Corrélation de modèles de simulation du processeur EPI Rhea pour la vérification fonctionnelle

Bastien HUBERT

27 septembre 2023

European Processor Initiative

Pourquoi l'EPI ?

- *Lancée en 2019 par EuroHPC JU*
- *1/3 des ressources en super-calculateurs sont consommées en Europe*
- *1/20 des 500 super-calculateurs les plus puissants et 0% des processeurs qui les composent sont européens*
- *Besoin de souveraineté technologique de l'UE*
- *Enjeux modernes: sécurité, santé, énergies, climat, ingénierie*
- *Processeurs hautes performances et basses puissances*

European Processor Initiative

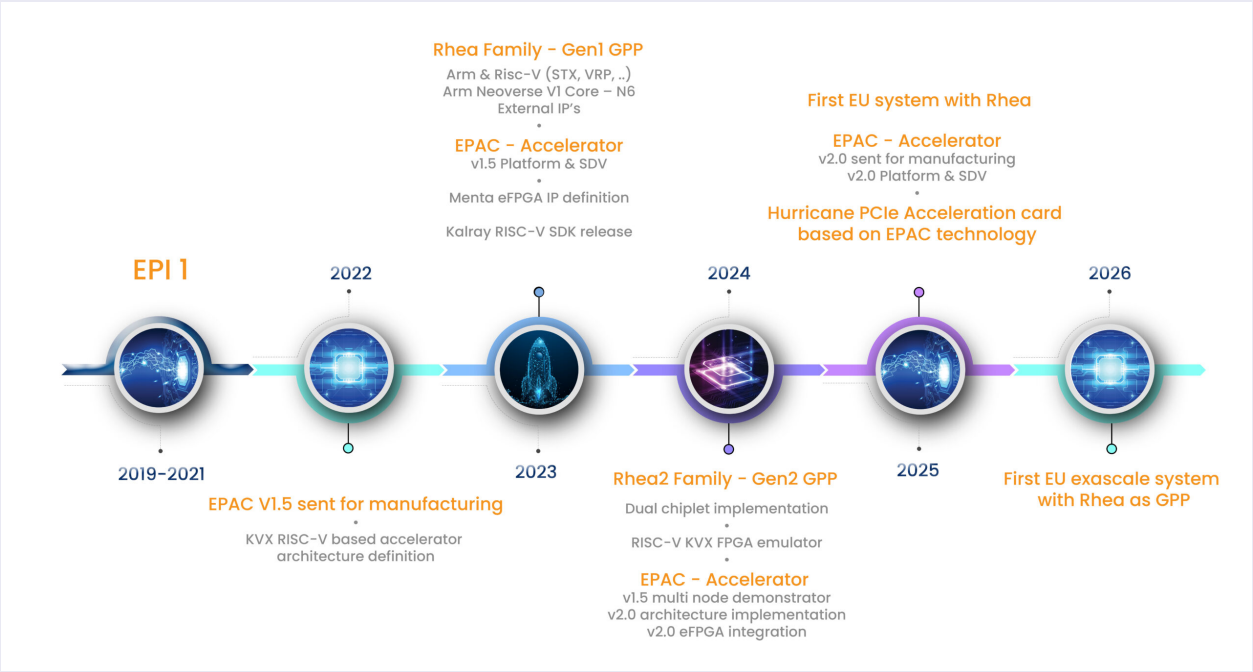
Membres de l'EPI

Sipearl

- *Chargé de concevoir le processeur européen : Rhea*

European Processor Initiative

Chronologie prévisionnelle de l'EPI



Objectif principal du stage

- *s'assurer que Rhea se comporte comme prévu par son cahier des charges*

Modèles de processeur

Machines à états finis (FSM)

$$M \stackrel{def}{=\!=} \{\Sigma,\ S,\ s_{init},\ \delta\} \tag{1}$$

Chemins d'états dans M

$$\hat{s}_0 \underset{M,\hat{\sigma}}{\rightsquigarrow} \hat{s}_1 \stackrel{def}{=\!=} \exists\,(s_0,\ s_1,\ \cdots,\ s_n) \in S^n\ :$$

$$\begin{cases} s_0 = \hat{s}_0 \\ s_n = \hat{s}_1 \\ \forall k \in [\![1, n]\!],\quad s_k = \delta(s_{k-1},\ \hat{\sigma}_{k-1}) \end{cases} \tag{2}$$

Fonction de macro-transition

$$\hat{\delta} : \begin{cases} S \times \bigcup_{n\in\mathbb{N}} \Sigma^n & \to & S \\ (\hat{s}_0,\ \hat{\sigma}) & \mapsto & \hat{s}_1 \mid \hat{s}_0 \underset{M,\hat{\sigma}}{\rightsquigarrow} \hat{s}_1 \end{cases} \tag{3}$$

Modèles de processeur

Modèles équivalents

$$M_1 \sim M_2 \stackrel{def}{=\!=} \begin{cases} S_1 & = & S_2 \\ \Sigma_1 & = & \Sigma_2 \\ \hat{\delta}_1 & = & \hat{\delta}_2 \end{cases} \quad (4)$$

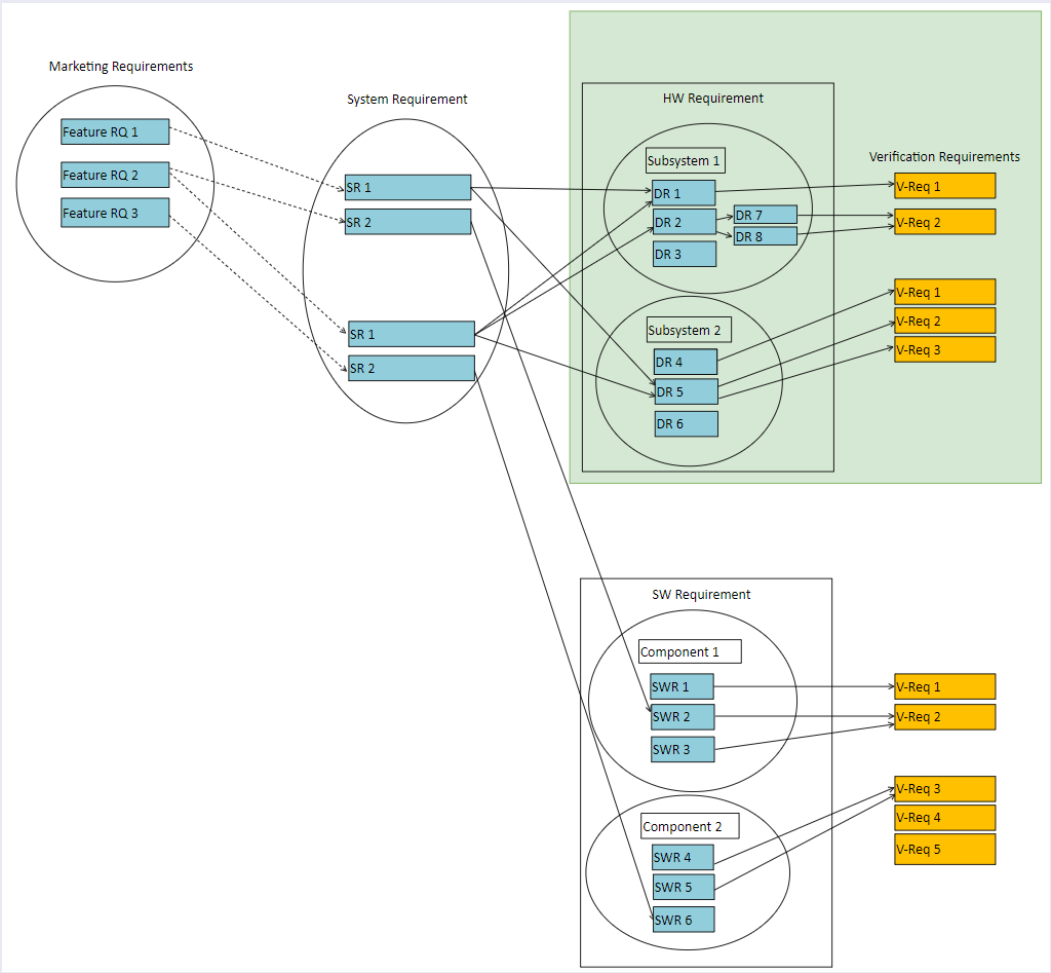
Sous-modèle

$$M_1 \preceq M_2 \stackrel{def}{=\!=} \begin{cases} S_1 & \subseteq & S_2 \\ \Sigma_1 & \subseteq & \Sigma_2 \\ \hat{\delta}_1 & = & \hat{\delta}_{2 \mid S_1 \times (\bigcup_{n \in \mathbb{N}} \Sigma^n)} \end{cases} \quad (5)$$

Hiérarchie des exigences

Structure des exigences pour Rhea

- *HAS : System Requirements*
- *MAS : Design Requirements*
- *Verification Requirements*
- *Coverage Items*



Coverage items et *SLM*

Coverage items

$$(\hat{s}_0,\ \hat{\sigma}, \hat{s}_1)\ :\ \hat{s}_0 \underset{M^*,\hat{\sigma}}{\rightsquigarrow} \hat{s}_1 \tag{6}$$

System Level Model

$$SLM \stackrel{def}{=} \{\Sigma_{SLM},\ S_{SLM},\ s_{init},\ \delta_{SLM}\}\ : \tag{7}$$

$$\delta_{SLM} : \begin{cases} S_{SLM} \times \Sigma_{SLM} & \rightarrow & S_{SLM} \\ (\hat{s}_0,\ \hat{\sigma}) & \mapsto & \hat{s}_1 \mid (\hat{s}_0,\ \hat{\sigma}, \hat{s}_1) \textit{ est un coverage item} \end{cases}$$

Vérification d'un *coverage item* par un modèle

$$M \textit{ vérifie } (\hat{s}_0,\ \hat{\sigma}, \hat{s}_1) \stackrel{def}{=} \hat{s}_0 \underset{M,\hat{\sigma}}{\rightsquigarrow} \hat{s}_1 \tag{8}$$

SEC par simulation

Méthodologie de vérification

SEC

- *vérification statique et formelle*
- *combiner 2 FSM en une machine produit*
- *les sorties de chaque état de la machine produit doivent valoir 1*

SEC par simulation

- *vérification dynamique et hybride*
- *rendre le modèle à vérifier comparable au FSM*
- *les valeurs des CSRs du modèle sont comparés à celles du SLM*

Inconvénients du SEC par simulation

- *les modèles doivent avoir les mêmes entrées, sorties et CSRs*
- *le modèle modifié doit être assez petit pour être simulé*

Plans de tests et réalisations de tests

Plan de tests

$$T \overset{def}{=} \left(\hat{\sigma}^{(k)}, \hat{s}_k\right)_{k\in[\![1,N]\!]} : \begin{cases} \forall k \in [\![1, N]\!], \hat{s}_{k-1} \underset{SLM,\hat{\sigma}^{(k-1)}}{\rightsquigarrow} \hat{s}_k \\ avec\ \hat{s}_0 = s_{init} \end{cases} \quad (9)$$

Réalisations de tests

$$T_M \overset{def}{=} \left(\hat{\sigma}^{(k)}, \tilde{s}_k\right)_{k\in[\![1,N]\!]} : \begin{cases} \forall k \in [\![1, N]\!], \tilde{s}_{k-1} \underset{SLM,\hat{\sigma}^{(k-1)}}{\rightsquigarrow} \tilde{s}_k \\ avec\ \tilde{s}_0 = s_{init} \end{cases} \quad (10)$$

Fonction de réalisation

$$R_M : \begin{cases} \{Plans\} & \rightarrow & \{M\text{-réalisations}\} \\ \left(\hat{\sigma}^{(k)}, \hat{s}_k\right)_{k\in[\![1,N]\!]} & \mapsto & \left(\hat{\sigma}^{(k)}, \tilde{s}_k\right)_{k\in[\![1,N]\!]} \end{cases} \quad (11)$$

Plans de tests et réalisations de tests

Preuve par tests

$$T \text{ contient } c \stackrel{def}{=} \exists\, k \in [\![1, N]\!] \mid c = \left(\hat{s}_{k-1}, \hat{\sigma}^{(k)}, \hat{s}_k\right) \tag{12}$$

$$T_M \text{ évalue } c \stackrel{def}{=} R_M^{-1}(T_M) \text{ contient } c \tag{13}$$

$$T_M \text{ vérifie } c \stackrel{def}{=} \exists\, k \in [\![1, N]\!] \mid c = \left(\tilde{s}_{k-1}, \hat{\sigma}^{(k)}, \tilde{s}_k\right) \tag{14}$$

$$T_M \text{ vérifie } c \implies M \text{ vérifie } c \tag{15}$$

Remarques

- *M n'est pas équivalent au SLM car le SLM ne décrit pas tous les états de M, mais c'est un (plus grand) surmodèle du SLM*
- *2 plus grands surmodèles du SLM ne sont pas équivalents, mais leurs réstrictions au SLM le sont*

Métriques de vérification

Taux de couverture

$$\mu_c(T_M) \stackrel{def}{=} \frac{\mu(\{c \text{ évalué par } T_M\})}{\mu(\{c \text{ dans le SLM}\})} \quad (16)$$

Taux de succès

$$\mu_s(T_M) \stackrel{def}{=} \frac{\mu(\{c \text{ vérifié } T_M\})}{\mu(\{c \text{ évalué par } T_M\})} \quad (17)$$

Taux de redondance

$$\mu_r(T_{M_1}, T_{M_2}) \stackrel{def}{=} \frac{\mu(\{c \text{ évalué par } T_{M_1}\} \cap \{c \text{ évalué par } T_{M_2}\})}{\mu(\{c \text{ dans le SLM}\})} \quad (18)$$

Métriques de vérification

Taille des plans de tests

Peu de tests complexes	**Beaucoup de tests simples**
Peut vérifier plus de VRs	*Peut vérifier des VRs séparément*
Globalement plus court	*Meilleur parallélisme*
Plus long par simulation	*Globalement plus gros*
Plus de branches à couvrir	*Peut rater des branches*

Indice de Jaccard

$$
\begin{aligned}
J(T_{M_1}, T_{M_2}) &= \frac{\mu(\{c \text{ evaluated by } T_{M_1}\} \cap \{c \text{ evaluated by } T_{M_2}\})}{\mu(\{c \text{ evaluated by } T_{M_1}\} \cup \{c \text{ evaluated by } T_{M_2}\})} \\
&= \frac{\mu_r(T_{M_1}, T_{M_2})}{\mu_c(T_{M_1}, T_{M_2})} \\
&= \frac{\mu_r(T_{M_1}, T_{M_2})}{\mu_c(T_{M_1}) + \mu_c(T_{M_2}) - \mu_r(T_{M_1}, T_{M_2})}
\end{aligned}
\tag{19}
$$

RTL et Prototype Virtuel

2 nouveaux modèles

Modèle RTL *(SystemVerilog)*	**Modèle VP** *(SystemC)*
Pour des mesure de performance et de vérification	*Pour le débogage et l'exploration architecturale*
Point d'entrée pour la synthèse	*Permet une synthèse HLS*
Plus précis	*Plus rapide et petit*
Définit le comportement final	*Doit être corrélé au RTL*
Monolithique et difficile à modifier	*Simple à modifier*

Cosimulations

- *Permet de combiner le meilleur des 2 modèles*
- *Permet de se concentrer sur un sous-système à vérifier*
- *Doit pouvoir établir la communication entre les modèles*
- *Ici, 4 cosimulations : CC/SYSCTRL, DDR, HBM, et PCIe*

Communication entre les langages

Universal Verification Methodology Connect

SystemC

```
#include "uvmc.h"
#include "consumer.h"

int sc_main(int argc, char *argv[]) {
  consumer cons("cons");

  uvmc.uvmc_connect(cons.in, "foo");
  sc_core.sc_start();

  return 0;
}
```

SystemVerilog

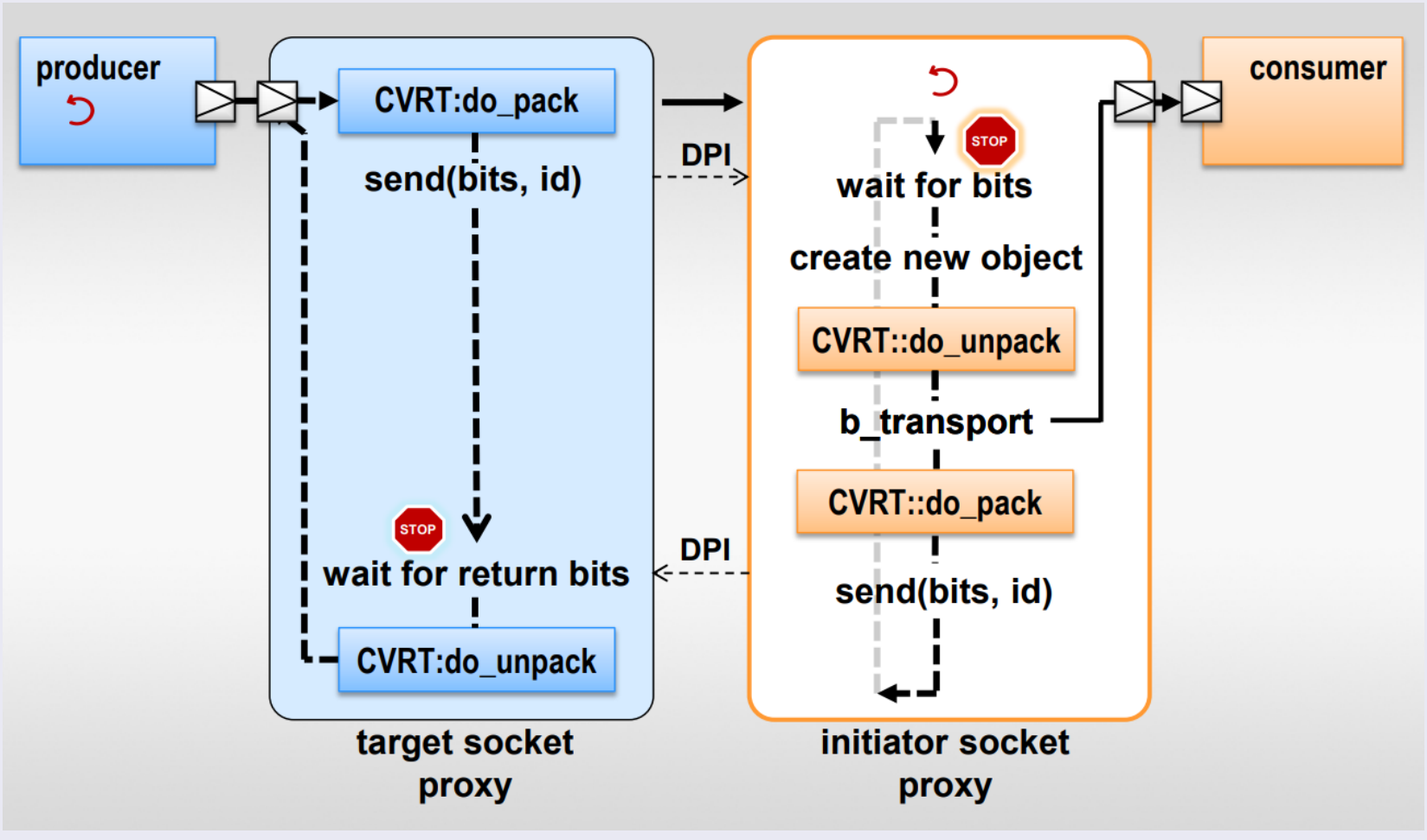
```
import uvmc_pkg::*;
`include "producer.sv"

module sv_main;
  producer prod = new("prod");
  initial begin
    uvmc_tlm #()::connect(prod.out, "foo");
    run_test();
  end

endmodule
```

Communication entre les langages

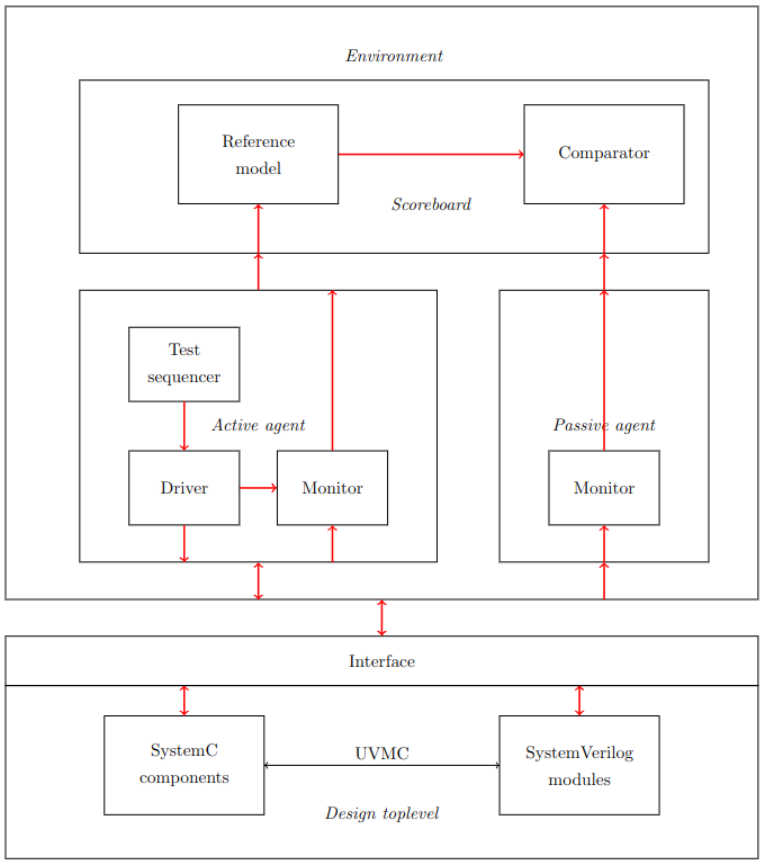
Fonctionnement d'UVMC avec TLM et DPI



Device Under Tests

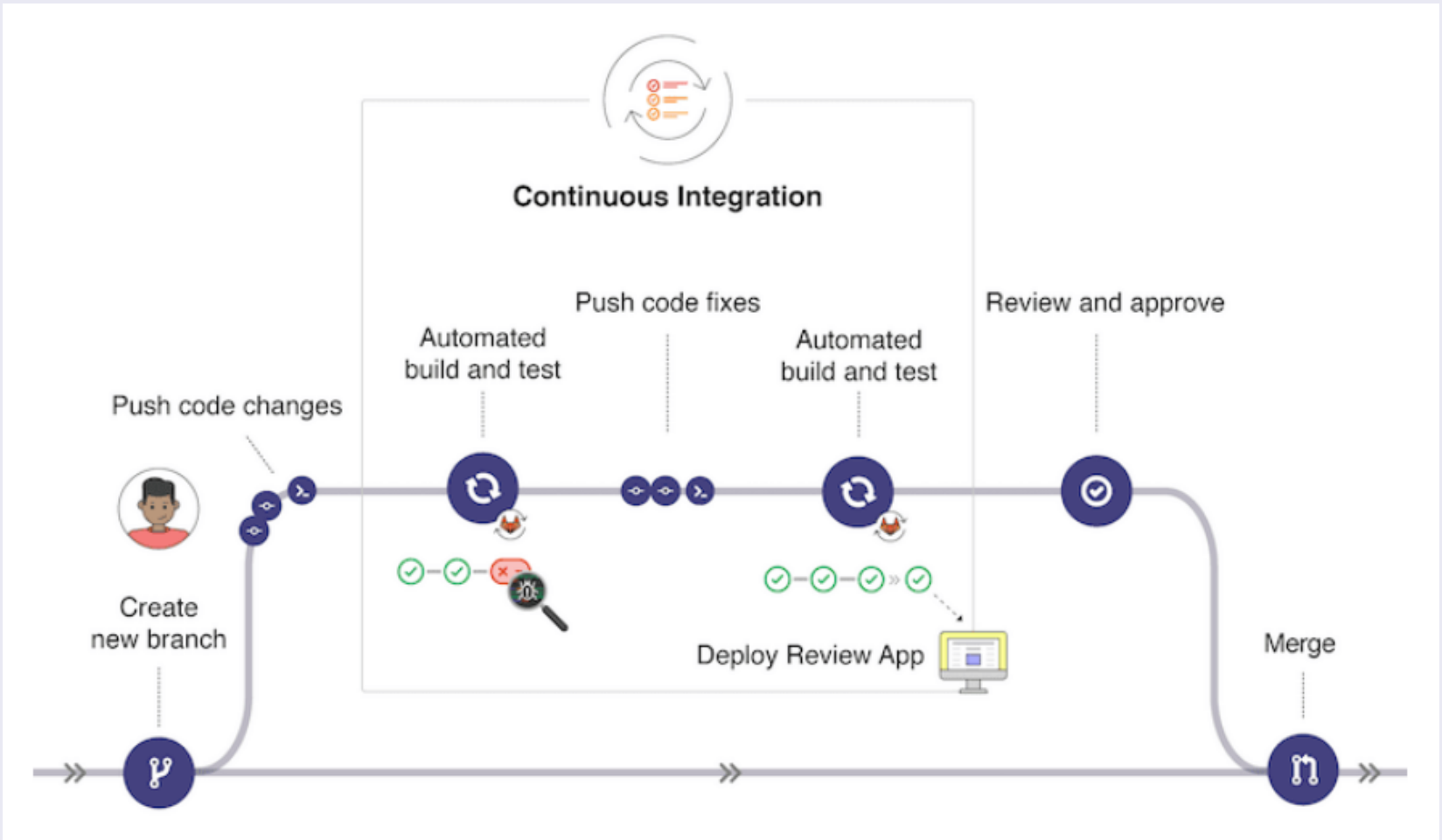
Structure d'un DUT

- *Le modèle à vérifier et son interface*
- *Des agents actifs avec leur séquenceur de tests*
- *Des agents passifs monitorer le modèle*
- *Un scoreboard pour enregistrer les résultats des tests*



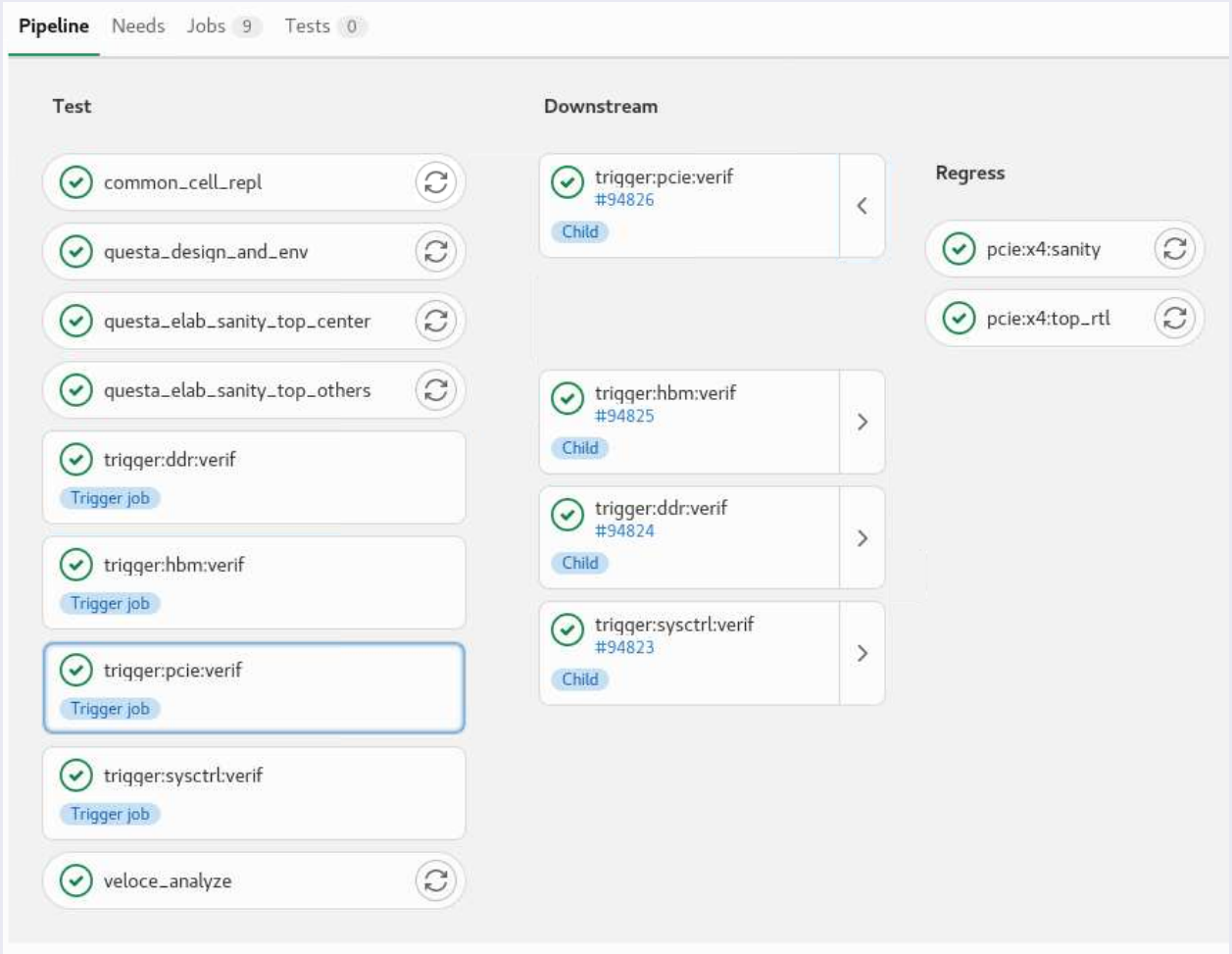
Gitlab CI

Continuous Integration



Gitlab CI

Pipeline



Hierarchy Dependency Analyser

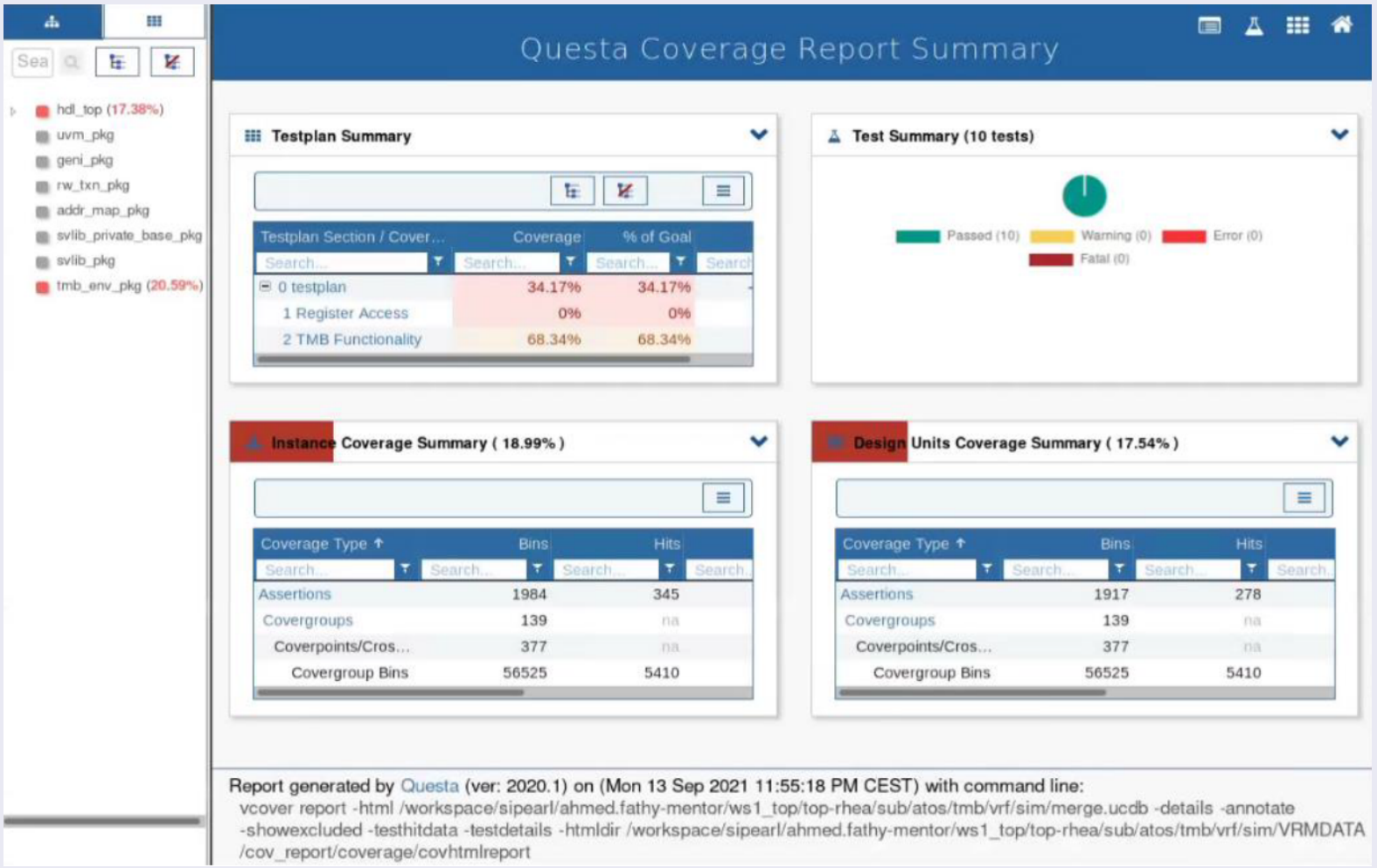
Chaîne de compilation HDA

compilation	**commande HDA**
clean	*make clean*
build tests	*make tests*
compile testbench	*make compile CMN_DIR=gen_ori*
elaborate testbench	*make elab CMN_DIR=gen_ori ELAB_BEH_MEM=1 SC=1 CDL=0 TTNDPI=0 SVA=0 DEBUGDB=1 TOP=tb_rhea_top_sysctrl_dnoc_q0_cfg BATCH=0*
run simulation	*make run_opt ELAB_BEH_MEM=1 SC=1 CDL=0 TTNDPI=0 SVA=0 DEBUGDB=1 TESTNAME=gic_spi_col TESTNAME_AP=hello_world PREBUILT_ELF=0 TIMEOUT_CYCLES=8000000 TOP=tb_rhea_top_sysctrl_dnoc_q0_cfg BATCH=0 &*

- *Lance QuestaSim avec la commande make run_opt*
- *Questa génère un rapport de couverture d'où on peut extraire la liste des coverage items évalués*

QuestaSim

Rapport de couverture



QuestaSim

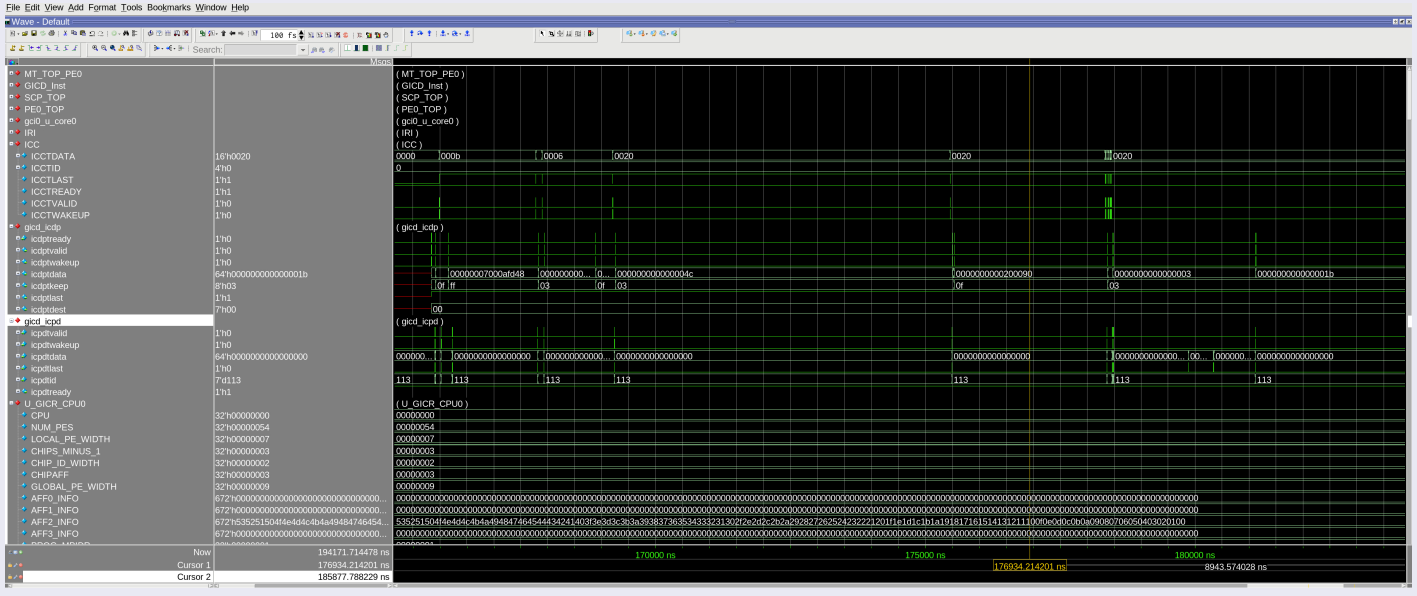
Scripts Tcl

Transcript

```
# ** Info: puts_severity_task: SCP: SYSCTRL TMB running on clk PLL.
#    Time: 100596713837 fs  Scope: tb_cdl_rhea_top.u_tb_rhea_top.u_rhea_top.top_center_q0_inst.top_center_q0_south_inst.top_sys_sh2_inst.top_sys_sh1_inst.top_sys_sh0_inst.sysctrl_inst.cp_top_inst.u_cp_top.u_axi_puts_severity_task.severity_task
File: /scratch/sipearl/bastien.hubert/sc_ws6_top/top-rhea/sub/soc/units/cp/source/sv/tb/axi_puts_severity_task.sv Line: 149
# ** Info: puts_severity_task: SCP: Chip-0 CMN700 Mesh Config
#    Time: 100866713077 fs  Scope: tb_cdl_rhea_top.u_tb_rhea_top.u_rhea_top.top_center_q0_inst.top_center_q0_south_inst.top_sys_sh2_inst.top_sys_sh1_inst.top_sys_sh0_inst.sysctrl_inst.cp_top_inst.u_cp_top.u_axi_puts_severity_task.severity_task
File: /scratch/sipearl/bastien.hubert/sc_ws6_top/top-rhea/sub/soc/units/cp/source/sv/tb/axi_puts_severity_task.sv Line: 149
# ** Info: puts_severity_task: SCP: Process HNFs...
#    Time: 100946713963 fs  Scope: tb_cdl_rhea_top.u_tb_rhea_top.u_rhea_top.top_center_q0_inst.top_center_q0_south_inst.top_sys_sh2_inst.top_sys_sh1_inst.top_sys_sh0_inst.sysctrl_inst.cp_top_inst.u_cp_top.u_axi_puts_severity_task.severity_task
File: /scratch/sipearl/bastien.hubert/sc_ws6_top/top-rhea/sub/soc/units/cp/source/sv/tb/axi_puts_severity_task.sv Line: 149
# ** Info: Checkpoint       11000 cycles at time 101668 ns.
#    Time: 101667964821 fs  Scope: tb_cdl_rhea_top.u_tb_rhea_top File: /scratch/sipearl/bastien.hubert/sc_ws6_top/top-rhea/sub/soc/units/top/source/verilog/tb/rtl/tb_rhea_top.sv Line: 193
# ** Info: puts_severity_task: SCP: Process RNSAMs...
#    Time: 102305463563 fs  Scope: tb_cdl_rhea_top.u_tb_rhea_top.u_rhea_top.top_center_q0_inst.top_center_q0_south_inst.top_sys_sh2_inst.top_sys_sh1_inst.top_sys_sh0_inst.sysctrl_inst.cp_top_inst.u_cp_top.u_axi_puts_severity_task.severity_task
File: /scratch/sipearl/bastien.hubert/sc_ws6_top/top-rhea/sub/soc/units/cp/source/sv/tb/axi_puts_severity_task.sv Line: 149
# ** Info: Checkpoint       12000 cycles at time 102918 ns.
#    Time: 102917964552 fs  Scope: tb_cdl_rhea_top.u_tb_rhea_top File: /scratch/sipearl/bastien.hubert/sc_ws6_top/top-rhea/sub/soc/units/top/source/verilog/tb/rtl/tb_rhea_top.sv Line: 193
```

Now: 103,745,464,204 fs Delta: 0 sim:/tb_cdl_rhea_top/u_tb_rhea_top/u_rhea_top - Limited Visibility Region

Waveform



Résultats

Nombre de *coverage items* évalués par les réalisations de tests

	T_{CC_1}	T_{CC_2}	T_{CC_3}	T_{DDR_1}	T_{HBM_1}	T_{PCIe_1}	T_{PCIe_1}
T_{CC_1}	**4579**	*2837*	*1619*	*841*	*925*	*454*	*557*
T_{CC_2}	*2837*	**7312**	*5452*	*1037*	*1129*	*236*	*325*
T_{CC_3}	*1619*	*5452*	**6821**	*769*	*631*	*378*	*417*
T_{DDR_1}	*841*	*1037*	*769*	**3427**	*947*	*158*	*206*
T_{HBM_1}	*925*	*1129*	*631*	*947*	**3546**	*141*	*189*
T_{PCIe_1}	*454*	*236*	*378*	*158*	*141*	**1658**	*1246*
T_{PCIe_2}	*557*	*325*	*417*	*206*	*189*	*1246*	**2246**

- *Taux de couverture total : 36.63% des 37920 coverage items du SLM*
- *Taux de réussite de 100% pour chaque réalisation de tests*

Résultats

Taux de redondance entre les réalisations de tests

	T_{CC_1}	T_{CC_2}	T_{CC_3}	T_{DDR_1}	T_{HBM_1}	T_{PCIe_1}	T_{PCIe_1}
T_{CC_1}	**12.08 %**	*7.48 %*	*4.27 %*	*2.22 %*	*2.44 %*	*1.20 %*	*1.47 %*
T_{CC_2}	*7.48 %*	**19.28 %**	*14.38 %*	*2.73 %*	*2.98 %*	*0.62 %*	*0.86 %*
T_{CC_3}	*4.27 %*	*14.38 %*	**17.99 %**	*2.03 %*	*1.66 %*	*1.00 %*	*1.10 %*
T_{DDR_1}	*2.22 %*	*2.73 %*	*2.03 %*	**9.04 %**	*2.50 %*	*0.42 %*	*0.54 %*
T_{HBM_1}	*2.44 %*	*2.98 %*	*1.66 %*	*2.50 %*	**9.35 %**	*0.37 %*	*0.50 %*
T_{PCIe_1}	*1.20 %*	*0.62 %*	*1.00 %*	*0.42 %*	*0.37 %*	**4.37 %**	*3.29 %*
T_{PCIe_2}	*1.47 %*	*0.86 %*	*1.10 %*	*0.54 %*	*0.50 %*	*3.29 %*	**5.92 %**

Matrice de Jaccard des réalisations de tests

	T_{CC_1}	T_{CC_2}	T_{CC_3}	T_{DDR_1}	T_{HBM_1}	T_{PCIe_1}	T_{PCIe_1}
T_{CC_1}	**100 %**	*31.32 %*	*16.55 %*	*11.75 %*	*12.85 %*	*7.87 %*	*8.89 %*
T_{CC_2}	*31.32 %*	**100 %**	*62.82 %*	*10.67 %*	*11.62 %*	*2.69 %*	*3.53 %*
T_{CC_3}	*16.55 %*	*62.82 %*	**100 %**	*8.12 %*	*6.46 %*	*4.68 %*	*4.82 %*
T_{DDR_1}	*11.75 %*	*10.67 %*	*8.12 %*	**100 %**	*15.73 %*	*3.23 %*	*3.74 %*
T_{HBM_1}	*12.85 %*	*11.62 %*	*6.46 %*	*15.73 %*	**100 %**	*2.77 %*	*3.39 %*
T_{PCIe_1}	*7.87 %*	*2.69 %*	*4.68 %*	*3.23 %*	*2.77 %*	**100 %**	*47.00 %*
T_{PCIe_2}	*8.89 %*	*3.53 %*	*4.82 %*	*3.74 %*	*3.39 %*	*47.00 %*	**100 %**

Conclusion

Conclusion

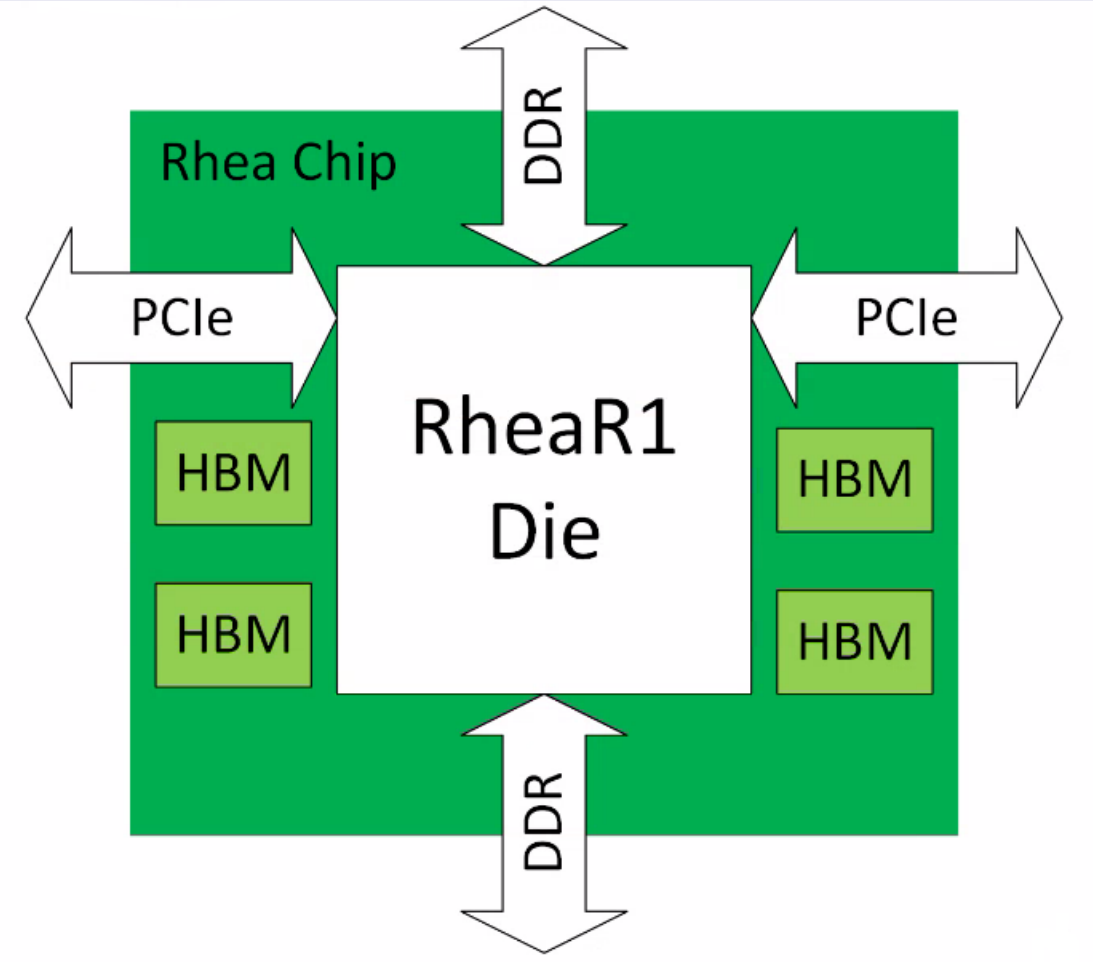
- *Pose d'un cadre théorique pour la vérification fonctionnelle*
- *Utilisation de 3 métriques simples pour une étude quantitative*
- *Réalisation d'un montage expérimental et analyse des résultats sur un exemple concret*
- *Étude de l'architecture de Rhea et réalisation de missions secondaires*

Ouverture

- *Généraliser le processus de cosimulation*
- *Ajouter des tests unitaires et des tests sur le VP*
- *Redresser l'indice de Jaccard en éliminant les coverage items non-pertinents*

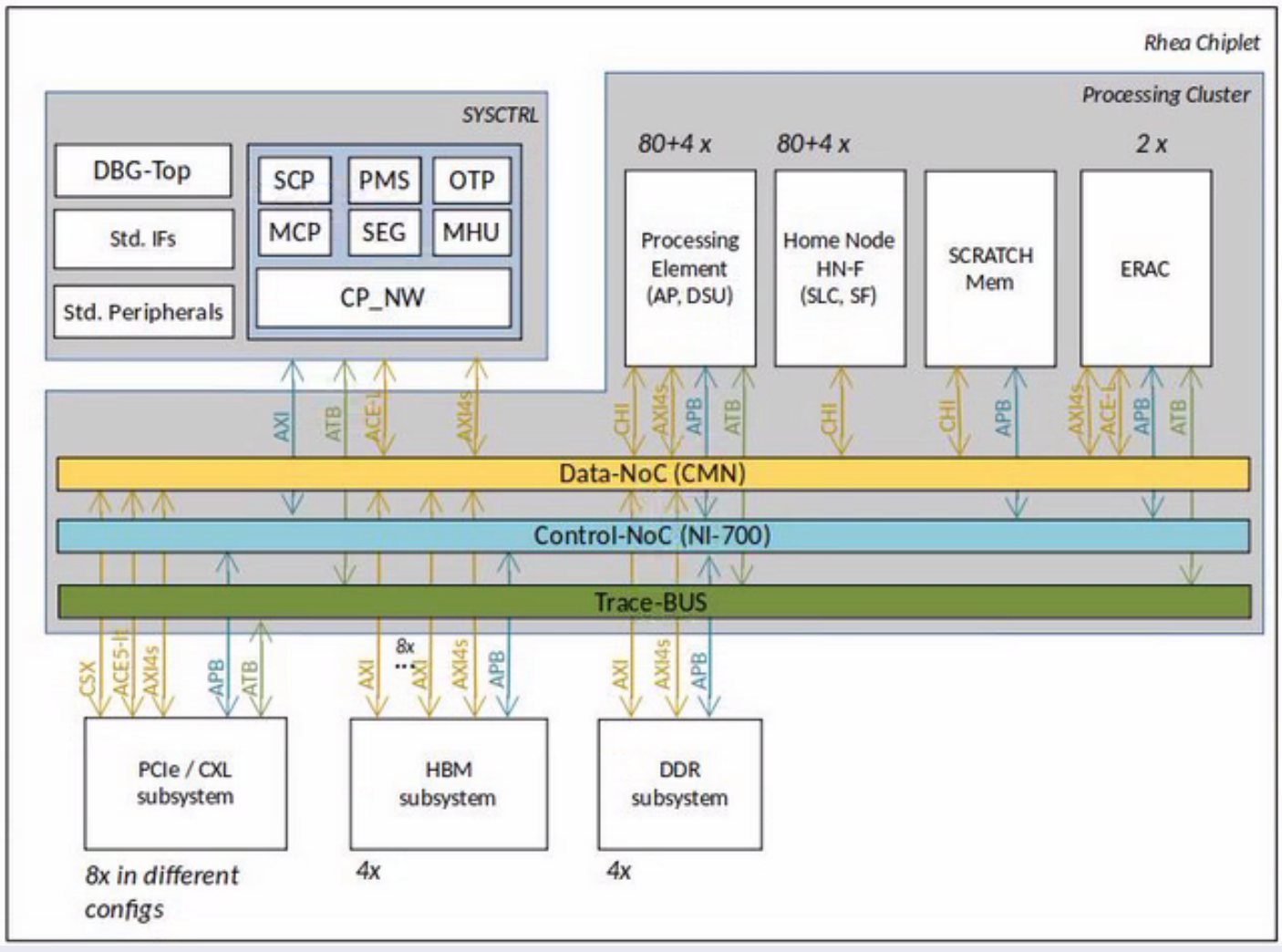
Architecture de Rhea

Vue d'ensemble de Rhea



Architecture de Rhea

Vue architecturale du *die* de Rhea



Principales références

Principales références

- *Eléments de théorie des automates, J. Sakarovitch, 2003*
- *"A Survey on Formal Verification Techniques for Safety-Critical Systems-on-Chip", T. Grimm, D. Lettnin, M. Hübner, 2018*
- *"Simulation-based equivalence checking between SystemC models at different levels of abstraction", D. Große, M. Groß, U. Kühne, R. Drechsler, 2011*
- *"Sequential equivalence checking between system level and RTL descriptions", S. Vasudevan, V. Viswanath, J. A. Abraham, J. J. Tu, 2006*
- *https://vlsiverify.com/uvm*