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

April 26, 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.

1.1 Preliminary Definitions

1.2 Behavior Preservation Theorem

1.2.1 Proof Notations

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

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

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

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

1.2.2 Behavior Preservation Theorem and Proof

Theorem 1 (Behavior Preservation)**.** *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\tau \in \mathbb{N}$, $E_c \in \mathbb{N} \to \mathcal{C} \to \mathbb{B}$, $\theta_s \in \texttt{list}(S(sitpn))$ s.t.*

- *SITPN sitpn translates into design d: $\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$*
- *SITPN sitpn yields the execution trace θ_s after τ execution cycles in environment E_c:* $E_c, \tau \vdash sitpn \xrightarrow{full} \theta_s$.

Constants and signals reference			
Full name	*Alias*	*Category*	*Type*
″input_conditions″	″ic″	input port (T)	$\mathbb{B}$
″input_conditions″	″ic″	input port (T)	$\mathbb{B}$
″reinit_time″	″rt″	input port (T)	$\mathbb{B}$
″input_arcs_valid″	″iav″	input port (T)	$\mathbb{B}$
″fired″	″f″	output port (T)	$\mathbb{B}$
″s_condition_combination″	″scc″	internal signal (T)	$\mathbb{B}$
″s_reinit_time_counter″	″srtc″	internal signal (T)	$\mathbb{B}$
″s_priority_combination″	″spc″	internal signal (T)	$\mathbb{B}$
″s_fired″	″sf″	internal signal (T)	$\mathbb{B}$
″s_firable″	″sfa″	internal signal (T)	$\mathbb{B}$
″s_enabled″	″se″	internal signal (T)	$\mathbb{B}$
″input_arcs_number″	″ian″	generic constant (T)	$\mathbb{N}$
″transition_type″	″tt″	generic constant (T)	{NOT_TEMP, TEMP_A_B, TEMP_A_A, TEMP_A_INF}
″conditions_number″	″cn″	generic constant (T)	$\mathbb{N}$
″maximal_time_counter″	″mtc″	generic constant (T)	$\mathbb{N}$
″s_marking″	″sm″	internal signal (P)	$\mathbb{N}$
″s_output_token_sum″	″sots″	internal signal (P)	$\mathbb{N}$
″s_input_token_sum″	″sits″	internal signal (P)	$\mathbb{N}$
″reinit_transition_time″	″rtt″	output port (P)	$\mathbb{B}$
″output_arcs_types″	″oat″	input port (P)	{BASIC, TEST, INHIB}
″output_arcs_weights″	″oaw″	input port (P)	$\mathbb{N}$
″output_transition_fired″	″otf″	input port (P)	$\mathbb{B}$
″input_arcs_weights″	″iaw″	input port (P)	$\mathbb{N}$
″input_transition_fired″	″itf″	input port (P)	$\mathbb{B}$

then there exists $\Delta \in ElDesign(d, \mathcal{D}_{\mathcal{H}})$ *s.t. for all* $E_p \in (\mathbb{N} \times \{\uparrow, \downarrow\}) \rightarrow Ins(\Delta) \rightarrow value$ *verifying*

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

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

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

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

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

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

Use Full Bisimulation theorem to show traces similarity.

□

Theorem 2 (Elaboration). *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$ s.t.*

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

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

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

Theorem 3 (Initialization). *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_\mathcal{H})$, $\sigma_e \in \Sigma(\Delta)$ s.t.*

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

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

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

Theorem 4 (Simulation). *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\Delta \in ElDesign(d, \mathcal{D}_\mathcal{H})$, $\sigma_e, \sigma_0 \in \Sigma(\Delta)$ s.t.*

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

then for all $E_p \in (\mathbb{N} \times \{\uparrow, \downarrow\}) \rightarrow Ins(\Delta) \rightarrow value$, $\tau \in \mathbb{N}$, there exists $\theta_\sigma \in \texttt{list}(\Sigma(\Delta))$ s.t.

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

1.2.3 Bisimulation Theorem and Proof

Theorem 5 (Full Bisimulation). *For all $sitpn \in SITPN$, $d \in design$, $\gamma \in WM(sitpn, d)$, $\tau \in \mathbb{N}$, $E_c \in \mathbb{N} \rightarrow \mathcal{C} \rightarrow \mathbb{B}$, $\theta_s \in \texttt{list}(S(sitpn))$, $\Delta \in ElDesign(d, \mathcal{D}_\mathcal{H})$, $E_p \in (\mathbb{N} \times \{\uparrow, \downarrow\}) \rightarrow Ins(\Delta) \rightarrow value$, $\theta_\sigma \in \texttt{list}(\Sigma(\Delta))$ s.t.*

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

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

- $E_c, \tau \vdash sitpn \xrightarrow{full} \theta_s$

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

then $\theta_s \sim \theta_\sigma$

Proof. Case analysis on τ (2 CASES).

- **CASE** $\tau = 0$. By definition of the SITPN full execution and the $\mathcal{H}$-VHDL full simulation relations:

  - $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$
  - $\Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$
  - $\theta_s = [s_0]$ and $\theta_\sigma = [\sigma_0]$

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

- **CASE** $\tau > 0$. By definition of the SITPN full execution and the $\mathcal{H}$-VHDL full execution relations:

  - $E_c, \tau \vdash s_0 \xrightarrow{\uparrow_0} s_0$
  - $E_c, \tau \vdash s_0 \xrightarrow{\downarrow} s$
  - $E_c, \tau - 1 \vdash sitpn, s \rightarrow \theta_s$
  - $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$
  - $\Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$
  - $E_p, \Delta, \tau, \sigma_0 \vdash \mathrm{d.cs} \rightarrow \theta$

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

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

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

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

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

  3 subgoals (by def. of **??**).

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

□

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

- *$\lfloor sitpn \rfloor_{\mathcal{H}} = (d, \gamma)$ and $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash d \xrightarrow{elab} (\Delta, \sigma_e)$ and $\gamma \vdash E_p \overset{env}{=} E_c$*
- *σ_0 is the initial state of Δ: $\Delta, \sigma_e \vdash d.cs \xrightarrow{init} \sigma_0$*

- *First execution cycle for d:* $E_p, \Delta, \tau, \sigma_0 \vdash d.cs \xrightarrow{\uparrow,\downarrow} \sigma$
- *Particular first execution cycle for sitpn (first rising edge is idle):*

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

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

Proof. Let's show that the first execution cycle leads to two states verifying the **??** relation: $\boxed{\gamma \vdash s \overset{\downarrow}{\sim} \sigma.}$

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

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

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

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

Then, we can apply the **??** lemma to solve $\boxed{\gamma, E_c, \tau \vdash s_0 \overset{\uparrow}{\sim} \sigma'.}$

□

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

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

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

Proof. Induction on τ.

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

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

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

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

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

2 subgoals (by def. of **??**).

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

□

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

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

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

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

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

Solved by applying **??** and then "Falling Edge" lemmas. □

1.3 Initial States

1.4 First Rising Edge

1.5 Rising Edge

1.6 Falling Edge

Definition 1 (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* $\mathcal{D}_{\mathcal{H}}, \emptyset \vdash \mathrm{d} \xrightarrow{elab} \Delta, \sigma_e$
- $\gamma, E_c, \tau \vdash s \overset{\uparrow}{\sim} \sigma$

- $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$
- $\texttt{Inject}_{\downarrow}(\sigma, E_p, \tau, \sigma_i)$ *and* $\Delta, \sigma_i \vdash \mathrm{d.cs} \xrightarrow{\downarrow} \sigma_{\downarrow}$ *and* $\Delta, \sigma_{\downarrow} \vdash \mathrm{d.cs} \xrightarrow{\rightsquigarrow} \sigma'$
- *State* σ *is a stable design state:* $\mathcal{D}_{\mathcal{H}}, \Delta, \sigma \vdash \mathrm{d.cs} \xrightarrow{comb} \sigma$

Lemma 4 (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. 1, then* $\gamma \vdash s' \overset{\downarrow}{\sim} \sigma'$.

Proof. By definition of **??**, 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 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 5 (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. 1, 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 $E_c, \tau \vdash sitpn, s \xrightarrow{\downarrow} s'$:

$$s.M(p) = s'.M(p) \tag{1.1}$$

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

$$\sigma'(id_p)(''s\_marking'') = \sigma(id_p)(''s\_marking'') \tag{1.2}$$

Rewriting the goal with (1.1) and (1.2): $\boxed{s.M(p) = \sigma(id_p)(''s\_marking'').}$

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

□

Lemma 6 (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. 1, then $\forall p, id_p$ s.t. $\gamma(p) = id_p$, $\sum\limits_{t \in Fired(s')} pre(p,t) = \sigma'(id_p)(''s\_output\_token\_sum'')$.*

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

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

$$\sigma'(id_p)(''sots'') = \sum_{i=0}^{\Delta(id_p)(''oan'')-1} \begin{cases} \sigma'(id_p)(''oaw'')[i] \text{ }\mathtt{if}\text{ } (\sigma'(id_p)(''otf'')[i] \\ \qquad\qquad . \sigma'(id_p)(''oat'')[i] = \mathtt{BASIC}) \\ 0 \; otherwise \end{cases} \tag{1.3}$$

Rewriting the goal with (1.3):

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

Let us unfold the definition of the left sum term:

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

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

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

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

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

   $$\Delta(id_p)(''oan'') = 1 \tag{1.4}$$

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

   $$\begin{aligned} \sigma'(id_p)(''oat'')[0] &= \texttt{BASIC} && (1.5) \\ \sigma'(id_p)(''otf'')[0] &= \texttt{true} && (1.6) \\ \sigma'(id_p)(''oaw'')[0] &= 0 && (1.7) \end{aligned}$$

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

   $$\sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } pre(p,t) = (\omega, \texttt{basic}) \\ 0\ otherwise \end{cases} = 0 \tag{1.8}$$

   Rewriting the goal with (1.4), (1.5), (1.6), (1.7) and (1.8), tautology.

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

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

   $$\Delta(id_p)(''oan'') = |output(p)| \tag{1.9}$$

Rewriting the goal with (1.9): $\boxed{\sum\limits_{t \in Fired(s')} f(t) = \sum\limits_{i=0}^{|output(p)|-1} g(i).}$

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

- **BASE CASE**:

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

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

- **INDUCTION CASE**:

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

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

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

  By definition of g:

  $$g(0) = \begin{cases} \sigma'(id_p)("oaw")[0] \texttt{ if } (\sigma'(id_p)("otf")[0] \\ \qquad\qquad\qquad\qquad . \sigma'(id_p)("oat")[0] = \texttt{BASIC}) \\ 0 \; otherwise \end{cases} \tag{1.10}$$

  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: $\sum\limits_{t \in Fired(s')} f(t) = \sum\limits_{i=1}^{|output(p)|-1} g(i).$

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

$$pre(p,t) = (\omega, \texttt{basic}) \tag{1.11}$$

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

$$\sigma'(id_t)("fired") = \texttt{true} \tag{1.12}$$

Appealing to Lemma **??**, we know $t \in Fired(s')$.
As $t \in Fired(s')$, we can rewrite the left sum term of the goal as follows:
$\boxed{f(t) + \sum\limits_{t' \in Fired(s') \setminus \{t\}} f(t') = g(0) + \sum\limits_{i=1}^{|output(p)|-1} g(i)}$
We know that $g(0) = \sigma'(id_p)("oaw")[0]$, and by property of the stabilize relation and $<\texttt{output\_arcs\_weights}(0) \Rightarrow \omega> \in ipm_p$:

$$\sigma'(id_p)("oaw")[0] = \omega \tag{1.13}$$

Rewriting the goal with (1.13):
$\boxed{f(t) + \sum\limits_{t' \in Fired(s') \setminus \{t\}} f(t') = \omega + \sum\limits_{i=1}^{|output(p)|-1} g(i)}$
By definition of f, and as $pre(p,t) = (\omega, \texttt{basic})$, then $f(t) = \omega$; thus, rewriting the goal:
$\boxed{\omega + \sum\limits_{t' \in Fired(s') \setminus \{t\}} f(t') = \omega + \sum\limits_{i=1}^{|output(p)|-1} g(i)}$
Then, knowing that $g(0) = \omega$, we can apply the induction hypothesis with $F = Fired(s') \setminus \{t\}$: $g(0) + \sum\limits_{t' \in Fired(s') \setminus \{t\}} f(t') = g(0) + \sum\limits_{i=1}^{|output(p)|-1} g(i)$.

□

Lemma 7 (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. 1, then $\forall p, id_p$ s.t. $\gamma(p) = id_p$, $\sum\limits_{t \in Fired(s')} post(t,p) = \sigma'_p("s\_input\_token\_sum")$.*

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

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 $\texttt{comp}(id_p, "place", gm_p, ipm_p, opm_p) \in d.cs$:

$$\sigma'(id_p)("sits") = \sum_{i=0}^{\Delta(id_p)("ian")-1} \begin{cases} \sigma'(id_p)("iaw")[i] \texttt{ if } \sigma'(id_p)("itf")[i] \\ 0 \textit{ otherwise} \end{cases} \tag{1.14}$$

Rewriting the goal with (1.14):

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

Let us unfold the definition of the left sum term:

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

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

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

   By construction, <`input_arcs_number` ⇒ `1`> $\in gm_p$, <`input_transitions_fired(0)` ⇒ `true`> $\in ipm_p$, and <`input_arcs_weights(0)` ⇒ `0`> $\in ipm_p$.

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

   $$\Delta(id_p)("ian") = 1 \tag{1.15}$$

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

   $$\sigma'(id_p)("itf")[0] = \texttt{true} \tag{1.16}$$
   $$\sigma'(id_p)("iaw")[0] = 0 \tag{1.17}$$

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

   $$\sum_{t \in Fired(s')} \begin{cases} \omega \texttt{ if } post(t,p) = \omega \\ 0 \; otherwise \end{cases} = 0 \tag{1.18}$$

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

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

   By construction, <`input_arcs_number` ⇒ $|input(p)|$> $\in gm_p$, and by property of the elaboration relation:

   $$\Delta(id_p)("ian") = |input(p)| \tag{1.19}$$

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

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

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

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

- **BASE CASE**:
  In that case, $0 > |input(p)| - 1$ and $\sum_{i=0}^{|input(p)|-1} g(i) = 0$.
  As $0 > |input(p)| - 1$, then $|input(p)| = 0$, thus contradicting $input(p) \neq \emptyset$.
- **INDUCTION CASE**:
  In that case, $0 \leq |input(p)| - 1$.

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

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

  By definition of g:

  $$g(0) = \begin{cases} \sigma'(id_p)("iaw")[0] \texttt{ if } \sigma'(id_p)("itf")[0] \\ 0\ otherwise \end{cases} \tag{1.20}$$

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

  $$post(t,p) = \omega \tag{1.21}$$

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

$$\sigma'(id_t)("fired") = \texttt{true} \tag{1.22}$$

Appealing to Lemma **??** and (1.22), 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 $<\texttt{input\_arcs\_weights}(0) \Rightarrow \omega> \in ipm_p$:

$$\sigma'(id_p)("iaw")[0] = \omega \tag{1.23}$$

Rewriting the goal with (1.23):

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

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 8 (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. 1, 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, $\texttt{Inject}_{\downarrow}$, $\mathcal{H}$-VHDL rising edge and stabilize relations, and $\texttt{comp}(id_t, \text{"}transition\text{"}, gm_t, ipm_t, opm_t) \in d.cs$:

$$\begin{aligned}\sigma(id_t)(\text{"}se\text{"}) = \texttt{true} \wedge \Delta(id_t)(\text{"}tt\text{"}) \neq \texttt{NOT\_TEMPORAL} \wedge \sigma(id_t)(\text{"}srtc\text{"}) = \texttt{false}\\ \wedge\sigma(id_t)(\text{"}stc\text{"}) < \Delta(id_t)(\text{"}mtc\text{"}) \Rightarrow \sigma'(id_t)(\text{"}stc\text{"}) = \sigma(id_t)(\text{"}stc\text{"}) + 1\end{aligned} \tag{1.24}$$

$$\begin{aligned}\sigma(id_t)(\text{"}se\text{"}) = \texttt{true} \wedge \Delta(id_t)(\text{"}tt\text{"}) \neq \texttt{NOT\_TEMPORAL} \wedge \sigma(id_t)(\text{"}srtc\text{"}) = \texttt{false}\\ \wedge\sigma(id_t)(\text{"}stc\text{"}) \geq \Delta(id_t)(\text{"}mtc\text{"}) \Rightarrow \sigma'(id_t)(\text{"}stc\text{"}) = \sigma(id_t)(\text{"}stc\text{"})\end{aligned} \tag{1.25}$$

$$\begin{aligned}\sigma(id_t)(\text{"}se\text{"}) = \texttt{true} \wedge \Delta(id_t)(\text{"}tt\text{"}) \neq \texttt{NOT\_TEMPORAL}\\ \wedge\sigma(id_t)(\text{"}srtc\text{"}) = \texttt{true} \Rightarrow \sigma'(id_t)(\text{"}stc\text{"}) = 1\end{aligned} \tag{1.26}$$

$$\sigma(id_t)(\text{"}se\text{"}) = \texttt{false} \vee \Delta(id_t)(\text{"}tt\text{"}) = \texttt{NOT\_TEMPORAL} \Rightarrow \sigma'(id_t)(\text{"}stc\text{"}) = 0 \tag{1.27}$$

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 $\gamma, E_c, \tau \vdash s \overset{\downarrow}{\sim} \sigma$:

   $$\sigma(id_t)(\text{"}se\text{"}) = \texttt{false} \tag{1.28}$$

   Thanks to (1.27) and (1.28):

   $$\sigma'(id_t)(\text{"}stc\text{"}) = 0 \tag{1.29}$$

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

   $$s'.I(t) = 0 \tag{1.30}$$

   Rewriting the goal with (1.29) and (1.30): tautology.

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

   $$\sigma(id_t)(\text{"}se\text{"}) = \texttt{true} \tag{1.31}$$

   By construction, and as $upper(I_s(t)) = \infty$, $<\texttt{transition\_type} \Rightarrow \texttt{TEMP\_A\_INF}> \in gm_t$. By property of the elaboration relation:

   $$\Delta(id_t)(\text{"}tt\text{"}) = \texttt{TEMP\_A\_INF} \tag{1.32}$$

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

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

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

$$\sigma(id_t)(''srtc'') = \mathtt{true} \tag{1.33}$$

Thanks to (1.26), (1.31), (1.32) and (1.33):

$$\sigma'(id_t)(''stc'') = 1 \tag{1.34}$$

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

$$s'.I(t) = 1 \tag{1.35}$$

Rewriting the goal with (1.34) and (1.35): tautology.

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

$$\sigma(id_t)(''srtc'') = \mathtt{false} \tag{1.36}$$

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, $<\mathtt{maximal\_time\_counter} \Rightarrow a> \in gm_t$, and by property of the elaboration relation:

$$\Delta(id_t)(''mtc'') = a \tag{1.37}$$

By definition of $E_c, \tau \vdash s \xrightarrow{\downarrow} s'$, and knowing that $t \in Sens(s.M)$ and $s.reset_t(t) = \mathtt{false}$ and $upper(I_s(t)) = \infty$:

$$s'.I(t) = s.I(t) + 1 \tag{1.38}$$

Rewriting the goal with (1.38): $\boxed{s.I(t) + 1 = \sigma'(id_t)(''stc'').}$
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 $upper(I_s(t)) = \infty$:

$$s.I(t) = \sigma(id_t)(''stc'') \tag{1.39}$$

Thanks to (1.37), (1.39) and $s.I(t) < a$:

$$\sigma(id_t)(''stc'') < \Delta(id_t)(''mtc'') \tag{1.40}$$

Thanks to (1.24), (1.40), (1.36) and (1.31):

$$\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'') + 1 \tag{1.41}$$

Rewriting the goal with (1.41) and (1.39): 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, <`maximal_time_counter` $\Rightarrow a$> $\in gm_t$, and <`transition_type` $\Rightarrow$ `TEMP_A_INF`> $\in gm_t$ by property of the elaboration relation:

$$\begin{aligned} \Delta(id_t)("mtc") &= a && (1.42) \\ \Delta(id_t)("tt") &= \texttt{TEMP\_A\_INF} && (1.43) \end{aligned}$$

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

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

By definition of $E_c, \tau \vdash s \overset{\downarrow}{\rightarrow} s'$, 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 $t \in Sens(s.M)$:

$$\sigma(id_t)("se") = \texttt{true} \tag{1.44}$$

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

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

By definition of $E_c, \tau \vdash s \overset{\downarrow}{\rightarrow} s'$: $s'.I(t) = 1$.
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 $s.reset_t(t) = \texttt{false}$:

$$\sigma(id_t)("srtc") = \texttt{false} \tag{1.45}$$

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

$$\begin{aligned} s'.I(t) = s.I(t) + 1 &\Rightarrow s.I(t) + 1 > lower(I_s(t)) \\ &\Rightarrow s.I(t) \geq lower(I_s(t)) \end{aligned} \tag{1.46}$$

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

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

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

$$\sigma(id_t)("stc") = lower(I_s(t)) \tag{1.47}$$

Thanks to (1.25):

$$\sigma'(id_t)("stc") = \sigma(id_t)("stc") \tag{1.48}$$

Rewriting the goal with (1.47) and (1.55): tautology.

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

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

$$s.I(t) = \sigma(id_t)(''stc'') \tag{1.49}$$

Thanks to (1.25):

$$\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'') \tag{1.50}$$

Rewriting the goal with (1.57), (1.56) and $s.I(t) = lower(I_s(t))$: tautology.

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

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

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 $tt \in \{$

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

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

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

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

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

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

$$\sigma(id_t)(''se'') = \texttt{true} \tag{1.51}$$

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

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

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

Contradicts $upper(I_s(t)) > 0$.

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

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

$$\sigma(id_t)(''srtc'') = \texttt{false} \tag{1.52}$$

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

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

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

$$s'.I(t) = s.I(t) \tag{1.53}$$

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

$$\sigma(id_t)(''stc'') = upper(I_s(t)) \tag{1.54}$$

Thanks to (1.25):

$$\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'') \tag{1.55}$$

Rewriting the goal with (1.55) and (1.54): tautology.

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

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

$$s.I(t) = \sigma(id_t)(''stc'') \tag{1.56}$$

Thanks to (1.25):

$$\sigma'(id_t)(''stc'') = \sigma(id_t)(''stc'') \tag{1.57}$$

Rewriting the goal with (1.57), (1.56) and $s.I(t) = lower(I_s(t))$: tautology.

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

□

1.6.3 Falling edge and firable transitions

Lemma 9 (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. 1, and $\forall t, id_t$ s.t. $\gamma(t) = id_t$ and $\sigma'(id_t) = \sigma'_t$, then $t \in Firable(s') \Leftrightarrow \sigma'_t(''s\_firable'') = \mathtt{true}$.*

Proof. □

1.7 A detailled proof: equivalence of fired transitions

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
  - …