

Optimisation Multi-objectif du Processeur RISC-V CV32A6 pour l'Apprentissage Automatique

Soutenance de Stage

Bastien HUBERT

École Nationale Supérieure des Techniques Avancées Paris

vendredi 26 août 2022



Introduction

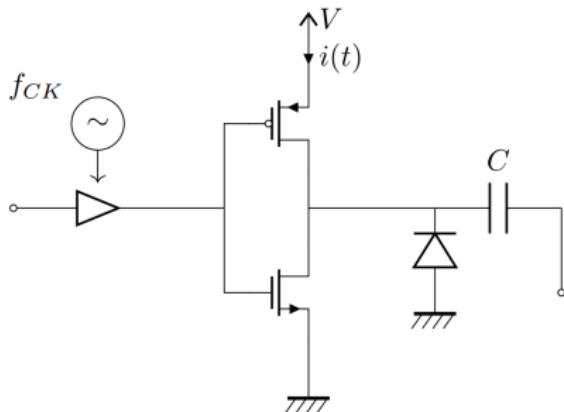
Motivations

- *Besoin d'autonomie dans les systèmes embarqués*
- *Applications de vision artificielle (apprentissage automatique, reconnaissance d'images) très énergivores*
- *RISC-V de plus en plus populaire dans les systèmes embarqués et l'IA car très modulaire*

Résultats du stage

- *Développement d'un cadre théorique extensif*
- *Établissement d'un flot de travail automatisable, très peu présent dans l'État de l'Art*
- *Obtention de résultats très encourageants sur un exemple concret*

Répartition de la Puissance Électrique dans un Circuit



$$P_{tot} = \underbrace{V \times I_{leak}}_{P_{static}} + \underbrace{\overbrace{V \times I_{SC} \times \tau_{SC} \times f_{CK}}^{P_{SC}} + P_{sw}}_{P_{dynamic}}$$

$$P_{sw} = \frac{V}{t_2 - t_1} \int_{t_1}^{t_2} i_{sw}(t) dt = \frac{V}{t_2 - t_1} \times C \times V \times (t_2 - t_1) \times f = C \times V^2 \times \alpha \times f_{CK}$$

Modèle Mathématique

Configuration de processeur

$$\forall N \in \mathbb{N}, \forall c \in \mathbb{N}^N, c \in \mathcal{C} \stackrel{\text{def}}{\iff} \exists k \in \mathbb{N} \mid \left\{ \begin{array}{ll} \forall i \in \llbracket 1, k \rrbracket, & \exists! j \in \llbracket 1, N \rrbracket : c_j = i \\ & \forall i > k, \quad \nexists j \in \llbracket 1, N \rrbracket : c_j = i \end{array} \right.$$

$c_j = 0$ \Rightarrow paramètre non utilisé
 $c_j = i \in \llbracket 1, k \rrbracket$ \Rightarrow paramètre utilisé en position i

$(0, 0, 0), (0, 1, 0), (2, 0, 1), (2, 3, 1) \in \mathcal{C}$
 $(1, 0, 3), (2, 2, 1), (0, 0, -1) \notin \mathcal{C}$

$\forall k \in \mathbb{N}, \mathcal{C}_k := \{c \in \mathcal{C} \mid \deg(c) = k\}$

$$\#\mathcal{C}_k = k! \times \binom{N}{N-k} = \frac{N!}{(N-k)!}$$

Formule de Taylor-Lagrange

$$e = \sum_{i=0}^N \frac{1}{i!} + \int_0^1 \frac{(1-t)^N}{N!} e^t dt$$

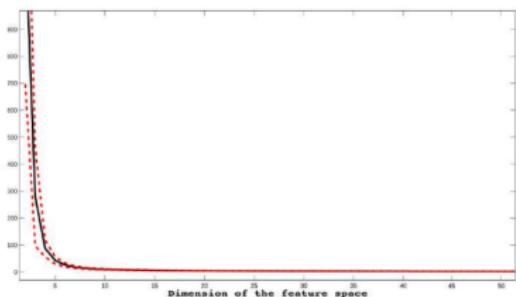
$$\#\mathcal{C} = \sum_{k=0}^N \#\mathcal{C}_k = N! \times \left(e - \int_0^1 \frac{(1-t)^N}{N!} e^t dt \right)$$

$$\#\mathcal{C} \underset{N \rightarrow \infty}{\sim} N! e$$

Fléau de la Dimension

$$d : \left\{ \begin{array}{ccc} \mathcal{C} \times \mathcal{C} & \rightarrow & \mathbb{R}_+ \\ (c, c') & \mapsto & \sqrt{\sum_{k=1}^N (c_k - c'_k)^2} \end{array} \right.$$

$$R := \frac{\max_{(i,j) \in [\![1, M]\!]^2, i \neq j} \{d(v_i, v_j)\}}{\min_{(i,j) \in [\![1, M]\!]^2, i \neq j} \{d(v_i, v_j)\}} \xrightarrow[N \rightarrow \infty]{} 1$$



$$E^0, P^0, R^0 : \left\{ \begin{array}{ccc} \mathcal{C} & \rightarrow & \mathbb{R}_+ \\ c & \mapsto & \frac{E(0_c)}{E(c)}, \frac{P(c)}{P(0_c)}, \frac{R(0_c)}{R(c)} \end{array} \right.$$

Fonction de fitness

$$f : \left\{ \begin{array}{ccc} \mathcal{C} & \rightarrow & \mathcal{EP}\mathcal{R} := \mathbb{R}^3 \\ c & \mapsto & (E^0(c), P^0(c), R^0(c)) \end{array} \right.$$

$$d_{eff} : \left\{ \begin{array}{ccc} \mathcal{EP}\mathcal{R} \times \mathcal{EP}\mathcal{R} & \rightarrow & \mathbb{R}_+ \\ (x, y) & \mapsto & \sqrt{\sum_{k=1}^3 (x_k - y_k)^2} \end{array} \right.$$

$(\mathcal{C}, (f, f) \circ d_{eff})$ est un espace pseudo-métrique

Réduction des Paramètres

L -distinction

$$\forall L \in \mathcal{P}(\llbracket 1, N \rrbracket), \forall (c, c') \in \mathcal{C}^2, \\ c \mathcal{R}_L c' \stackrel{\text{def}}{\iff} \forall i \in \llbracket 1, N \rrbracket \setminus L, c_i = c'_i$$

Deux configurations L -distinctes ne peuvent différer que par leurs valeurs dont les indices sont dans L

L -variations

$$[c]_L^0 := \{c' \in [c]_L \mid \forall i \in L, c'_i \neq 0\}$$

Une L -variation de c lui est identique à une permutation des valeurs dont les indices sont dans L près

$$\forall k \in \llbracket 1, N \rrbracket, \mathcal{C}_k = \bigsqcup_{L \in \{\mathcal{P}(\llbracket 1, N \rrbracket) \mid \#L=k\}} [0_c]_L^0$$

$$\forall L \in \{I \in \mathcal{P}(\llbracket 1, N \rrbracket) \mid \#I = k\}, \\ D(L) := \max_{(c, c') \in ([0_c]_L^0)^2} \{d_{\text{eff}}(f(c), f(c'))\}$$

$D(L)$ petit correspond à des paramètres entre lesquels l'ordre importe peu

Clustering

Pour $D(L_0) < \varepsilon$, les paramètres d'indices dans L_0 sont remplacés par un uplet ordonné de ces paramètres

ε -clustering : répété tant que $D(L_0) < \varepsilon$

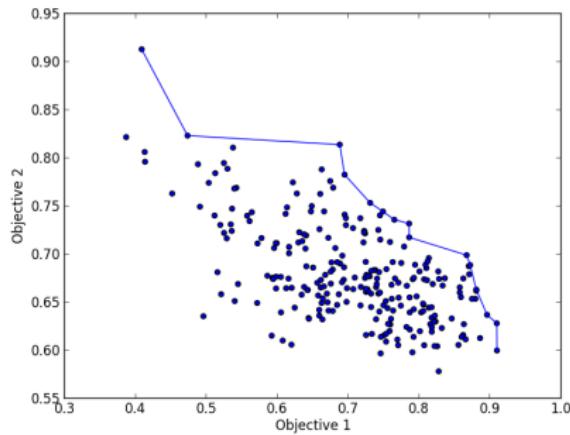
n -clustering : répété n fois

Surface de Pareto

Relation de domination

$$\forall (x, y) \in \mathbb{R}^2, x \succeq y \stackrel{\text{def}}{\iff} \forall k \in \llbracket 1, 3 \rrbracket, x_k \geq y_k$$

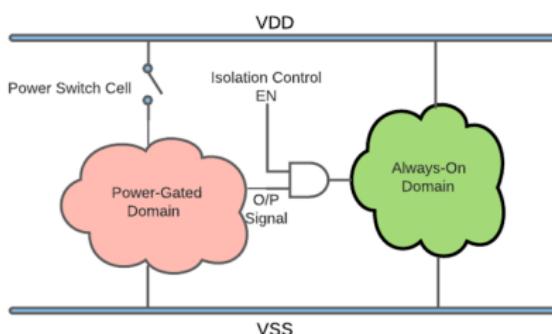
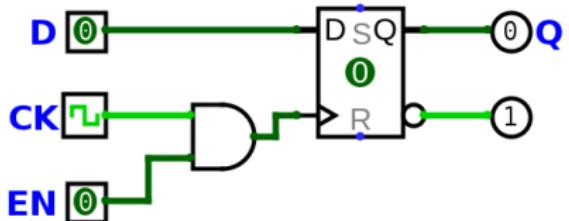
$$c \in \mathbf{P}(\mathcal{C}) \stackrel{\text{def}}{\iff} \nexists c' \in \mathcal{C} \setminus \{c\} \mid f(c') \succeq f(c)$$



Techniques Basse Puissance

Clock Gating, Power Gating

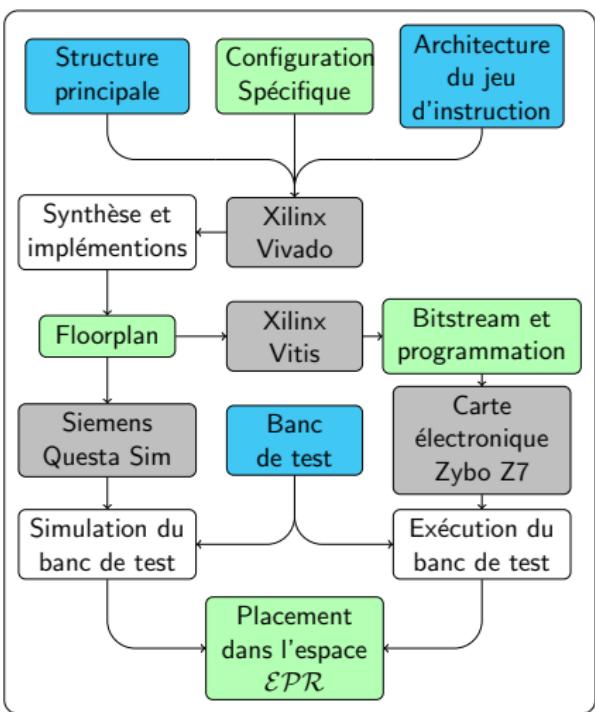
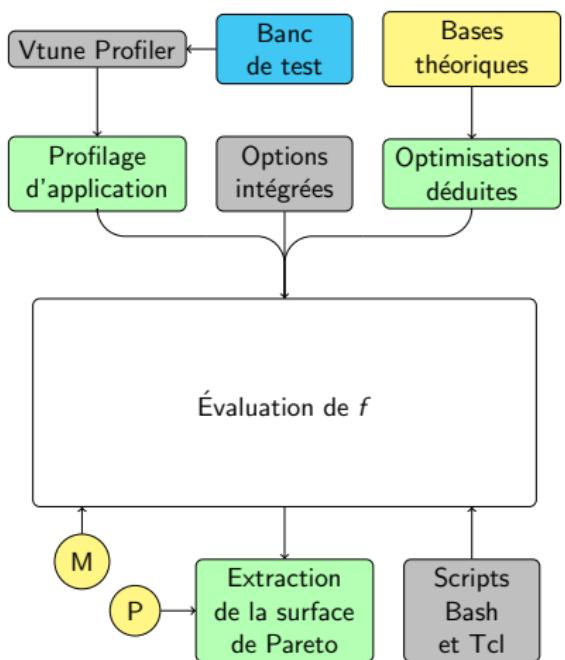
But : faire diminuer α pour réduire la puissance dissipée



Frequency and Voltage Islands, DVFS

Créer des sous-horloges/sources d'alimentation en fonction des performances des sous-circuits, statiquement ou dynamiquement

Méthodologie



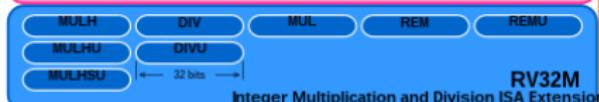
Architecture RISC-V

RV32IMAC



RV32A

Atomic Instruction ISA Extension



RV32M

Integer Multiplication and Division ISA Extension



RV32I

Base Integer ISA

C.LW C.AND

C.FLW C.ANDI

C.FLD C.OR

C.LWSP C.XOR

C.FLWSP C.LI

C.FLDSP C.LUI

C.SW C.SLLI

C.FSW C.SRLI

C.FSD C.SRAI

C.SWSP C.BEQZ

C.FSWSP C.BNEZ

C.FSDSP C.J

C.ADD C.JR

C.ADDI C.JAL

C.ADDI16SP C.JALR

C.ADDI4SPN C.EBREAK

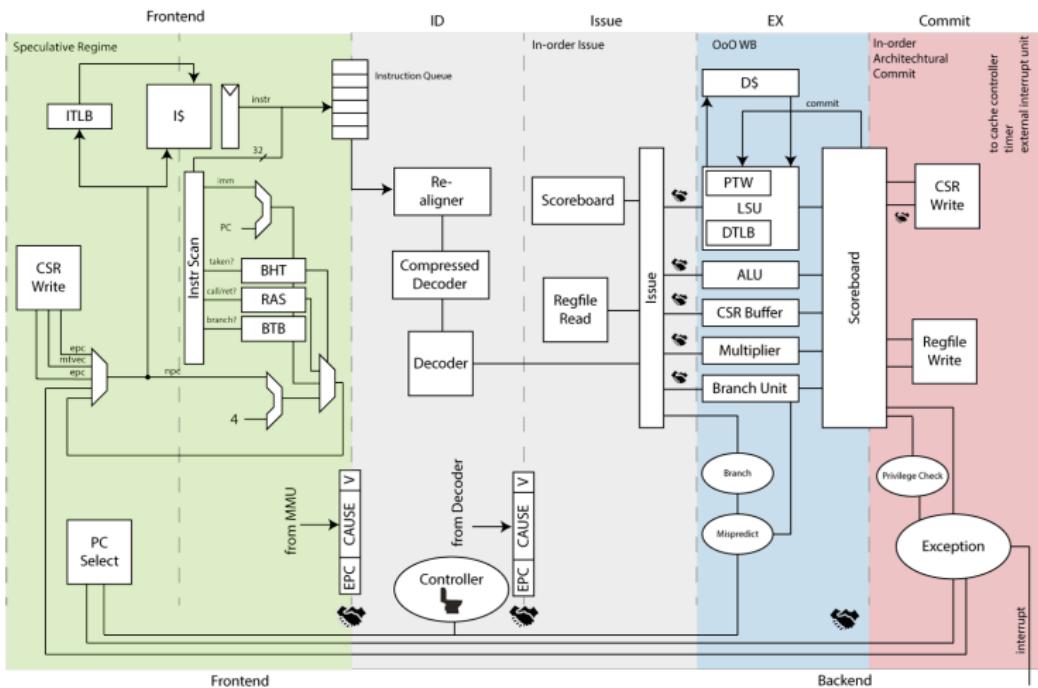
C.SUB C.MV

RV32C

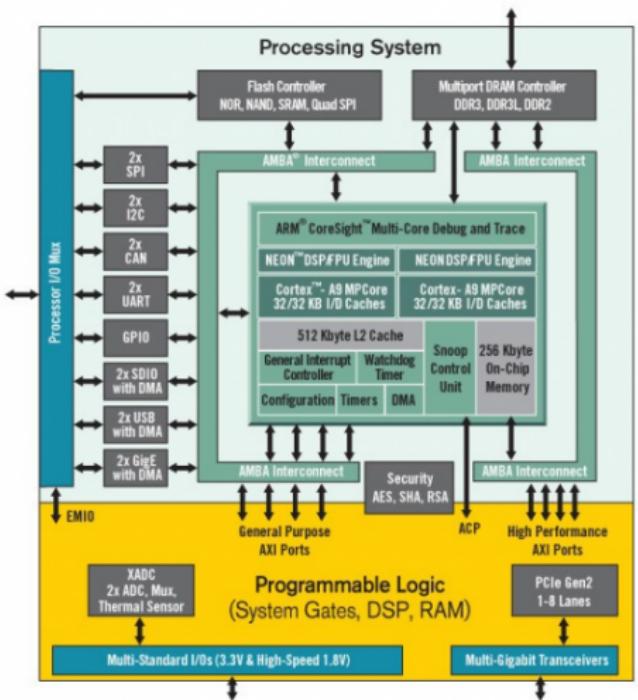
Compressed ISA Extension



Processeur ARIANE/CV32A6

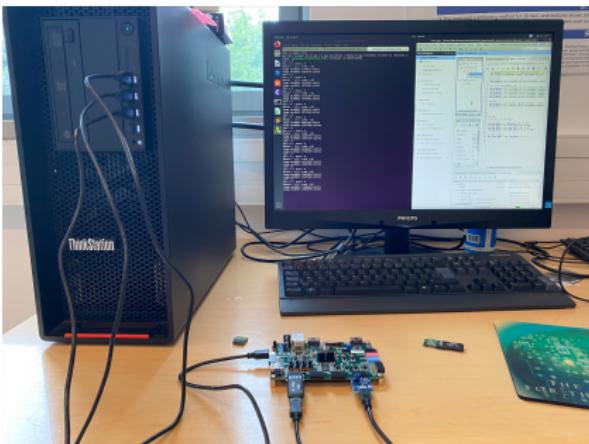
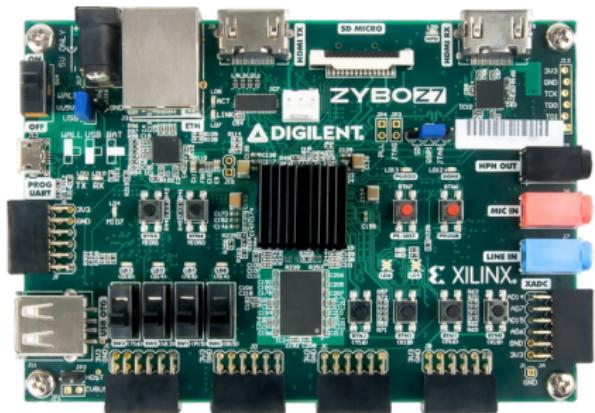


Circuit FPGA Xilinx reconfigurable ZYNQ 7020

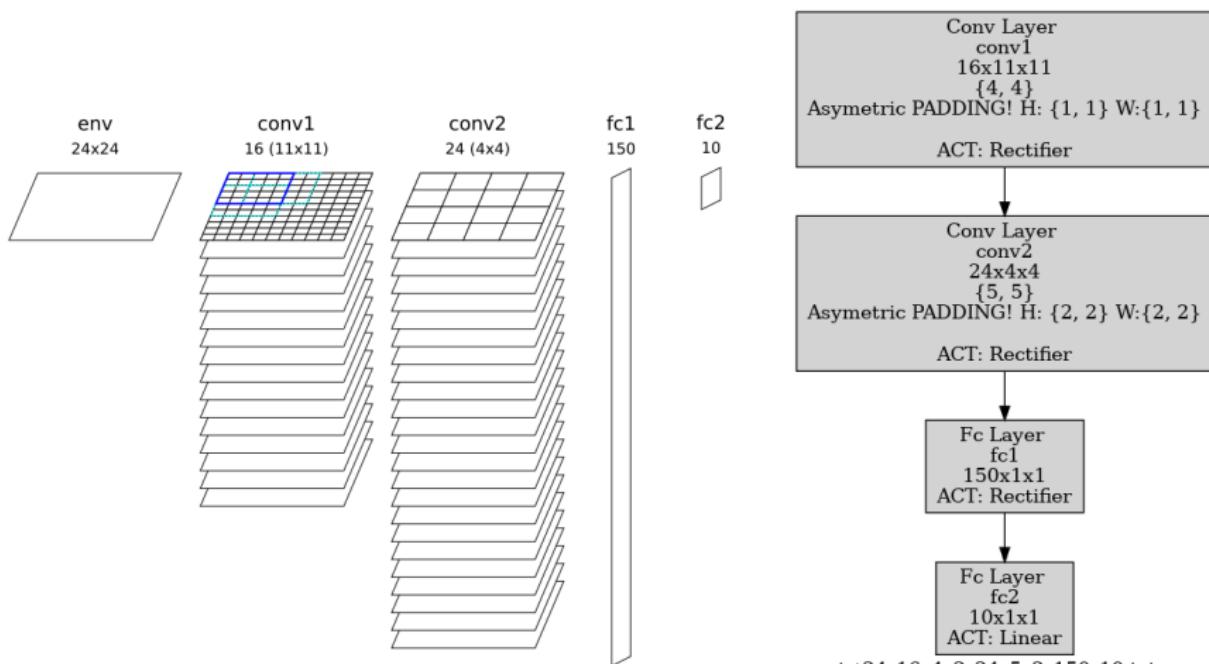


Device Name	Z-7020
Part Number	XC7Z020
Xilinx 7 Series Programmable Logic Equivalent	Artix-7 FPGA
Programmable Logic Cells	85K
Look-Up Tables (LUTs)	53,200
Flip-Flops	106,400
Block RAM (# 36 Kb Blocks)	4.9 Mb (140)
DSP Slices (18x25 MACCs)	220
Peak DSP Performance (Symmetric FIR)	276 GMACs
PCI Express (Root Complex or Endpoint) ⁽³⁾	

Carte électronique SoC complexes Zybo Z7



Application : Réseau de neurones convolutif (sélectionné par Thalès)

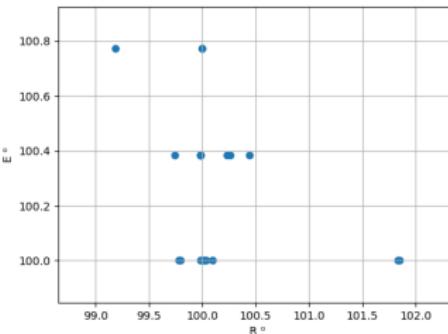
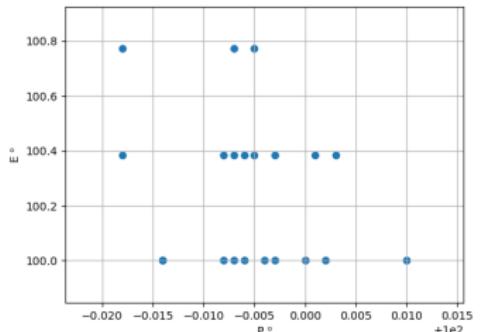
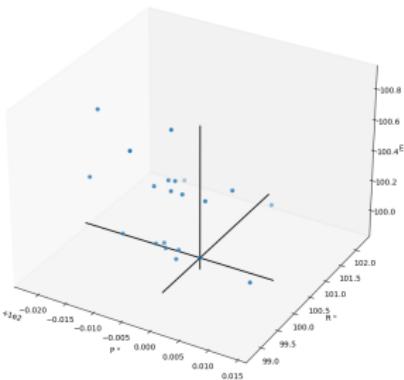
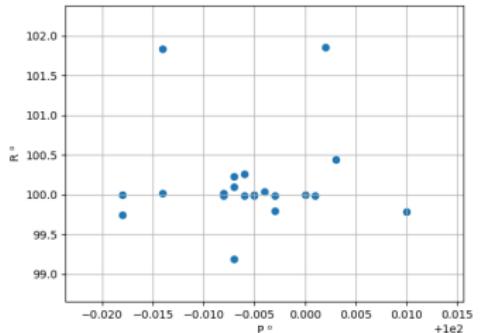


Outils de conception et de modélisation

<i>Distribution</i>	<i>Ubuntu 18.04</i>
<i>Description HDL</i>	<i>SystemVerilog</i>
<i>Synthèse et implementation</i>	<i>Xilinx Vivado 2020.1</i>
<i>Génération du bitsream</i>	<i>Xilinx Vivado 2020.1</i>
<i>Exécution du banc de test</i>	<i>Xilinx Vitis 2020.1</i>
<i>Simulation du banc de test</i>	<i>Siemens Questa Sim 10.7</i>
<i>Automatisation de l'évaluation</i>	<i>Scripts Tcl</i>
<i>Automatisation du flot</i>	<i>Scripts Bash</i>

3 outils de CAO industriels complexes

*SystemVerilog est un langage de conception
et de modélisation standard dans l'industrie*

Résultats de l'évaluation \mathcal{EPR} 

Conclusion

- *Sujet de stage complexe, multidisciplinaire*
- *En phase direct avec les besoins des grands acteurs de l'industrie des semi-conducteurs*
- *Traitements de questions théoriques mathématiques (problème d'optimisation et dimensionnalité)*
- *Prise en main de nombreux outils de CAO industriels complexes*
- *Objectifs du PRe atteint et production de résultats scientifiques de recherche originaux absents de l'État de l'Art*
- *Travail facilement extensible pour des résultats plus fournis*
- *Soumission d'un article scientifique IEEE pour le 31 août*