb{C}(t,c) = 1 \Rightarrow s'.cond(c) = \texttt{true} \text{ and } \mathbb{C}(t,c) = -1 \Rightarrow s'.cond(c) = \texttt{false}}$

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 $\sigma(id_t)(''se'') = \texttt{true} \Leftrightarrow t \in Sens(s.M)$.

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

2. $\boxed{\forall c \in \mathcal{C},\ \mathbb{C}(t,c) = 1 \Rightarrow s'.cond(c) = \texttt{true} \text{ and } \mathbb{C}(t,c) = -1 \Rightarrow s'.cond(c) = \texttt{false}}$

   Given a $c \in \mathcal{C}$, there are two points to prove:

   (a) $\boxed{\mathbb{C}(t,c) = 1 \Rightarrow s'.cond(c) = \texttt{true}.}$

   (b) $\boxed{\mathbb{C}(t,c) = -1 \Rightarrow s'.cond(c) = \texttt{false}.}$

   Let us prove these two points:

(a) Assuming that $\mathbb{C}(t,c) = 1$, let us show $\boxed{s'.cond(c) = \texttt{true}.}$

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

$$\sigma(id_t)(''scc'') = \prod_{c \in conds(t)} \begin{cases} E_c(\tau,c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau,c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} = \texttt{true} \tag{1.106}$$

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:

$$E_c(\tau,c)\ .\prod_{c' \in conds(t) \setminus \{c\}} \begin{cases} E_c(\tau,c') & if\ \mathbb{C}(t,c') = 1 \\ \texttt{not}(E_c(\tau,c')) & if\ \mathbb{C}(t,c') = -1 \end{cases} = \texttt{true} \tag{1.107}$$

From (1.107), we can deduce that $E_c(\tau,c) = \texttt{true}$.

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 $E_c(\tau,c) = \texttt{true}$: tautology.

(b) Assuming that $\mathbb{C}(t,c) = -1$, let us show $\boxed{s'.cond(c) = \texttt{false}.}$

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

$$\sigma(id_t)(''scc'') = \prod_{c \in conds(t)} \begin{cases} E_c(\tau,c) & if\ \mathbb{C}(t,c) = 1 \\ \texttt{not}(E_c(\tau,c)) & if\ \mathbb{C}(t,c) = -1 \end{cases} = \texttt{true} \tag{1.108}$$

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:

$$\texttt{not}\ E_c(\tau,c)\ .\prod_{c' \in conds(t) \setminus \{c\}} \begin{cases} E_c(\tau,c') & if\ \mathbb{C}(t,c') = 1 \\ \texttt{not}(E_c(\tau,c')) & if\ \mathbb{C}(t,c') = -1 \end{cases} = \texttt{true} \tag{1.109}$$

From (1.109), we can deduce that $E_c(\tau,c) = \texttt{false}$.

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 $E_c(\tau,c) = \texttt{false}$: tautology.

3. $\boxed{t \notin T_i \vee s'.I(t) \in I_s(t)}$

Reasoning on $\texttt{checktc}(\Delta(id_t), \sigma(id_t)) = \texttt{true}$, there are 3 cases:

(a) $\left(\texttt{not}\ \sigma(id_t)(''srtc'')\ .\ [\dots]\right) = \texttt{true}$[1]

(b) $\left(\sigma(id_t)(''srtc'')\ .\ \Delta(id_t)(''tt'') \neq \texttt{NOT\_TEMP}\ .\ \sigma(id_t)(''A'') = 1\right) = \texttt{true}$

(c) $\left(\Delta(id_t)(''tt'') = \texttt{NOT\_TEMP}\right) = \texttt{true}$

(a) $\left(\texttt{not}\ \sigma(id_t)(''srtc'')\ .\ [\dots]\right) = \texttt{true}$:

[1] See equation (1.94) for the full definition

Then, we can deduce $\texttt{not}\ \sigma(id_t)(''srtc'') = \texttt{true}$ and $[\dots] = \texttt{true}$. From $\texttt{not}\ \sigma(id_t)(''srtc'') = \texttt{true}$, we can deduce $\sigma(id_t)(''srtc'') = \texttt{false}$, and from $[\dots] = \texttt{true}$, we have three other cases:

i. $\big(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B}\ .\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1)\ .\ (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1)\big) = \texttt{true}$

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. $\big(\Delta(id_t)(''tt'') = \texttt{TEMP\_A\_B}\ .\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1)\ .\ (\sigma(id_t)(''stc'') \leq \sigma(id_t)(''B'') - 1)\big) = \texttt{true}$:

   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` $\Rightarrow a$>, 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 $\sigma(id_t)(''se'') = \mathtt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)(''srtc'') = \mathtt{false}$, we can deduce $s.reset_t(t) = \mathtt{false}$. 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. $(\Delta(id_t)(''tt'') = \mathtt{TEMP\_A\_INF}\ .\ (\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1)) = \mathtt{true}$:
Then, converting boolean equalities into intuitionistic predicates, we have:

- $\Delta(id_t)(''tt'') = \mathtt{TEMP\_A\_INF}$
- $\sigma(id_t)(''stc'') \geq \sigma(id_t)(''A'') - 1$

By property of the elaboration relation, and $\Delta(id_t)(''tt'') = \mathtt{TEMP\_A\_INF}$, 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` $\Rightarrow a$>, 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 $\sigma(id_t)(''se'') = \mathtt{true}$, we can deduce $t \in Sens(s.M)$, and from $\sigma(id_t)(''srtc'') = \mathtt{false}$, we can deduce $s.reset_t(t) = \mathtt{false}$. 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, $<\texttt{time\_A\_value} \Rightarrow a> \in ipm_t$, 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 13 (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. 8, 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 14 (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. 8, 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 $\boxed{t \in Fired(s') \Leftrightarrow \sigma'(id_t)(''fired'') = \texttt{true}.}$ The proof is in two parts:

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

   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 $\sigma'(id_t)(''fired'') = \texttt{true}$, let us show $\boxed{t \in Fired(s').}$

   By definition of $t \in Fired(s')$, let us show that $\boxed{\exists fset \subseteq T \text{ s.t. } IsFiredSet(s', fset) \ \wedge \ t \in fset}$

   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.

   Add ref. defined S

$\square$

Lemma 15 (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. 8, then $\forall t, id_t$ s.t. $\gamma(t) = id_t$, $t \notin Fired(s') \Leftrightarrow \sigma'_t(''fired'') = \texttt{false}$.*

Proof. Proving the above lemma is trivial by appealing to Lemma Falling Edge Equal Fired and by reasoning on contrapositives. $\square$

Lemma 16 (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. 8, then $\forall t \in T$, $id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t$, $\forall fset \subseteq T$, s.t. $IsFiredSet(s', fset)$, $t \in fset \Leftrightarrow \sigma'(id_t)(''fired'') = true$.*

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 $\boxed{t \in fset \Leftrightarrow \sigma'(id_t)(''fired'') = true.}$

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):

$$\boxed{\begin{array}{l}\forall t' \in T, id_{t'} \in Comps(\Delta) \text{ s.t. } \gamma(t') = id_{t'},\\ (t' \in \emptyset \Rightarrow \sigma'(id_{t'})(''fired'') = \texttt{true}) \wedge (\sigma'(id_{t'})(''fired'') = \texttt{true} \Rightarrow t' \in \emptyset \ \vee \ t' \in T).\end{array}}$$

Given a $t' \in T$ and an $id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$, there are two points to prove:

1. $\boxed{t' \in \emptyset \Rightarrow \sigma'(id_{t'})(''fired'') = \texttt{true}}$

2. $\boxed{\sigma'(id_{t'})(''fired'') = \texttt{true} \Rightarrow t' \in \emptyset \ \vee \ t' \in T}$

Let us show these two points:

1. Assuming $t' \in \emptyset$, let us show $\boxed{\sigma'(id_{t'})(''fired'') = \texttt{true}.}$

   $t' \in \emptyset$ is a contradiction.

2. Assuming $\sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true}$, let us show $\boxed{t' \in \emptyset \lor t' \in T.}$

   By definition, $t' \in T$.

$\square$

Lemma 17 (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. 8, 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. $\gamma(t') = id_{t'}$,
  $(t' \in fired \Rightarrow \sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true}) \land (\sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true} \Rightarrow t' \in fired \lor t' \in T_s)$.

then $t \in fset \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = \texttt{true}$.

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 $\boxed{t \in fset \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = \texttt{true}.}$
Let us reason by induction on $IsFiredSetAux(s', fired, T_s, fset)$.

- **BASE CASE:** $\boxed{t \in fired \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = \texttt{true}.}$

  In that case, $fired = fset$ and $T_s = \emptyset$, EH looks like this:
  $\forall t' \in T,\ id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
  $(t' \in fired \Rightarrow \sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true}) \land (\sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true} \Rightarrow t' \in fired \lor t' \in \emptyset)$.

  From EH, we can deduce $t \in fired \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = \texttt{true}$.

- **INDUCTION CASE:** $\boxed{t \in fset \Leftrightarrow \sigma'(id_t)(\text{"}fired\text{"}) = \texttt{true}.}$

  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. $\gamma(t') = id_{t'}$,
  $(t' \in fired' \Rightarrow \sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true}) \land (\sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true} \Rightarrow t' \in fired' \lor t' \in T_s \setminus tp)) \Rightarrow$
  $t \in fset \Leftrightarrow \sigma'_t(\text{"}fired\text{"}) = true$.

  Applying the induction hypothesis, then, the new goal is:

  $\forall t' \in T, id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$,
  $(t' \in fired' \Rightarrow \sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true})$
  $\land(\sigma'(id_{t'})(\text{"}fired\text{"}) = \texttt{true} \Rightarrow t' \in fired' \lor t' \in T_s \setminus tp)$

Apply Lemma Elect Fired Equal Fired to solve the goal.

□

Lemma 18 (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. 8, 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.* $\gamma(t') = id_{t'}$, $(t' \in fired \Rightarrow \sigma'(id_{t'})(''fired'') = \texttt{true}) \wedge (\sigma'(id_{t'})(''fired'') = \texttt{true} \Rightarrow t' \in fired\ \vee\ t' \in T_s)$

then $\forall t \in T, id_t \in Comps(\Delta)$ *s.t.* $\gamma(t) = id_t$, $\big(t \in fired' \Rightarrow \sigma'(id_t)(''fired'') = \texttt{true}\big) \wedge \big(\sigma'(id_t)(''fired'') = \texttt{true} \Rightarrow t \in fired' \vee t \in T_s \setminus tp\big)$.

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

$$\boxed{\big(t \in fired' \Rightarrow \sigma'(id_t)(''fired'') = \texttt{true}\big) \wedge \big(\sigma'(id_t)(''fired'') = \texttt{true} \Rightarrow t \in fired' \vee t \in T_s \setminus tp\big).}$$

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**:

   $$\boxed{\big(t \in fired \Rightarrow \sigma'(id_t)(''fired'') = \texttt{true}\big) \wedge \big(\sigma'(id_t)(''fired'') = \texttt{true} \Rightarrow t \in fired \vee t \in T_s\big).}$$

   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:

   - $IsTopPrioritySet(T_s, \{t_0\} \cup tp_0)$
   - $ElectFired(s', fired \cup \{t_0\}, tp_0, fired')$
   - $IsFiredSetAux(s', fired', T_s \setminus \{t_0\} \cup tp_0, fset)$
   - $t_0 \in Firable(s')$
   - $t_0 \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i))$
   - 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. $\gamma(t') = id_{t'},$
$(t' \in fired \cup \{t_0\} \Rightarrow \sigma'_{t'}(''f'') = \mathtt{true}) \wedge (\sigma'(id_{t'})(''f'') = \mathtt{true} \Rightarrow t' \in fired \cup \{t_0\} \ \vee \ t' \in T_s')) \Rightarrow$
$\forall t \in T,\ id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t,$
$(t \in fired' \Rightarrow \sigma'(id_t)(''f'') = \mathtt{true}) \wedge (\sigma'(id_t)(''f'') = \mathtt{true} \Rightarrow t \in fired' \vee t \in T_s' \setminus tp_0)$

$\forall t \in T,\ id_t \in Comps(\Delta)$ s.t. $\gamma(t) = id_t,$
$(t \in fired' \Rightarrow \sigma'_t(''f'') = \mathtt{true}) \wedge (\sigma'_t(''f'') = \mathtt{true} \Rightarrow t \in fired' \vee t \in T_s \setminus \{t_0\} \cup tp_0)$

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. $\gamma(t') = id_{t'},$
$(t' \in fired \cup \{t_0\} \Rightarrow \sigma'_{t'}(''f'') = \mathtt{true}) \wedge (\sigma'(id_{t'})(''f'') = \mathtt{true} \Rightarrow t' \in fired \cup \{t_0\} \ \vee \ t' \in T_s \setminus \{t_0\})$

Let us prove these three points:

(a) $\boxed{IsTopPrioritySet(T_s \setminus \{t_0\}, tp_0)}$

Not possible to prove right now.

(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. $\gamma(t') = id_{t'},$
$(t' \in fired \cup \{t_0\} \Rightarrow \sigma'(id_{t'})(''f'') = \mathtt{true}) \wedge (\sigma'(id_{t'})(''f'') = \mathtt{true} \Rightarrow t' \in fired \cup \{t_0\} \ \vee \ t' \in T_s \setminus \{t_0\})$

Given a $t' \in T$ and an $id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$, let us show
$(t' \in fired \cup \{t_0\} \Rightarrow \sigma'(id_{t'})(''f'') = \mathtt{true})$
$\wedge (\sigma'(id_{t'})(''f'') = \mathtt{true} \Rightarrow t' \in fired \cup \{t_0\} \ \vee \ t' \in T_s \setminus \{t_0\})$.
The proof is in two parts.

i. Assuming that $t' \in fired \cup \{t_0\}$, let us show $\boxed{\sigma'(id_{t'})(''f'') = \mathtt{true}.}$
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 $\texttt{comp}(id_{t'}, "transition", gm_{t'}, ipm_{t'}, opm_{t'}) \in d.cs$:
  $$\sigma(id_{t'})("f") = \sigma(id_{t'})("sfa") \,.\, \sigma(id_{t'})("spc") \tag{1.110}$$
  Rewriting the goal with (1.110): $\boxed{\sigma(id_{t'})("sfa") \,.\, \sigma(id_{t'})("spc") = \texttt{true}}$.
  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.

- $IsTopPrioritySet(T_s, \{t_0\} \cup tp_0)$
- $ElectFired(s', fired, tp_0, fired')$
- $IsFiredSetAux(s', fired', T_s \setminus \{t_0\} \cup tp_0, fset)$
- $\neg(t_0 \in Firable(s') \wedge t_0 \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i)))$
- 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. $\gamma(t') = id_{t'}$,
$(t' \in fired \Rightarrow \sigma'(id_{t'})("f") = \mathtt{true}) \wedge (\sigma'(id_{t'})("f") = \mathtt{true} \Rightarrow t' \in fired \vee t' \in T_s \setminus \{t_0\})$

Let us prove these three points:

(a) $\boxed{IsTopPrioritySet(T_s \setminus \{t_0\}, tp_0)}$

Not provable right now.

(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. $\gamma(t') = id_{t'}$,
$(t' \in fired \Rightarrow \sigma'(id_{t'})("f") = \mathtt{true}) \wedge (\sigma'(id_{t'})("f") = \mathtt{true} \Rightarrow t' \in fired \vee t' \in T_s \setminus \{t_0\})$
Given a $t' \in T$ and an $id_{t'} \in Comps(\Delta)$ s.t. $\gamma(t') = id_{t'}$, let us show

$(t' \in fired \Rightarrow \sigma'(id_{t'})("f") = \mathtt{true}) \wedge (\sigma'(id_{t'})("f") = \mathtt{true} \Rightarrow t' \in fired \vee t' \in T_s \setminus \{t_0\})$

The proof is in two parts:

i. Assuming that $t' \in fired$, let us show $\boxed{\sigma'(id_{t'})("f") = \mathtt{true}.}$
From $t' \in fired$ and EH, $\sigma'(id_{t'})("f") = \mathtt{true}$.

ii. Assuming that $\sigma'(id_{t'})("f") = \mathtt{true}$, let us show $\boxed{t' \in fired \vee t' \in T_s \setminus \{t_0\}.}$
Thanks to $\sigma'(id_{t'})("f") = \mathtt{true}$ 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 $t' \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i))$ based on $\sigma'(id_{t'})("f") = \mathtt{true}$.
By definition of $id_{t'}$, there exist a $gm_{t'}$, $ipm_{t'}$, $opm_{t'}$ s.t. $\mathtt{comp}(id_{t'}, "transition", gm_{t'}, ipm_{t'}, opm_{t'}) \in d.cs$.
By property of the stabilize relation and $\mathtt{comp}(id_{t'}, "transition", gm_{t'}, ipm_{t'}, opm_{t'}) \in d.cs$:

$$\sigma(id_{t'})("f") = \sigma(id_{t'})("sfa") \;.\; \sigma(id_{t'})("spc") = \mathtt{true} \tag{1.111}$$

From $\sigma(id_{t'})(''sfa'') = \texttt{true}$, and appealing to Lemma Falling Edge Equal Firable, we can deduce $t' \in Firable(s')$.
From $\sigma(id_{t'})(''spc'') = \texttt{true}$, and appealing to Lemma Stabilize Compute Priority Combination After Falling Edge, we can deduce $t' \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i))$.

Then, as $t' = t_0$, $\neg(t_0 \in Firable(s') \wedge t_0 \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i)))$ is a contradiction.

$\square$

Lemma 19 (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. 8, 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.* $\gamma(t') = id_{t'}$, $(t' \in fired \Rightarrow \sigma'(id_{t'})(''f'') = \texttt{true}) \wedge (\sigma'(id_{t'})(''f'') = \texttt{true} \Rightarrow t' \in fired \vee t' \in T_s)$.
- $t \in Firable(s')$

then $t \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i)) \Leftrightarrow \sigma'(id_t)(''spc'') = \texttt{true}$

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

$\boxed{t \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i)) \Leftrightarrow \sigma'(id_t)(''spc'') = \texttt{true}.}$

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 stabilize relation and $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$:

$$\sigma'(id_t)(''spc'') = \prod_{i=0}^{\Delta(id_t)(''ian'')-1} \sigma'(id_t)(''pauths'')[i] \tag{1.112}$$

Rewriting the goal with (1.112):

$\boxed{t \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i)) \Leftrightarrow \prod_{i=0}^{\Delta(id_t)(''ian'')-1} \sigma'(id_t)(''pauths'')[i] = \texttt{true}.}$

Then, the proof is in two parts:

1. $t \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i)) \Rightarrow \prod_{i=0}^{\Delta(id_t)(''ian'')-1} \sigma'(id_t)(''pauths'')[i] = \texttt{true}$

2. $\prod_{i=0}^{\Delta(id_t)(''ian'')-1} \sigma'(id_t)(''pauths'')[i] = \texttt{true} \Rightarrow t \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i))$

Let us prove both sides of the equivalence:

1. Assuming $t \in Sens(s'.M - \sum_{t_i \in fired} pre(t_i))$, 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, $<\mathtt{input\_arcs\_number} \Rightarrow 1> \in gm_t$ and $<\mathtt{priority\_authorizations(0)} \Rightarrow \mathtt{true}> \in ipm_t$.
     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, $<\mathtt{input\_arcs\_number} \Rightarrow |input(t)|> \in gm_t$.
     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.
     $<\mathtt{input\_arcs\_valid(i)} \Rightarrow \mathtt{id_{ji}}> \in ipm_t$ and $<\mathtt{output\_arcs\_valid(j)} \Rightarrow \mathtt{id_{ji}}> \in opm_t$.
     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, $<\mathtt{priority\_authorizations(i)} \Rightarrow \mathtt{true}> \in ipm_t$, 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, $<\mathtt{priority\_authorizations(i)} \Rightarrow \mathtt{true}> \in ipm_t$, 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.
$<\texttt{priority\_authorizations(i)} \Rightarrow \texttt{id}'_{\texttt{ji}}> \in ipm_t$ and
$<\texttt{priority\_authorizations(j)} \Rightarrow \texttt{id}'_{\texttt{ji}}> \in opm_p$.
By property of the stabilize relation, $\texttt{comp}(id_t, ''transition'', gm_t, ipm_t, opm_t) \in d.cs$ and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_t)(''pauths'')[i] = \sigma'(id'_{ji}) = \sigma'(id_p)(''pauths'')[j] \tag{1.113}$$

Rewriting the goal with (1.113): $\boxed{\sigma'(id_p)(''pauths'')[j] = \texttt{true}.}$
By property of the stabilize relation and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_p)(''pauths'')[j] = (\sigma'(id_p)(''sm'') \geq \texttt{rsum} + \sigma'(id_p)(''oaw'')[j]) \tag{1.114}$$

Let us define the $\texttt{rsum}$ term as follows:

$$\texttt{rsum} = \sum_{i=0}^{j-1} \begin{cases} \sigma'(id_p)(''oaw'')[i] \ \texttt{if} \ \sigma'(id_p)(''otf'')[i]. \\ \qquad\qquad\qquad\qquad \sigma'(id_p)(''oat'')[i] = \texttt{basic} \\ 0 \ otherwise \end{cases} \tag{1.115}$$

Rewriting the goal with (1.114): $\boxed{\sigma'(id_p)(''sm'') \geq \texttt{rsum} + \sigma'(id_p)(''oaw'')[j]}$
By definition of $t \in Sens(s'.M - \sum\limits_{t_i \in fired} pre(t_i))$, we have $s'.M(p) \geq \sum\limits_{t_i \in fired} pre(p, t_i) + \omega$.
Then, there are three points to prove:

(a) $\boxed{s'.M(p) = \sigma'(id_p)(''sm'')}$

(b) $\boxed{\omega = \sigma'(id_p)(''oaw'')[j]}$

(c) $\boxed{\sum\limits_{t_i \in fired} pre(p, t_i) = \texttt{rsum}}$

Let us prove these three points:

(a) $\boxed{s'.M(p) = \sigma'(id_p)(''sm'')}$

Appealing to Lemma Falling Edge Equal Marking: $s'.M(p) = \sigma'(id_p)(''sm'')$.

(b) $\boxed{\omega = \sigma'(id_p)(''oaw'')[j]}$

By construction, and as $pre(p, t) = (\omega, \texttt{basic})$, we have
$<\texttt{output\_arcs\_weights(j)} \Rightarrow \omega> \in ipm_p$.
By property of the stabilize relation and $\texttt{comp}(id_p, ''place'', gm_p, ipm_p, opm_p) \in d.cs$:
$\omega = \sigma'(id_p)(''oaw'')[j]$.

(c) $\boxed{\sum\limits_{t_i \in fired} pre(p, t_i) = \texttt{rsum}}$

Let us replace the left and right term of the equality by their full definition:

$$\sum_{t_i \in fired} \begin{cases} \omega \text{ if } pre(p, t_i) = (\omega, \texttt{basic}) \\ 0 \textit{ otherwise} \end{cases}$$
$$=$$
$$\sum_{i=0}^{j-1} \begin{cases} \sigma'(id_p)(''oaw'')[i] \texttt{ if } \sigma'(id_p)(''otf'')[i]. \\ \qquad\qquad \sigma'(id_p)(''oat'')[i] = \texttt{basic} \\ 0 \textit{ otherwise} \end{cases}$$

2. Assume $\forall i \in [0, \sigma'_t(''input\_arcs\_number'') - 1]$, $\sigma'_t(''pauths'')(i) = \texttt{true}$, show $t \in Sens(s'.M - \sum_{t_i \in Pr(t, fired)} pre(t_i))$.

   Then, unfold the definition of the *Sens* relation.
   $\forall p \in P, \omega \in \mathbb{N}^*$,
   $\big(pre(p,t) = (\omega, \texttt{basic}) \vee pre(p,t) = (\omega, \texttt{test}) \Rightarrow$
   $s'.M(p) - \sum_{t_i \in Pr(t, fired)} pre(p, t_i) \geq \omega\big)$
   $\wedge \big(pre(p,t) = (\omega, \texttt{inhib})\big) \Rightarrow s'.M(p) - \sum_{t_i \in Pr(t, fired)} pre(p, t_i) < \omega\big)$

   Then, treat the 3 different cases.

   (a) Assume $pre(p,t) = (\omega, \texttt{test})$,
   show $s'.M(p) - \sum_{t_i \in Pr(t, fired)} pre(p, t_i) \geq \omega$.
   Then, by 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 $s'.M(p) \geq \omega$.
   That we can prove because we know $t \in Firable(s')$, thus, $t \in Sens(s'.M)$, thus, $s'.M(p) \geq \omega$.

   (b) Assume $pre(p,t) = (\omega, \texttt{inhib})$,
   show $s'.M(p) - \sum_{t_i \in Pr(t, fired)} pre(p, t_i) < \omega$.
   Use the same strategy as above.

   (c) Assume $pre(p,t) = (\omega, \texttt{basic})$,
   show $s'.M(p) - \sum_{t_i \in Pr(t, fired)} pre(p, t_i) \geq \omega$.
   Then, there are 2 CASES.

   i. CASE For all pair of transitions in $output_c(p)$, all conflicts are solved by mutual exclusion.
   Then, assuming that the priority relation is well-defined, it must not be defined over the set $output_c(t)$, and we know that $t \in output_c(p)$ since $pre(p,t) = (\omega, \texttt{basic})$.
   Then, there exists no transition t_i connected to p by a `basic` arc that verifies $t_i \succ t$.
   Then, we can deduce $\sum_{t_i \in Pr(t, fired)} pre(p, t_i) = 0$.

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 $output_c(p)$.
Assuming $pre(p,t) = (\omega, \texttt{basic})$, then, by construction, there exist:
   - a Place component id_p implementing place p
   - two indexes $i \in [0, \sigma'_t(\text{"}input\_arcs\_number\text{"}) - 1]$ and $j \in [0, \sigma'_p(\text{"}output\_arcs\_number\text{"}) - 1]$
   - a signal sig connecting $\mathtt{id_p.pauths(j)}$ to $\mathtt{id_t.pauths(i)}$

Then, we can deduce that $\sigma'_t(\text{"}pauths\text{"})(i) = \sigma'(\text{"}sig\text{"}) = \sigma'_p(\text{"}pauths\text{"})(j)$.
Then, by specializing $\forall i \in [0, \sigma'_t(\text{"}input\_arcs\_number\text{"}) - 1]$, $\sigma'_t(\text{"}pauths\text{"})(i) = \texttt{true}$ with i, we can deduce $\sigma'_t(\text{"}pauths\text{"})(i) = \sigma'(\text{"}sig\text{"}) = \sigma'_p(\text{"}pauths\text{"})(j) = true$.
Then, we have all the premises necessary to apply Lemma Stabilize Compute Individual Priority Authorization After Falling Edge, and thus to solve the goal.

□

Lemma 20 (Stabilize Compute Individual Priority Authorization 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. 8, and*
$\forall t, id_t, \sigma'_t$, *s.t.* $\gamma(t) = id_t$ *and* $\sigma'(id_t) = \sigma'_t$,
$\forall p, id_p, \sigma'_p$, *s.t.* $\gamma(p) = id_p$ *and* $\sigma'(id_p) = \sigma'_p$,
$\forall fired, fired', T_s, tp, fset, sig \in Sigs(\Delta), i, j \in \mathbb{N}, \omega \in \mathbb{N}$, *assume that:*

- $IsTopPrioritySet(T_s, \emptyset, \emptyset, \{t\} \cup tp)$
- $ElectFired(s', fired, tp, fired')$
- $FiredAux(s', fired', T_s \setminus \{t\} \cup tp, fset)$
- *EH (Extra. Hypothesis):*
  $\forall t' \in T, id_{t'}$,
  $(t' \in fired \Rightarrow \sigma'_{t'}(\text{"}fired\text{"}) = \texttt{true}) \wedge (\sigma'_{t'}(\text{"}fired\text{"}) = \texttt{true} \Rightarrow t' \in fired \vee t' \in T_s)$.
- $\mathtt{id_p.pauths(j)} \Rightarrow \mathtt{sig} \Rightarrow \mathtt{id_t.pauths(i)}$
- $pre(p,t) = (\omega, \texttt{basic})$

then $\sigma'_p(\text{"}pauths\text{"})(j) = \texttt{true} \Leftrightarrow s'.M(p) - \sum_{t_i \in Pr(t, fired)} pre(p, t_i) \geq \omega$.

Proof. From the behavior of the VHDL Place component, we can deduce:
$\sigma'_p(\text{"}pauths\text{"})(j) = true \Leftrightarrow \sigma'_p(\text{"}s\_marking\text{"}) - \sum_{k \in HPF(\sigma'_p, j)} \sigma'_p(\text{"}out\_arc\_w\text{"})(k) \geq \sigma'_p(\text{"}out\_arc\_w\text{"})(j)$ where $k \in HPF(\sigma'_p, j) \equiv k \in [0, j-1] \wedge \sigma'_p(\text{"}out\_arc\_t\text{"})(k) = \texttt{basic} \wedge \sigma'_p(\text{"}out\_t\_fired\text{"})(k) = \texttt{true}$
Then, the new goal is:
$\sigma'_p(\text{"}s\_marking\text{"}) - \sum_{k \in HPF(\sigma'_p, j)} \sigma'_p(\text{"}out\_arc\_w\text{"})(k) \geq \sigma'_p(\text{"}out\_arc\_w\text{"})(j) \Leftrightarrow s'.M(p) - \sum_{t_i \in Pr(t, fired)} pre(p, t_i) \geq \omega$.
Proof by reflexivity. 3 subgoals.

1. Show $s'.M(p) = \sigma'_p(\text{"}s\_marking\text{"})$.
   From $\gamma \vdash s \sim \sigma$, we know $s.M(p) = \sigma_p(\text{"}s\_marking\text{"})$.

From $E_c, \tau \vdash sitpn, s \xrightarrow{\downarrow} s'$, we know $s.M(p) = s'.M(p)$.

By reasoning on the VHDL falling and stabilize relations, and on the Place component behavior, we know that the "s_marking" is idle from state σ_p to state σ'_p; thus, $\sigma_p(''s\_marking'') = \sigma'_p(''s\_marking'')$.

Then, the goal is trivially proved by using the rewriting rules.

2. Show $\omega = \sigma'_p(''out\_arc\_w'')(j)$.

   We know that $pre(p,t) = (\omega, \texttt{basic})$ and $\texttt{id}_\texttt{p}.\texttt{pauths}(\texttt{j}) \Rightarrow \texttt{sig} \Rightarrow \texttt{id}_\texttt{t}.\texttt{pauths}(\texttt{i})$.

   Then, by construction, $\texttt{id}_\texttt{p}.\texttt{output\_arcs\_weights}(\texttt{j})$ is connected to the constant ω in the input map of Place component id_p.

   Then, the goal is trivially solved by showing that ports that are mapped to constant are idle during the simulation of a VHDL design.

3. Show $\sum\limits_{t_i \in Pr(t,fired)} pre(p,t_i) = \sum\limits_{k \in HPF(\sigma'_p,j)} \sigma'_p(''out\_arc\_w'')(k)$.

   We can show $\sum\limits_{t_i \in Pr(t,fired)} pre(p,t_i) = \sum\limits_{t_i \in Pr(p,t,fired)} pre(p,t_i)$
   where $t_i \in Pr(p,t,fired) \equiv t_i \succ t \wedge t_i \in fired \wedge \exists \omega \in \mathbb{N},\ s.t.,\ pre(p,t_i) = (\omega, \texttt{basic})$.

   Then, we can show that the sets $Pr(p,t,fired)$ and $HPF(\sigma'_p,j)$ are in bijection, and that for each $t_i \in Pr(p,t,fired)$ mapped to a $k \in HPF(\sigma'_p,j)$, we have $pre(p,t_i) = \sigma'_p(''out\_arc\_w'')(k)$.

   2 subgoals to solve.

   (a) $\forall t_i \in Pr(p,t,fired), \exists k \in HPF(\sigma'_p,j)\ s.t.\ pre(p,t_i) = \sigma'_p(''out\_arc\_w'')(k)$.
   Given a transition $t_i \in Pr(p,t,fired)$, show $\exists k \in HPF(\sigma'_p,j)\ s.t.\ pre(p,t_i) = \sigma'_p(''out\_arc\_w'')(k)$.
   Unfold the definition of $t_i \in Pr(p,t,fired)$:

   - $\exists \omega \in \mathbb{N}\ s.t.\ pre(p,t_i) = (\omega, \texttt{basic})$.
     Let us call ω' the element of $\mathbb{N}^*$ verifying $pre(p,t_i) = (\omega', \texttt{basic})$.
     Then, by construction, there exists a Transition component id_{t_i} implementing transition t_i and an index $n \in \mathbb{N}^*$ such that $\texttt{id}_\texttt{p}.\ \texttt{output\_arcs\_weights}(\texttt{n})$ is connected to ω' and $\texttt{output\_arcs\_types}(\texttt{n})$ is connected to $\texttt{basic}$.
     Then, by reasoning on the VHDL falling and stabilize relation, we can show that $\sigma'_p(''output\_arcs\_weight$
     ω'.
   - $t_i \succ t$.
     By construction, there exists an index $m \in \mathbb{N}^*$ and a signal $sig' \in \texttt{Declared}(\Delta)$ such that $\texttt{id}_\texttt{p}.\texttt{pauths}(\texttt{n}) \Rightarrow \texttt{sig}' \Rightarrow \texttt{id}_{\texttt{t}_\texttt{i}}.\texttt{pauths}(\texttt{m})$
     Then, by construction, and since $t_i \succ t$, we know that $n < j$. Then, $n \in [0, j-1]$.
   - $t_i \in fired$.
     Thanks to the EH, we know that $\sigma'_{t_i}(''fired'') = \texttt{true}$.
     By construction, there exists a signal $sig'' \in \texttt{Declared}(\Delta)$ such that $\texttt{id}_{\texttt{t}_\texttt{i}}.\texttt{fired} \Rightarrow \texttt{sig}'' \Rightarrow \texttt{id}_\texttt{p}.\texttt{output\_tr}$
     Then, by reasoning on the VHDL stabilize relation, we can deduce $\sigma'_p(''output\_transitions\_fired'')(n) =$
     $\sigma'_{t_i}(''fired'') = \texttt{true}$.

   Then, we have $n \in HPF(\sigma'_p,j)$ and $pre(p,t_i) = \sigma'_p(''output\_arcs\_weights'')(n)$.
   Thus, let us take n to prove the goal by assumption.

(b) $\forall k \in HPF(\sigma'_p, j), \exists t_i \in Pr(p, t, fired)\ s.t.\ pre(p, t_i) = \sigma'_p(''out\_arc\_w'')(k)$.
Given an index $k \in HPF(\sigma'_p, j)$, show $\exists t_i \in Pr(p, t, fired)\ s.t.\ pre(p, t_i) = \sigma'_p(''out\_arc\_w'')(k)$.
Unfold the definition of $k \in HPF(\sigma'_p, j)$:

- $k \in [0, j-1]$.
By construction, there exists a $t_i \in T$ and an $\omega' \in \mathbb{N}^*$ such that $pre(p, t_i) = (\omega', \texttt{basic})$ and $t_i \succ t$ and $\texttt{id}_\texttt{p}.\texttt{output\_arcs\_weights}(\texttt{k}) \Rightarrow \texttt{!}'$ and $\texttt{id}_\texttt{p}.\texttt{output\_arcs\_types}(\texttt{k}) \Rightarrow \texttt{basic}$.
- $\sigma'_p(''output\_transitions\_fired'')(k) = \texttt{true}$.
By construction, there exists a Transition component id_{t_i} implementing transition t_i such that $\texttt{id}_{\texttt{t}_\texttt{i}}.\texttt{fired} \Rightarrow \texttt{id}_\texttt{p}.\texttt{output\_transitions\_fired}(\texttt{k})$.
Then, by reasoning on the VHDL falling and stabilize relations, we can deduce $\sigma'_p(''output\_transitions$
$\sigma'_{t_i}(''fired'') = \texttt{true}$.
Then, thanks to EH, we know that $t_i \in fired$ or $t_i \in T_s$.
  - CASE $t_i \in fired$. Then, take t_i to prove the goal by assumption.
  - CASE $t_i \in T_s$.
Since t is a *top-priority* transition of set T_s (given by $IsTopPrioritySet(T_s, \emptyset, \emptyset, \{t\} \cup tp)$, then there exists no transition $t' \in T_s$ such that $t' \succ t$. Since $t_i \in T_s$, then we have $t_i \not\succ t$ contradicting $t_i \succ t$.

□

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
  - ...