UMR IRISA

UNIVERSITE BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE RENNES 1

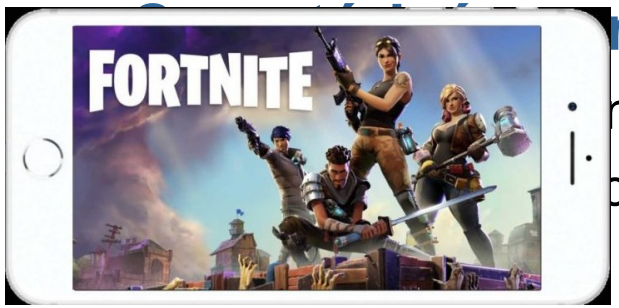
Accélération Matérielle pour la Traduction Dynamique de Programmes Binaires

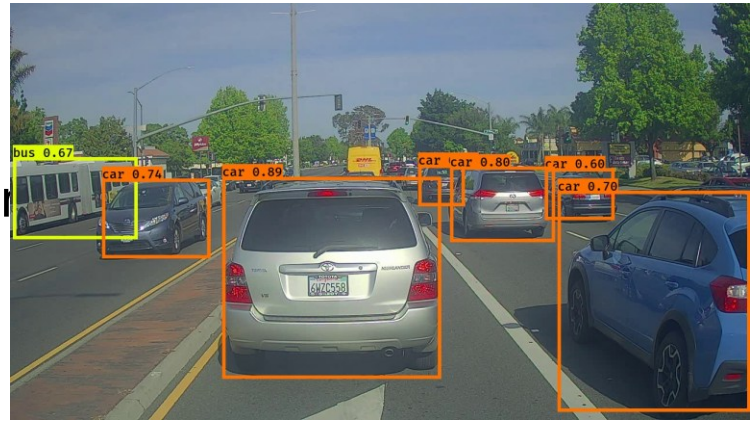
Simon Rokicki, Steven Derrien, Erven Rohou

01/02/2019

Qu’est-ce qu’un système embarqué ?

- **Système autonome, parfois spécialisé pour une tâche**
- **Contraintes fortes sur :**
  - **Performance**
  - **Consommation**



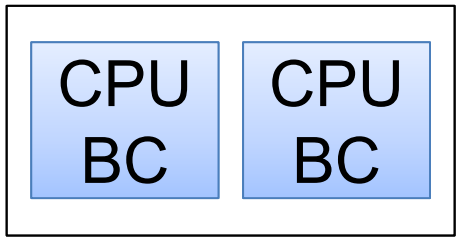


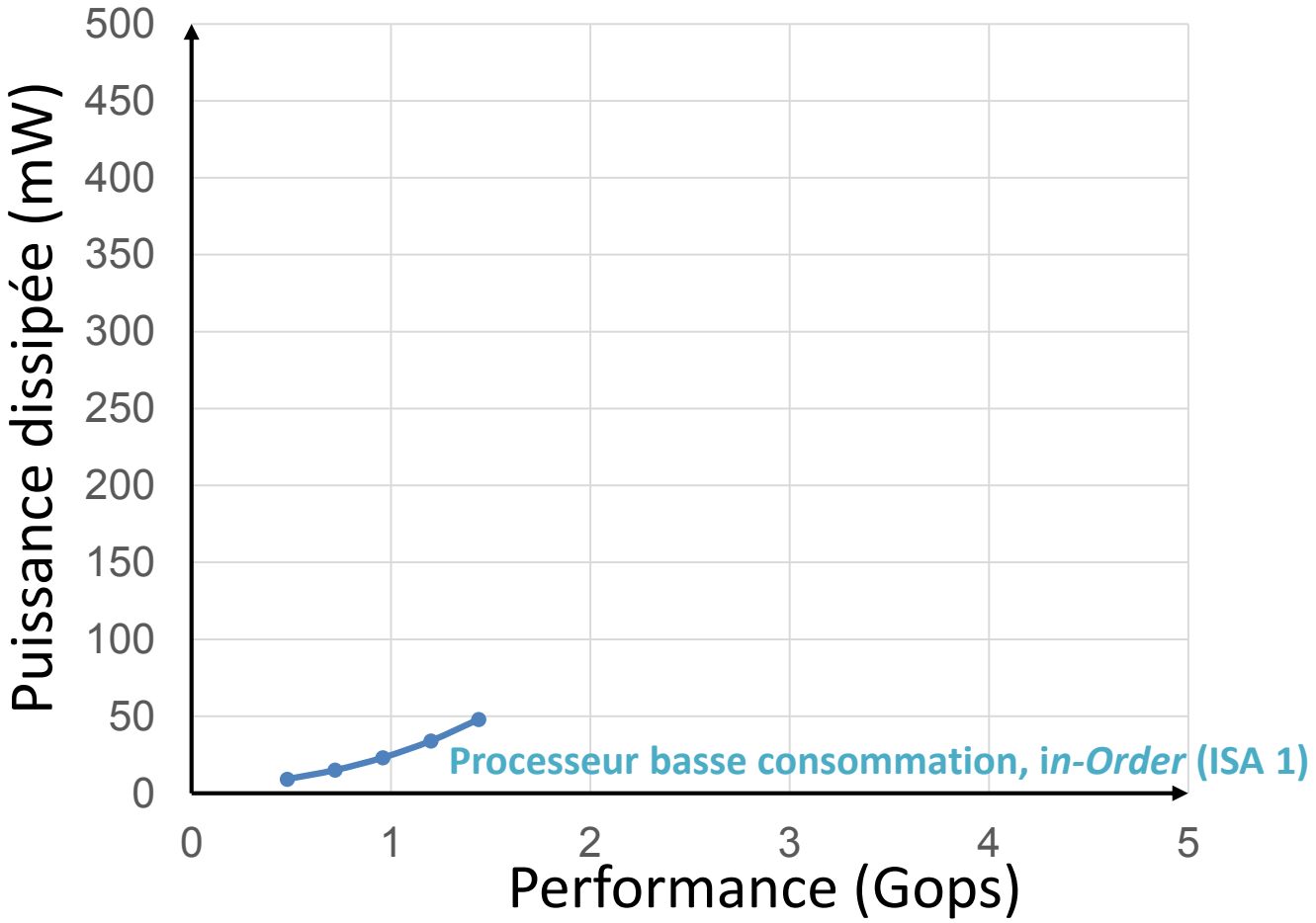
Compromis performance/énergie

- Différents profils de fonctionnement
  - Besoin de minimiser la consommation énergétique
  - Besoin de maximiser les performances
- Contrôle dynamique du compromis performance/énergie

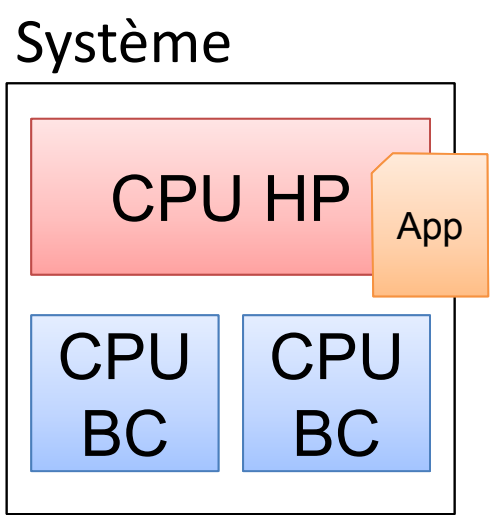
Compromis performance/énergie

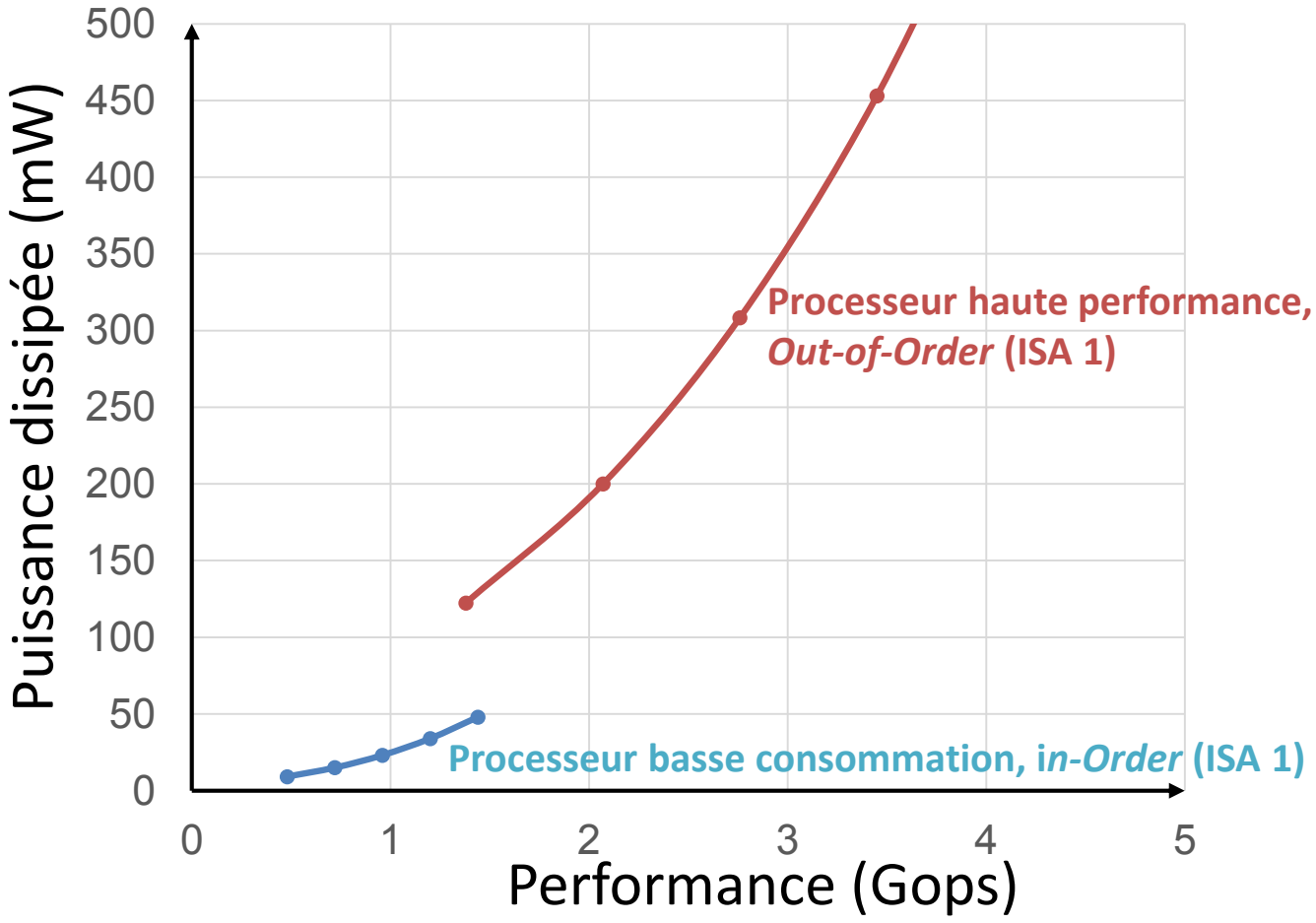
Système





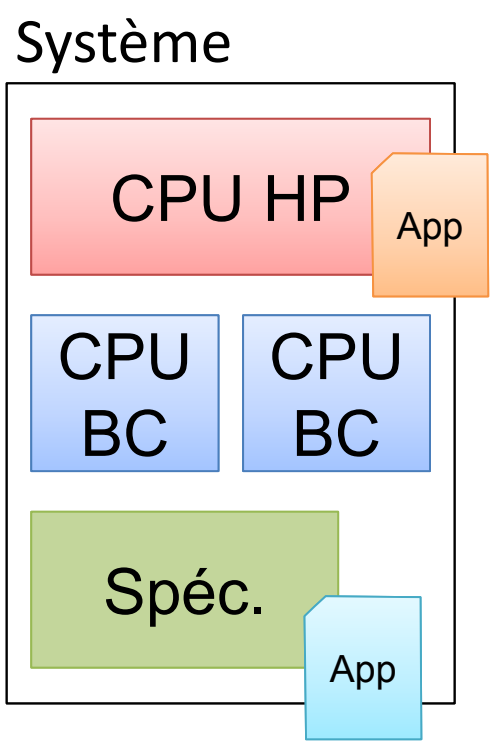
Compromis performance/énergie

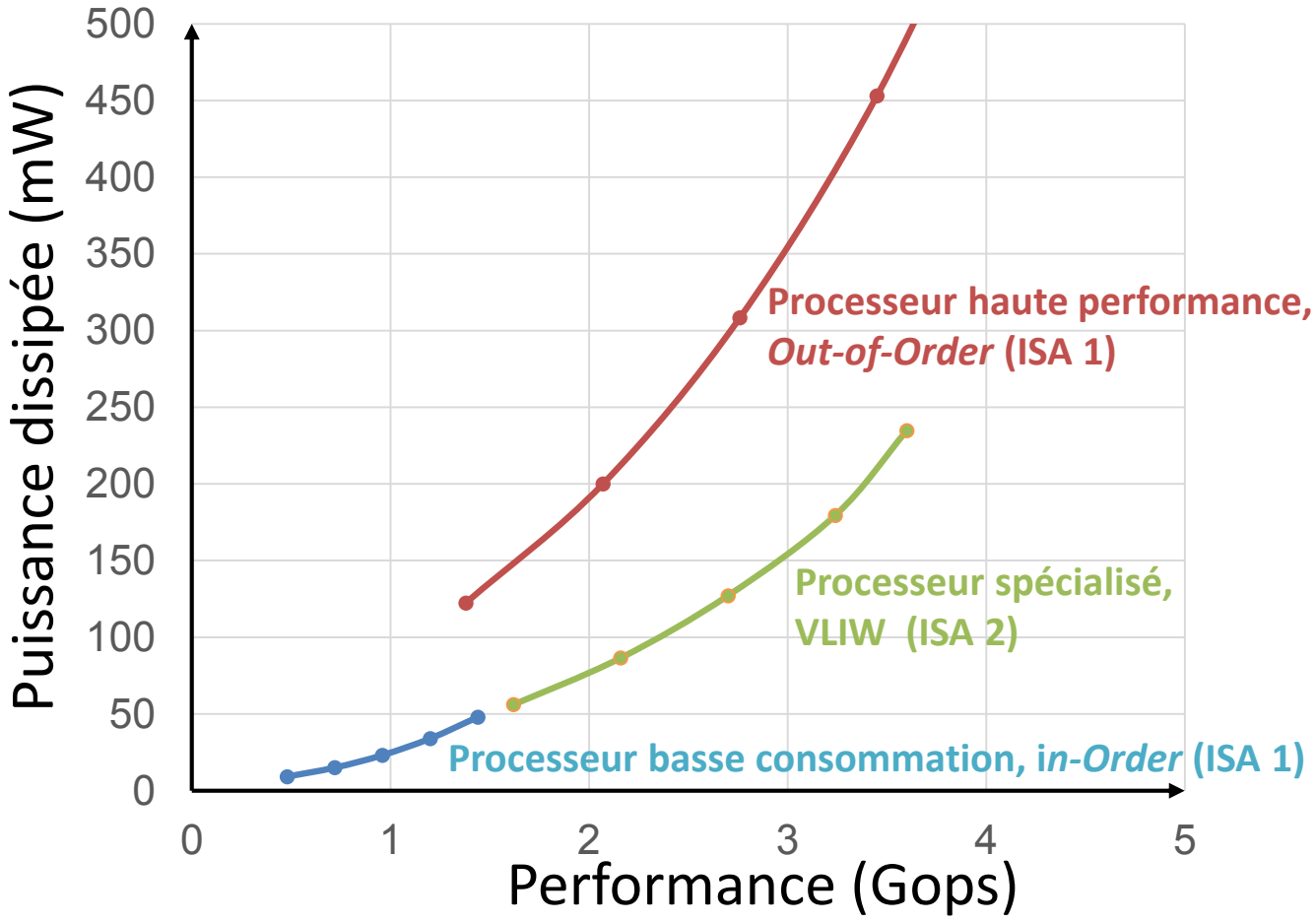




ISA : *Instruction Set Architecture* – Jeu d'instructions

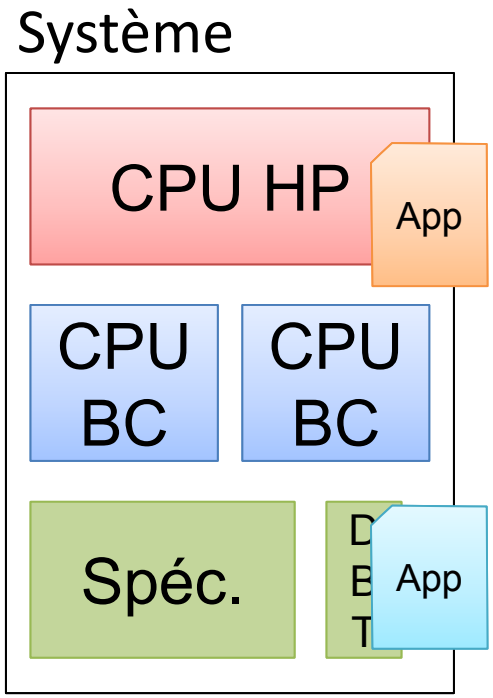
Compromis performance/énergie





ISA : *Instruction Set Architecture* – Jeu d'instructions

Traduction dynamique de binaires



- **Principe :**

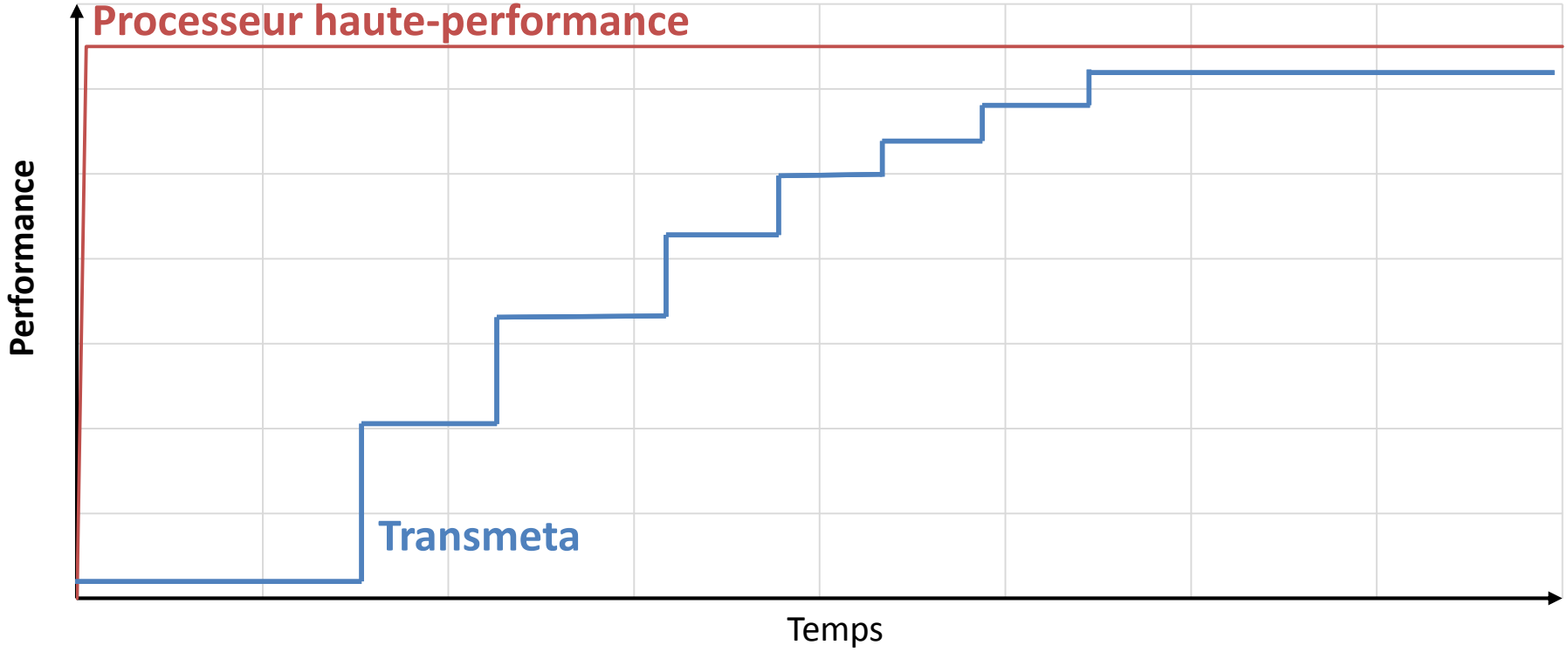
  Traduire et optimiser (pendant l'exécution) une application d'un ISA 1 vers un ISA 2

- **Exemples de processeurs basés sur la DBT :**

TRANSMETA™ NVIDIA®

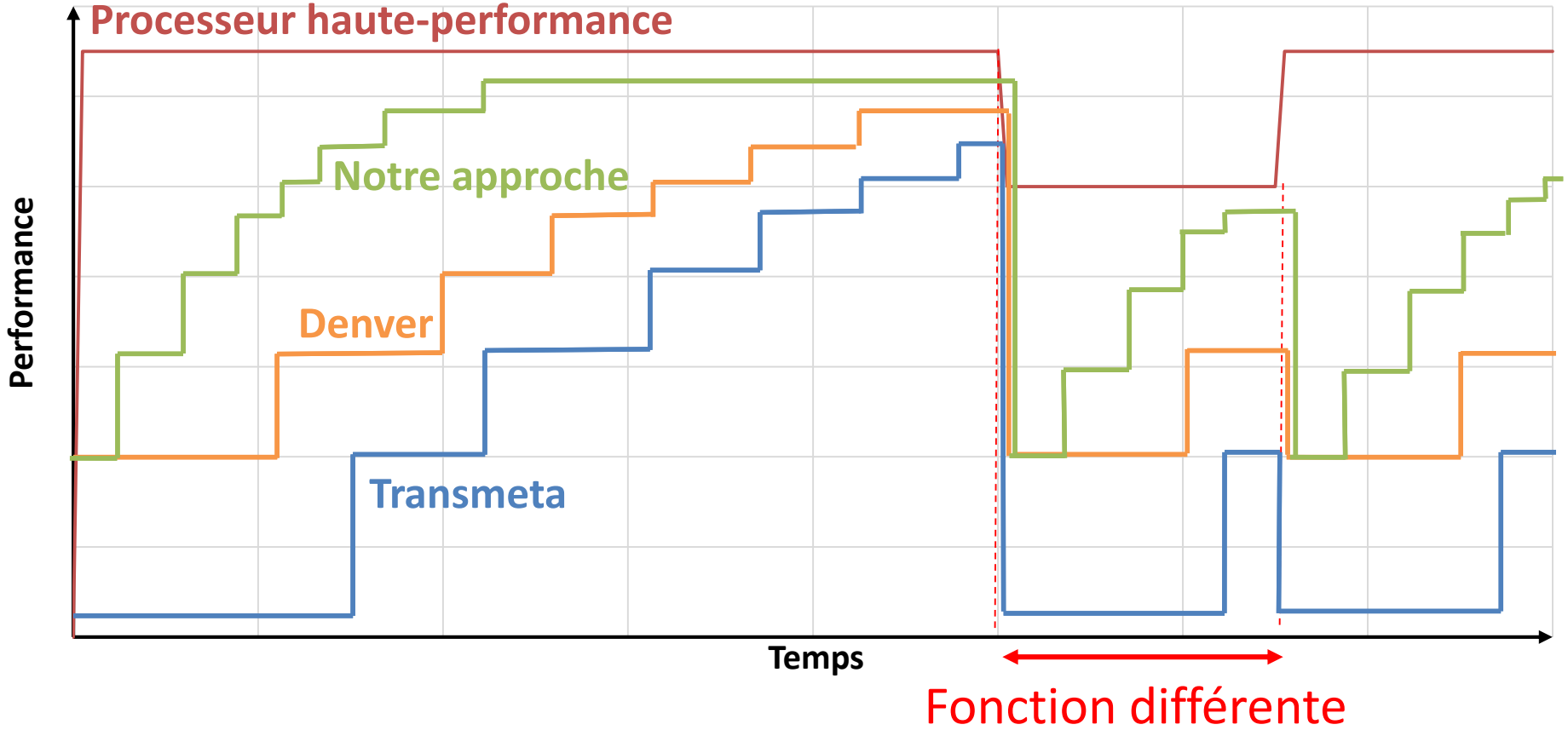
DBT : *Dynamic Binary Translation* – Traduction dynamique de binaires

Performance de la DBT

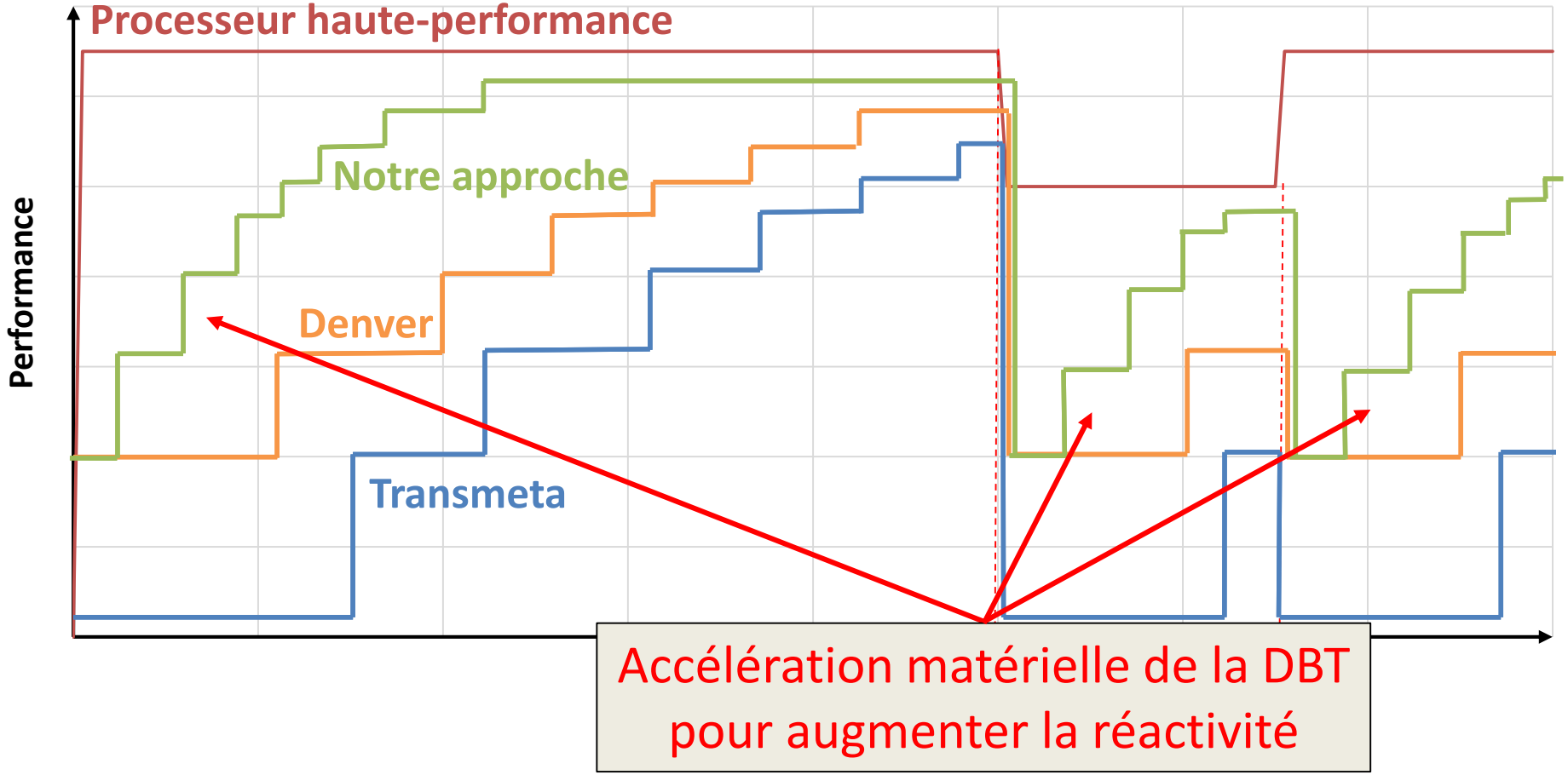


Références : *McFarlin2013*

Performance de la DBT



Performance de la DBT

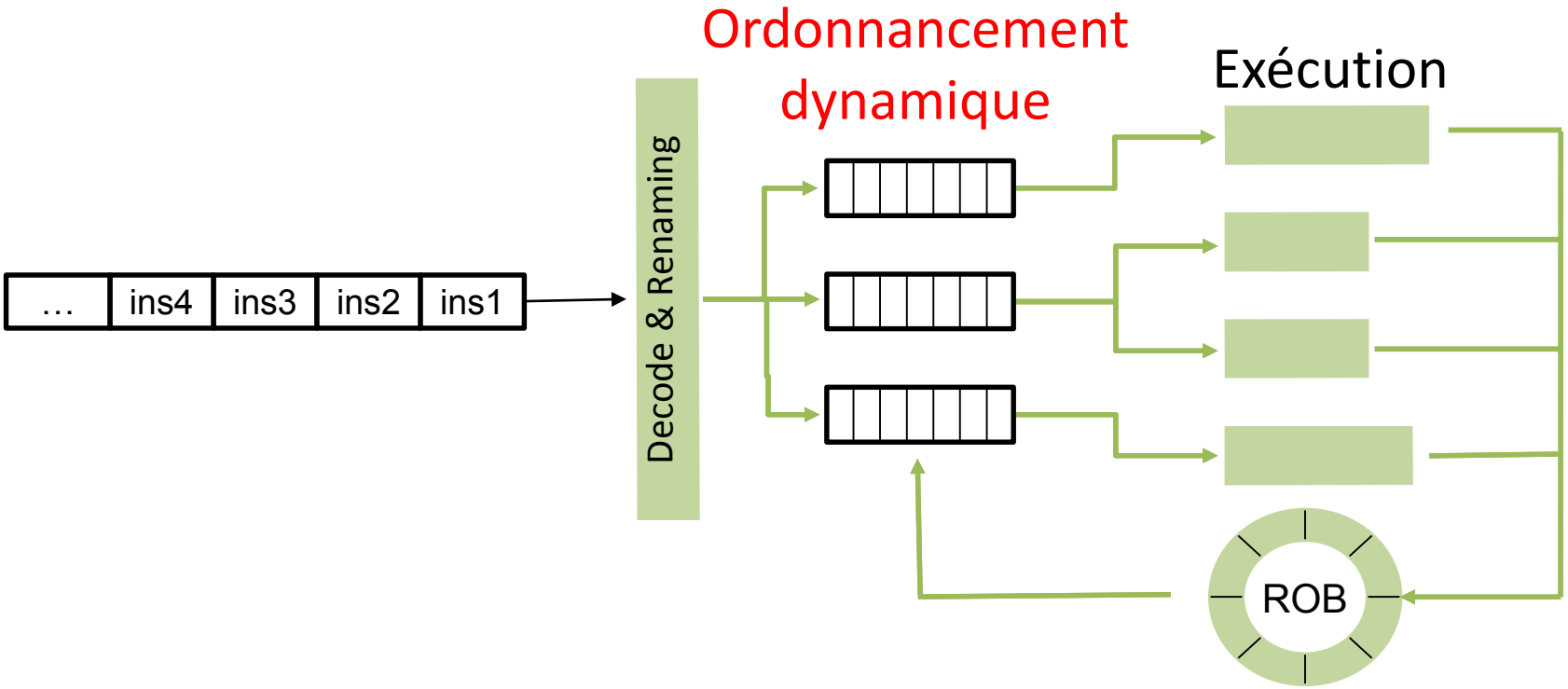


Plan de la présentation

- **Etat de l'art**
  - Architecture des processeurs
  - Traduction Dynamique de Binaires
- **Accélération matérielle de la DBT**
- **Conclusion**

Processeurs Out-of-Order (OoO)

- **Extraction dynamique de l'ILP**
- **Exécution spéculative**



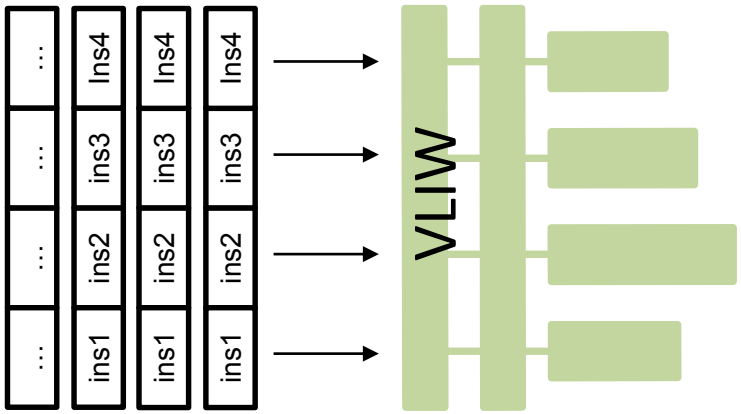
ROB: *Re-Order Buffer*

ILP: *Instruction Level Parallelism* – Parallélisme d'instruction

Les OoO dans l'embarqué

- **Mécanismes très couteux**
  - Les ordonnancements sont recalculés à chaque exécution
- **Comment réduire ces coûts ?**
  - Dans l'embarqué : processeurs OoO moins agressifs
  - Essayer de ne pas recalculer les ordonnancements
- **Approches proposées :**
  - réutilisation sur des architectures plus spécialisées
    (*Padmanabha2017*, *Brandalero2017*, *Villavieja2014*)
  - Preuves de concept académiques

Processeurs VLIW



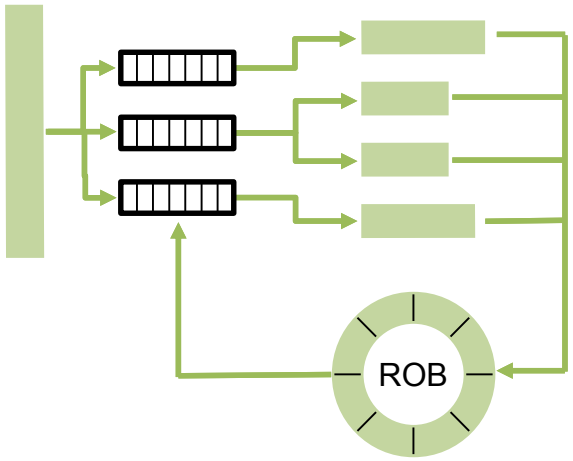
Bundles : Groupements d'instructions indépendantes

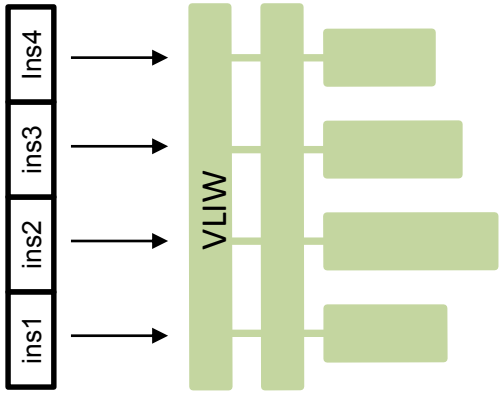
- **ISA explicitement parallèles**
  - Efficacité énergétique
- **ILP extrait par le compilateur :**
  - Ordonnancement des instructions
  - Allocation de registres
  - Construction de traces, *superblock*
  - Pipeline logiciel
- **Très utilisés dans l'embarqué**
  - Qualcom Hexagon

Références : *Hwu1993, Mahlke1992, Pinter1993, Heydemann2006*

ILP: *Instruction Level Parallelism* – Parallélisme d'instruction

Synthèse



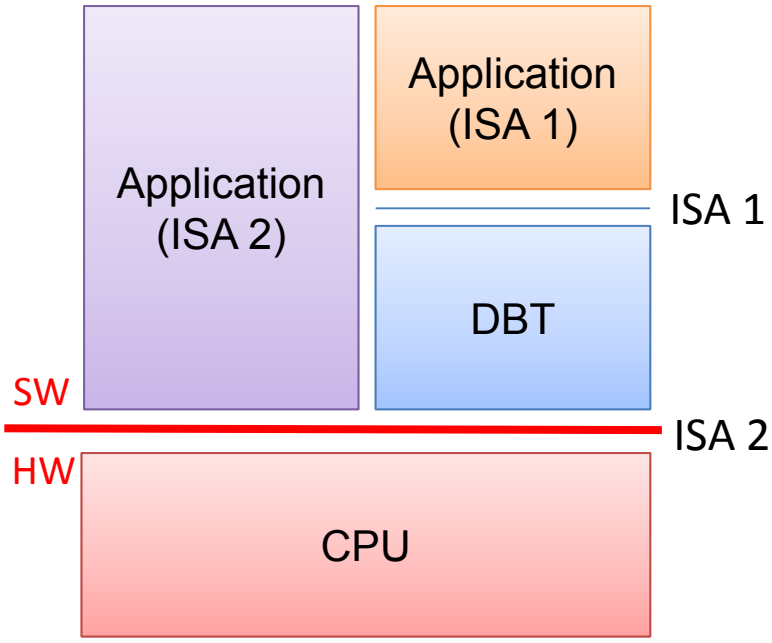


- **Processeur OoO :**
  - Ordonnancement dynamique
  - Consommation énergétique élevée
  - ISA séquentiel
- **Processeur VLIW :**
  - Ordonnancement statique
  - Efficacité énergétique
  - ISA explicitement parallèle

Plan de la présentation

- **Etat de l'art**
  - Architecture des processeurs
  - Traduction Dynamique de Binaires
- **Accélération matérielle de la DBT**
- **Conclusion**

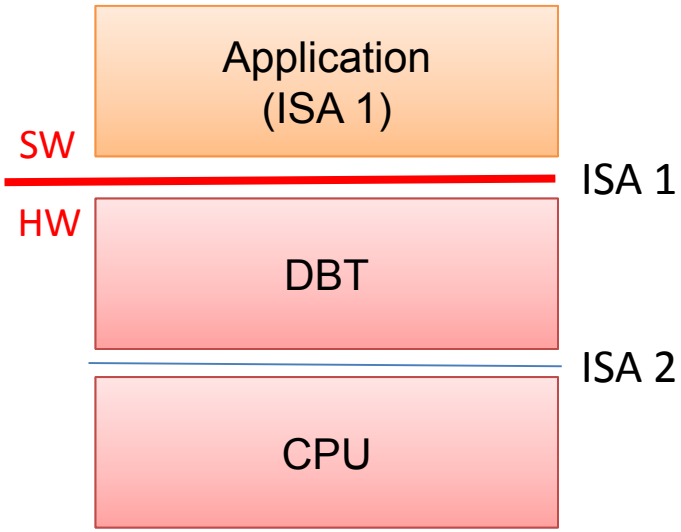
Traduction dynamique de binaires



- **Principe de la DBT :**
  - Traduction d'une application compilée pour un ISA 1 vers un ISA 2
- **Objectifs usuels :**
  - Simulation rapide (Qemu)
  - Compatibilité binaire (Apple Rosetta)
- **DBT pour VLIW**
  - IBM Daisy
  - Intel IA32 Execution Layer

Références : *Bellard2005, Ebcioğlu97, Baraz2003*

HW/SW co-designed machines

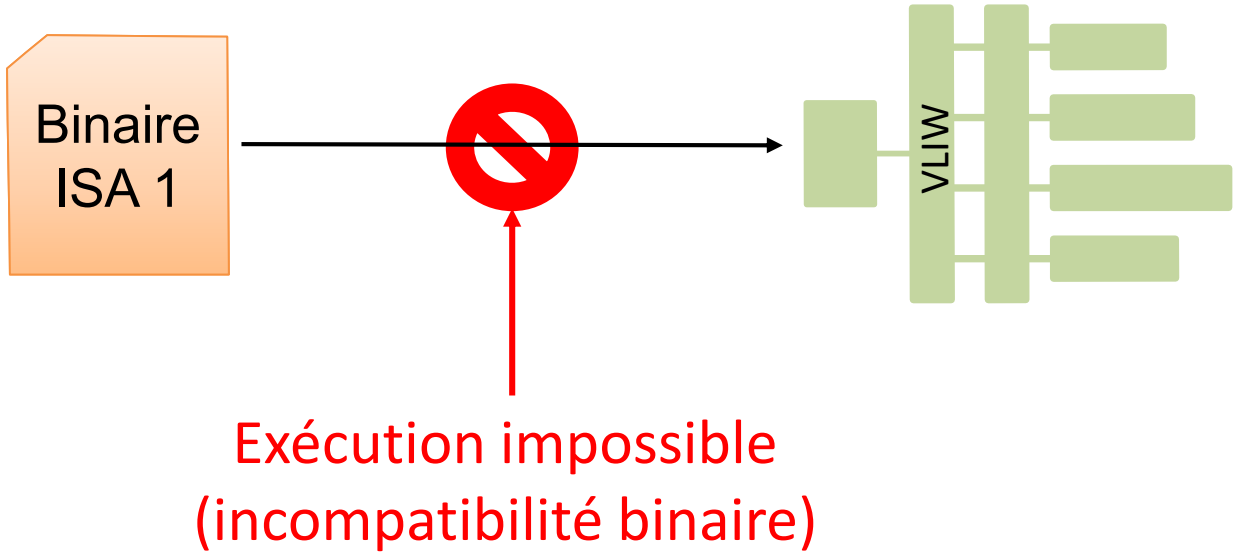


- Compatible ISA 1
- ISA 2 non exposé à l'utilisateur
- Objectif : performances
- Exemples :
  - micro-code Intel (x86 -> RISC)
  - Transmeta CMS (x86 -> VLIW)
  - Nvidia Denver (Arm -> VLIW)

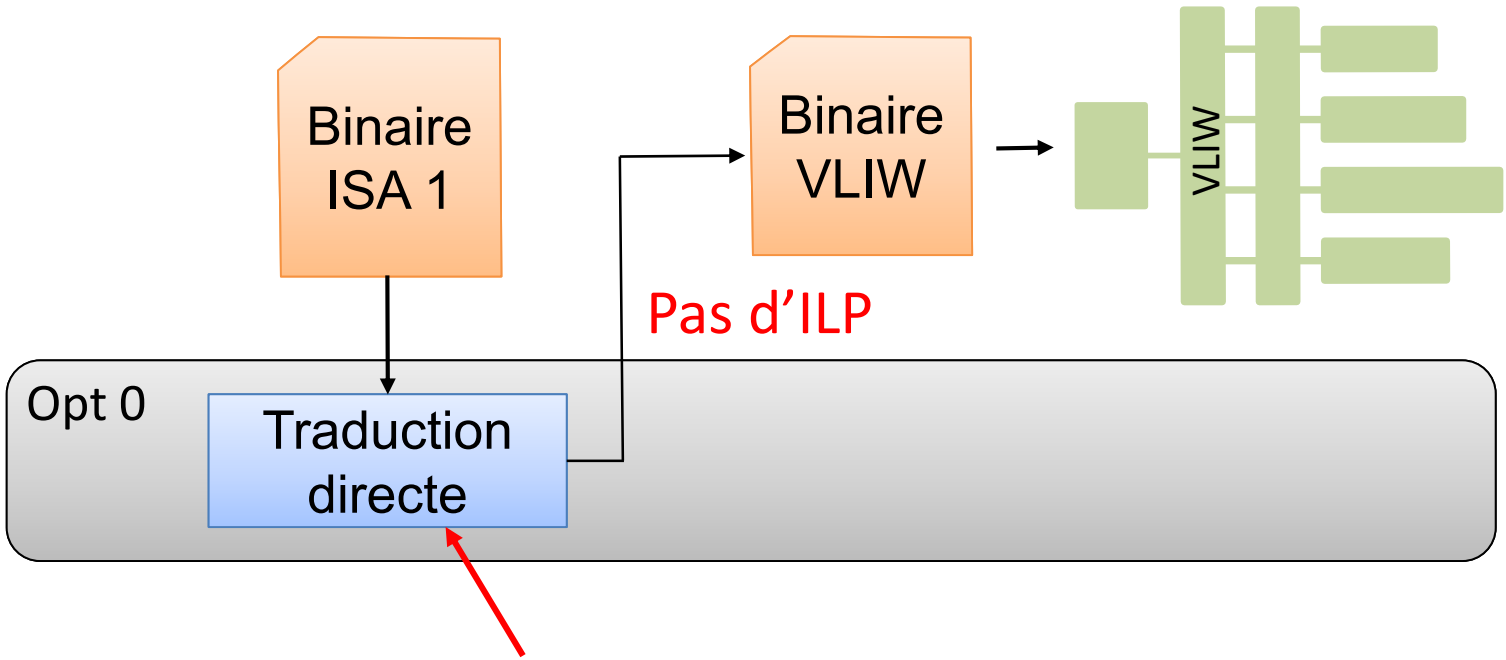
La performance et la qualité de la DBT sont critiques

Références : *Dehnert2003, Boggs2015*

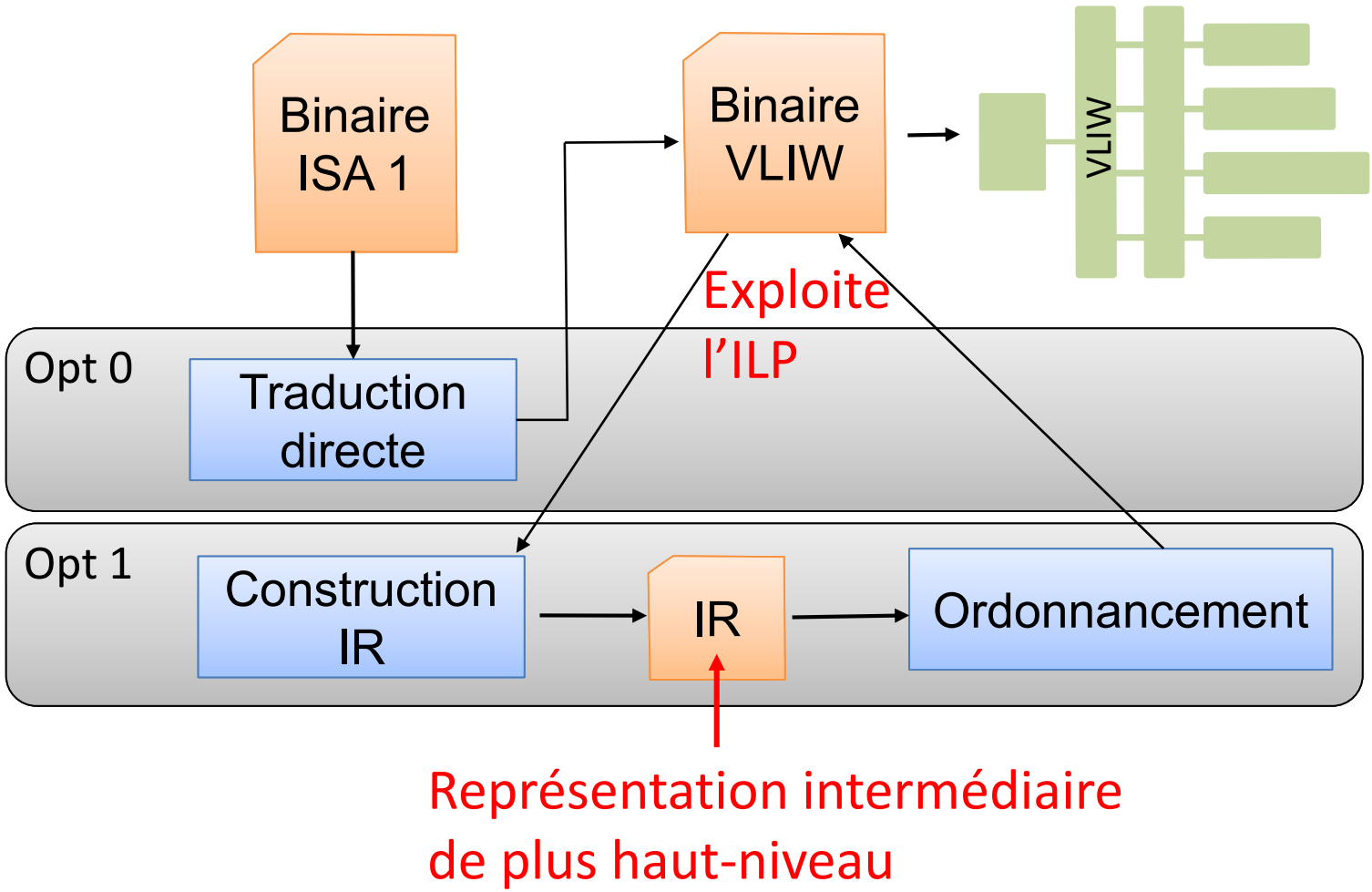
Enjeux de la traduction



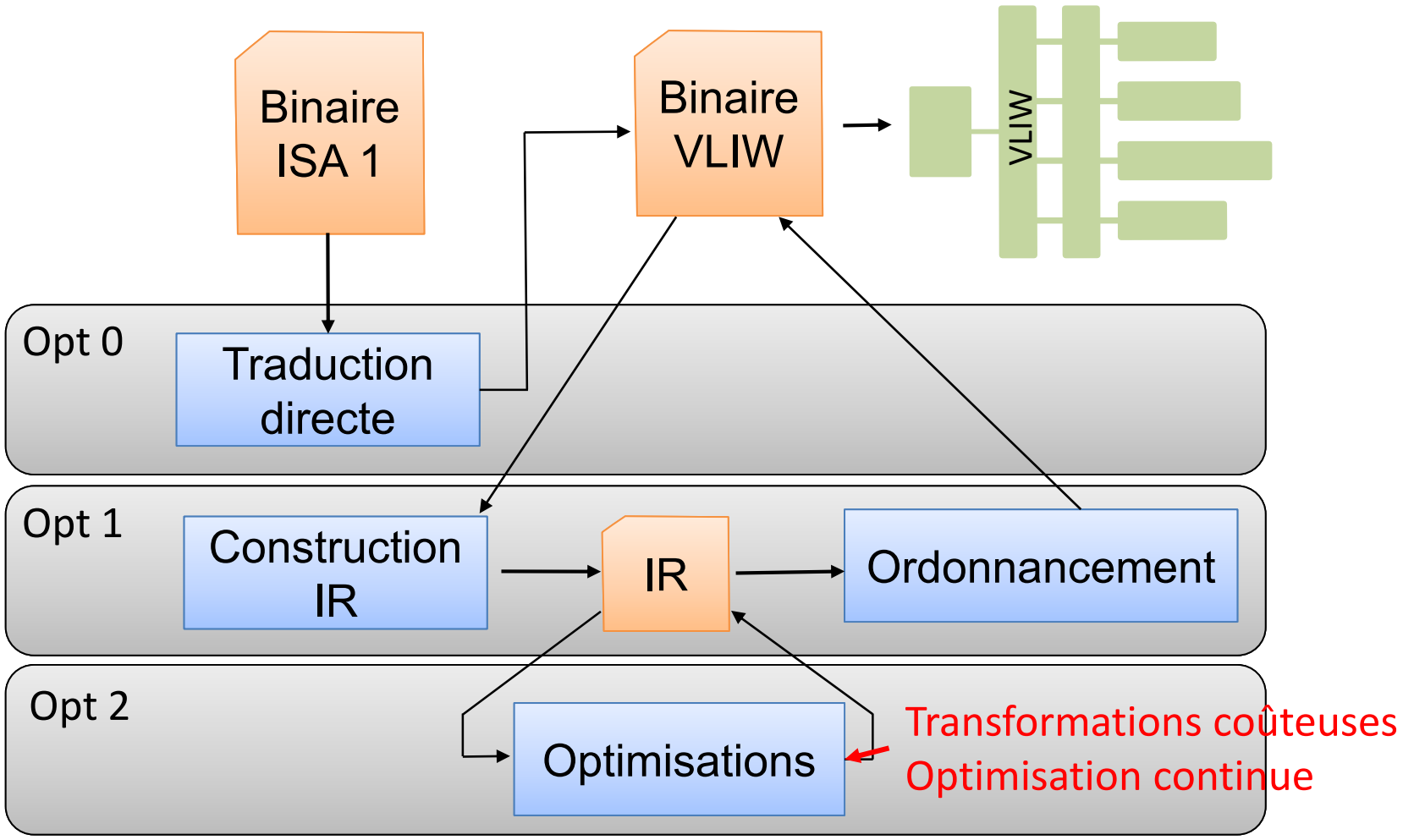
Enjeux de la traduction



Enjeux de la traduction



Enjeux de la traduction

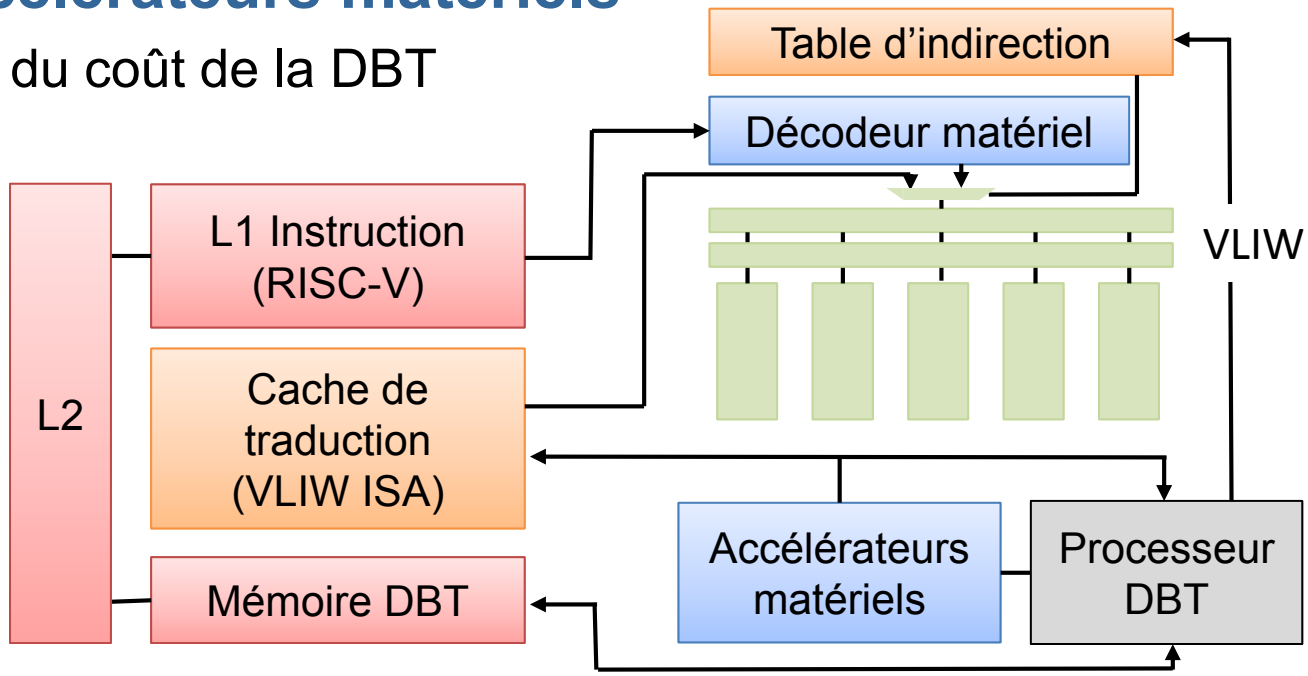


Plan de la présentation

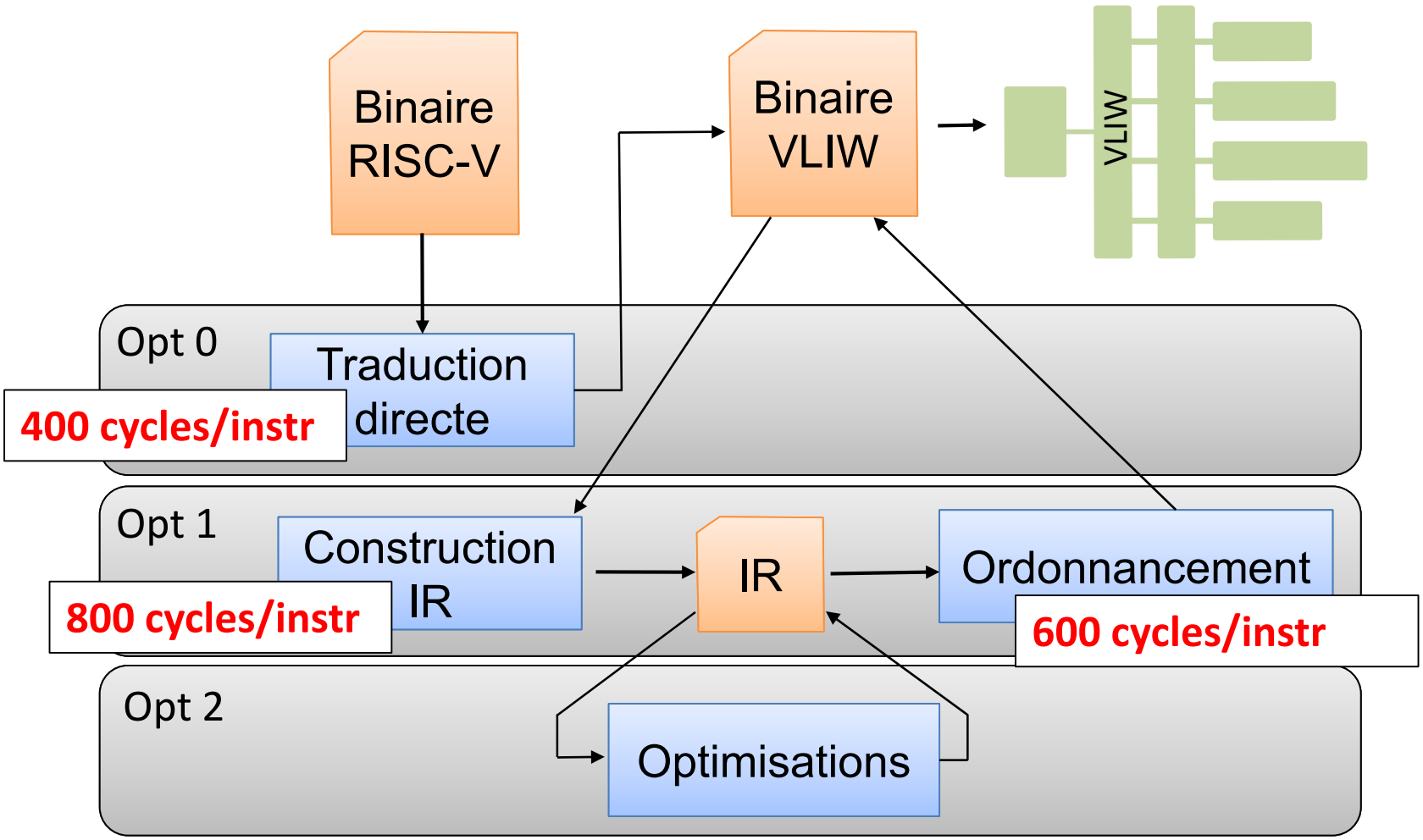
- **Etat de l'art**
  - Architecture des processeurs
  - Traduction Dynamique de Binaires
- **Accélération matérielle de la DBT**
- **Conclusion**

Notre approche

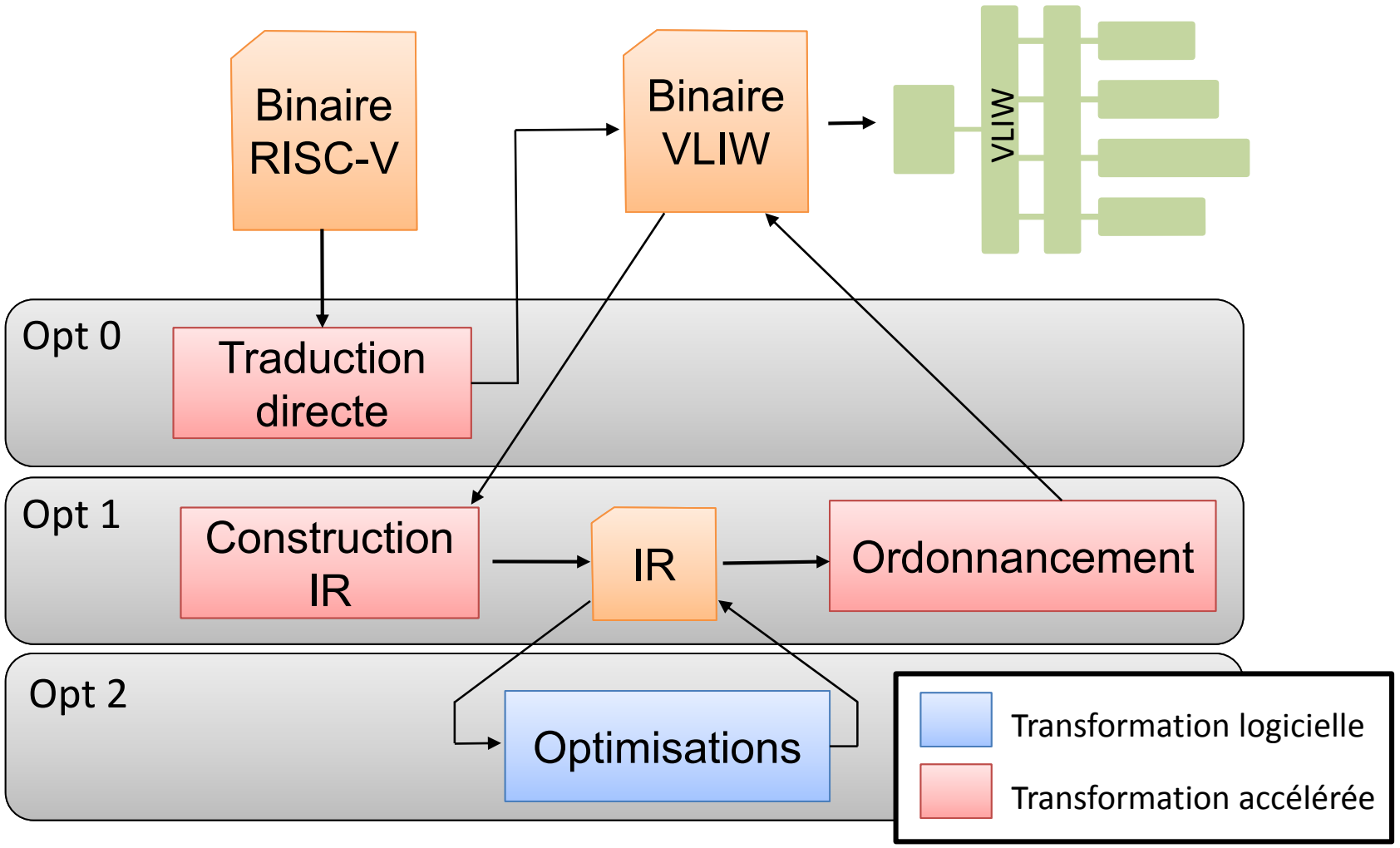
- **RISC-V vers VLIW**
- **Ajout d'un processeur dédié à la DBT**
  - Optimisation en parallèle de l'exécution
  - Optimisation en avance de phase
- **Ajout d'accélérateurs matériels**
  - Réduction du coût de la DBT



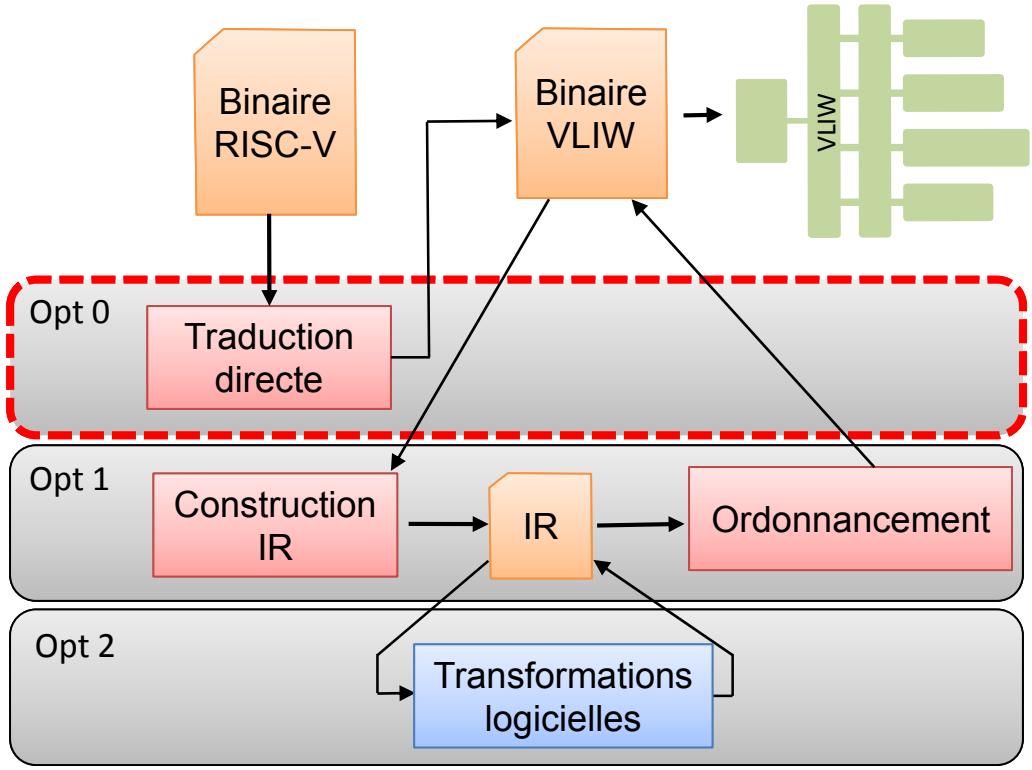
Qu'est-ce qui doit être accéléré ?



Qu'est-ce qui doit être accéléré ?

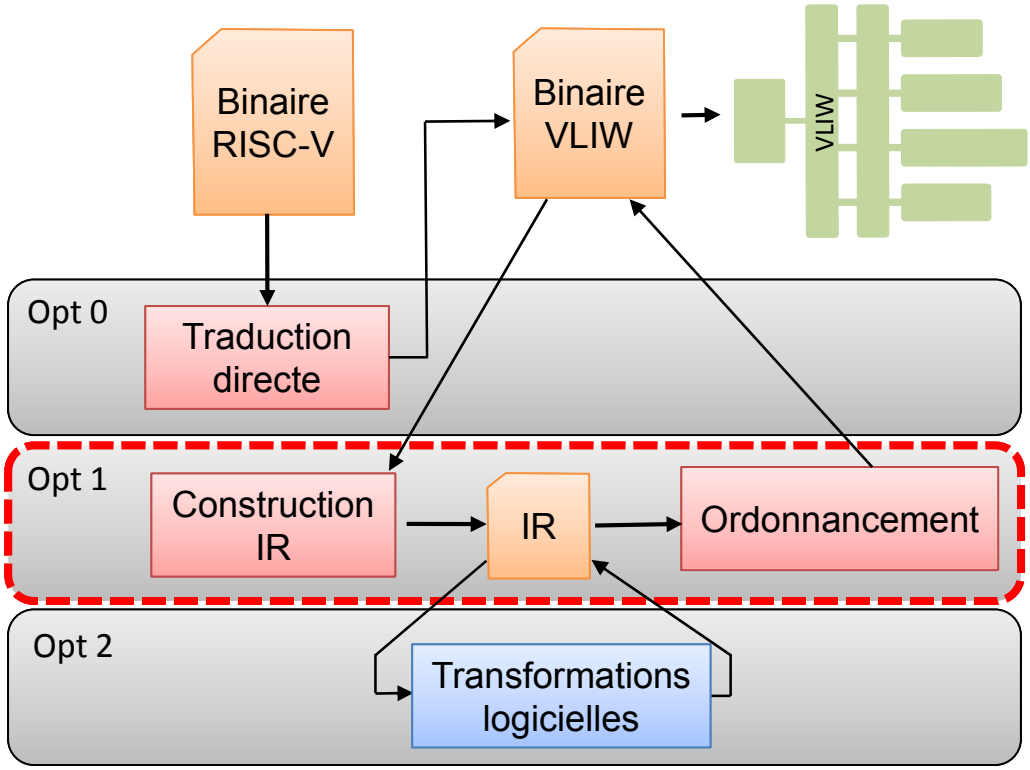


Niveau d'optimisation 0



- Traduction individuelle de chaque instructions
- Jeux d'instructions similaires

Niveau d'optimisation 1



- Construction du graphe de flot de données
- Ordonnancement des instructions
- Renommage des registres

Représentation intermédiaire

Binaire RISC-V

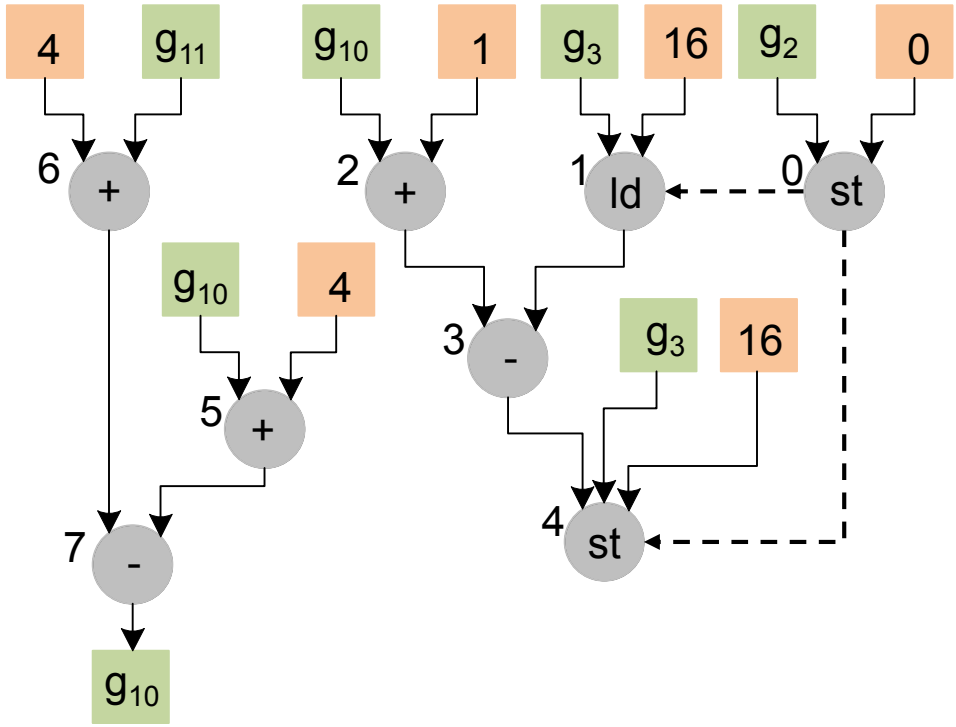
```
40848433   sub s0,s1,s0
0081a823   sw s0,16(gp)
00450413   addi s0,a0,4
00812023   sw s0,0(sp)
0101a403   lw s0,16(gp)
00150493   addi s1,a0,1
40848433   sub s0,s1,s0
0081a823   sw s0,16(gp)
00450413   addi s0,a0,4
00458493   addi s1,a1,4
40940533   sub a0,s0,s1
f60614e3   bnez a2,-32
00450413   addi s0,a0,4
00458493   addi s1,a1,4
40940533   sub a0,s0,s1
```

- Reconstruction du graphe de flot de contrôle
- Extraction des blocs

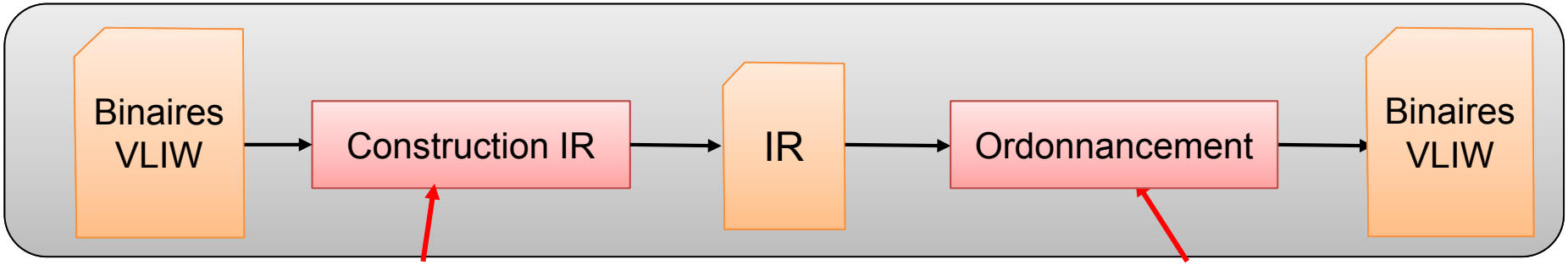
Représentation intermédiaire

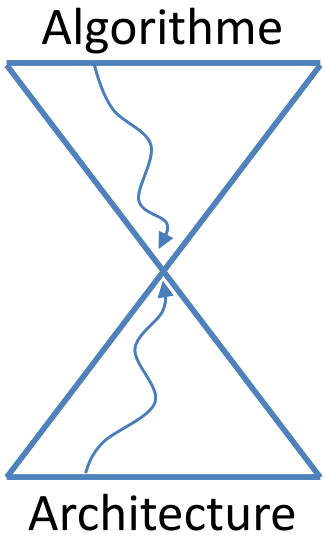
Binaire RISC-V

0 - 00812023 sw s0,0(sp)
1 - 0101a403 lw s0,16(gp)
2 - 00150493 addi s1,a0,1
3 - 40848433 sub s0,s1,s0
4 - 0081a823 sw s0,16(gp)
5 - 00450413 addi s0,a0,4
6 - 00458493 addi s1,a1,4
7 - 40940533 sub a0,s0,s1



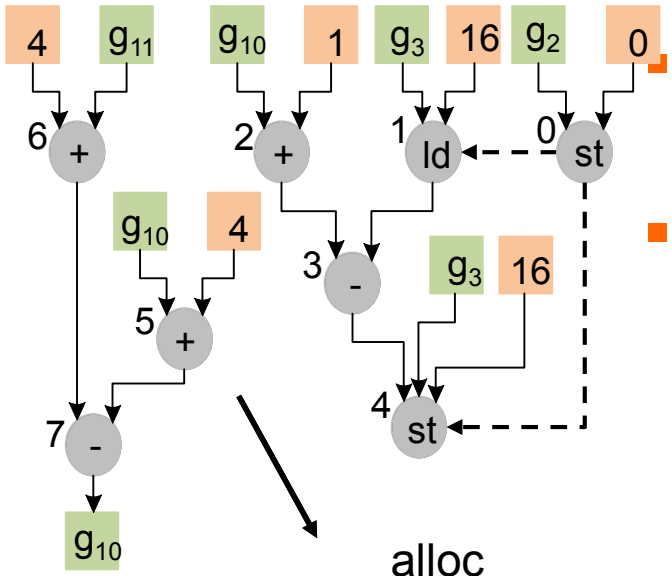
Chaîne de compilation en matériel



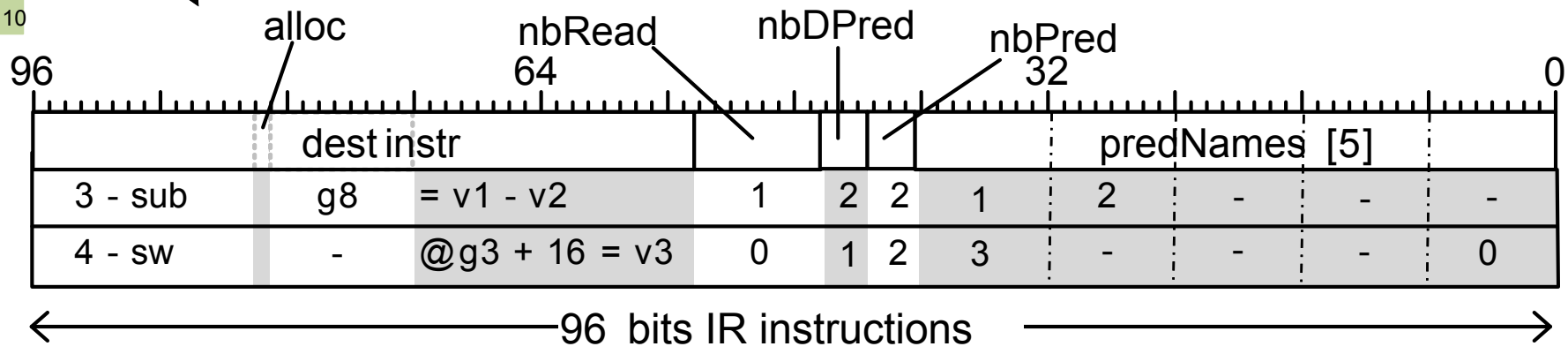


- **Trop complexe à développer en HDL**
- **Utilisation de la Synthèse de Haut-Niveau**
  - Déroulage / Pipeline de boucles
  - Partitionnement mémoire
  - Factorisation des accès mémoire
  - *Forwarding* explicite

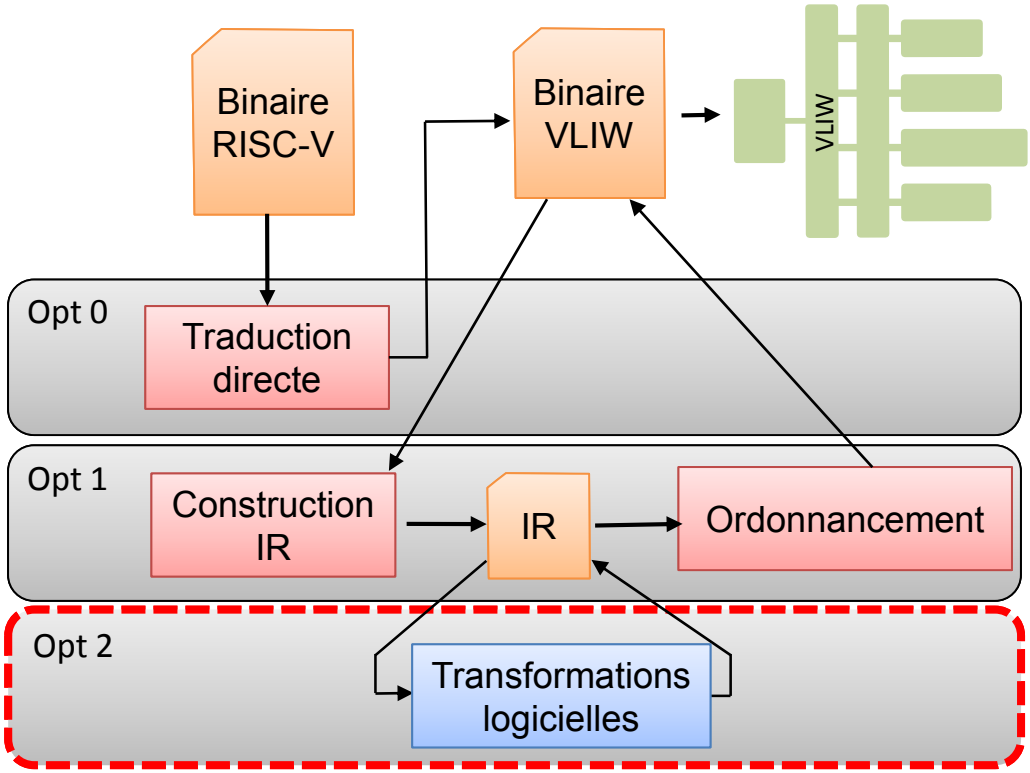
Représentation intermédiaire



- Graphe de flot de contrôle encodé dans l'IR
- Informations sur l'allocation de registres



Niveau d'optimisation 2



- **Optimisations logicielles**
- **Optimisation continue**
  - Reconfiguration dynamique du VLIW
  - Spéculation de dépendances mémoire

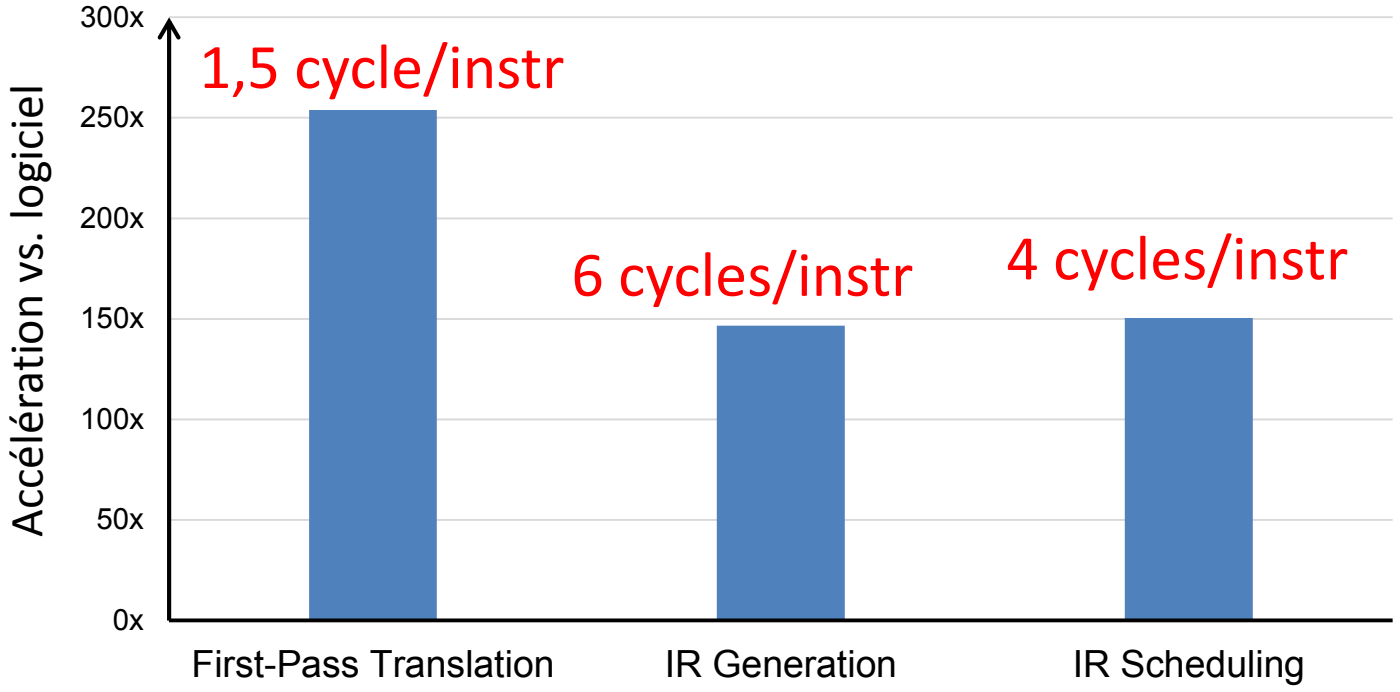
Etude expérimentale

- **Objectifs de l'étude :**
  - Impact des accélérateurs sur le coût de la DBT
  - Impact des accélérateurs sur la performance du système
  - Comparaison de notre approche avec OoO et inO
  - Coût matériel

- **Protocole expérimental**
  - Cible : **28nm,** fmax : **700MHz**
  - Performance : Gem5, Hybrid-DBT (cycle accurate)
  - Consommation : Simulation *gate-level* avec traces

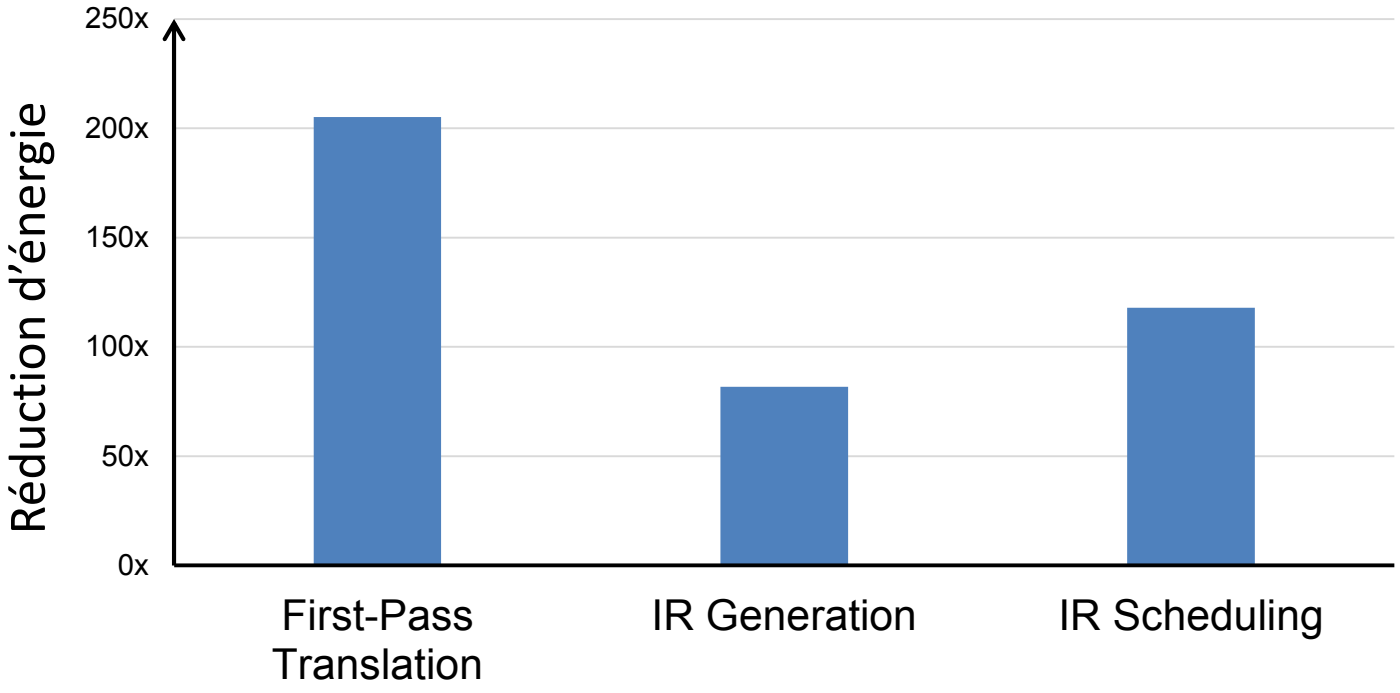
Performance des accélérateurs

- **Impact des accélérateurs sur le temps de DBT**
  - Facteur d'accélération comparé à une solution logicielle sur le processeur DBT



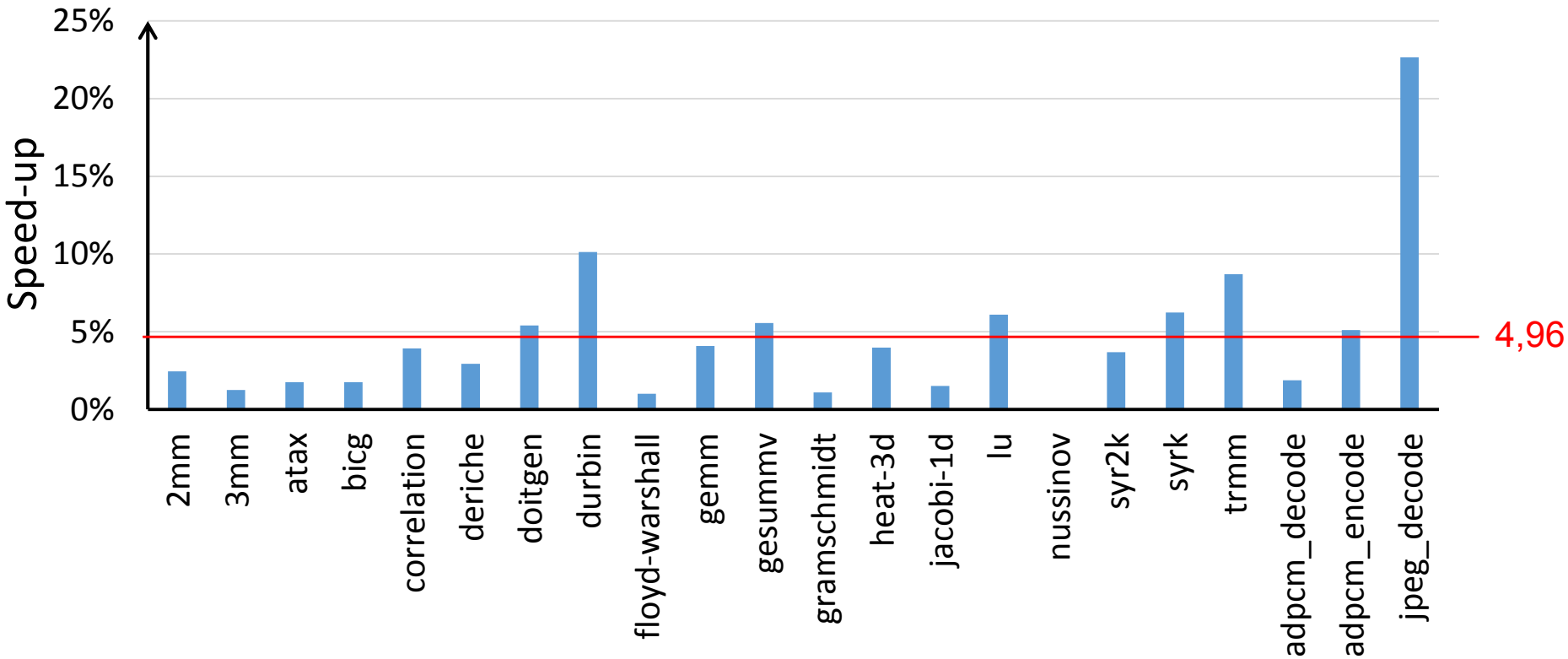
Consommation des accélérateurs

- **Impact des accélérateurs sur l'énergie de la DBT**
  - Facteur de réduction d'énergie comparé à une solution logicielle sur le processeur DBT



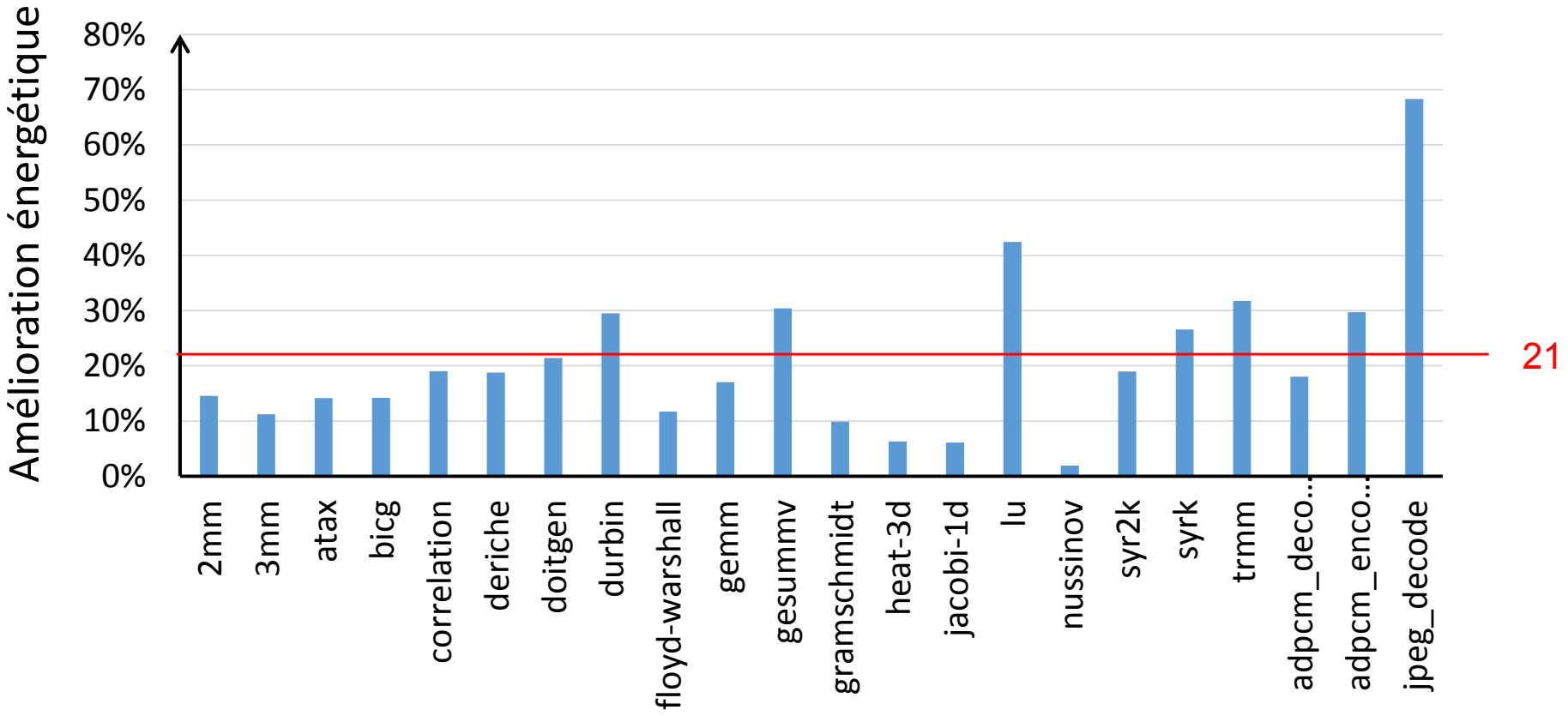
Intérêt de l'accélération matérielle

- **Impact des accélérateurs sur le temps d'exécution d'une application**



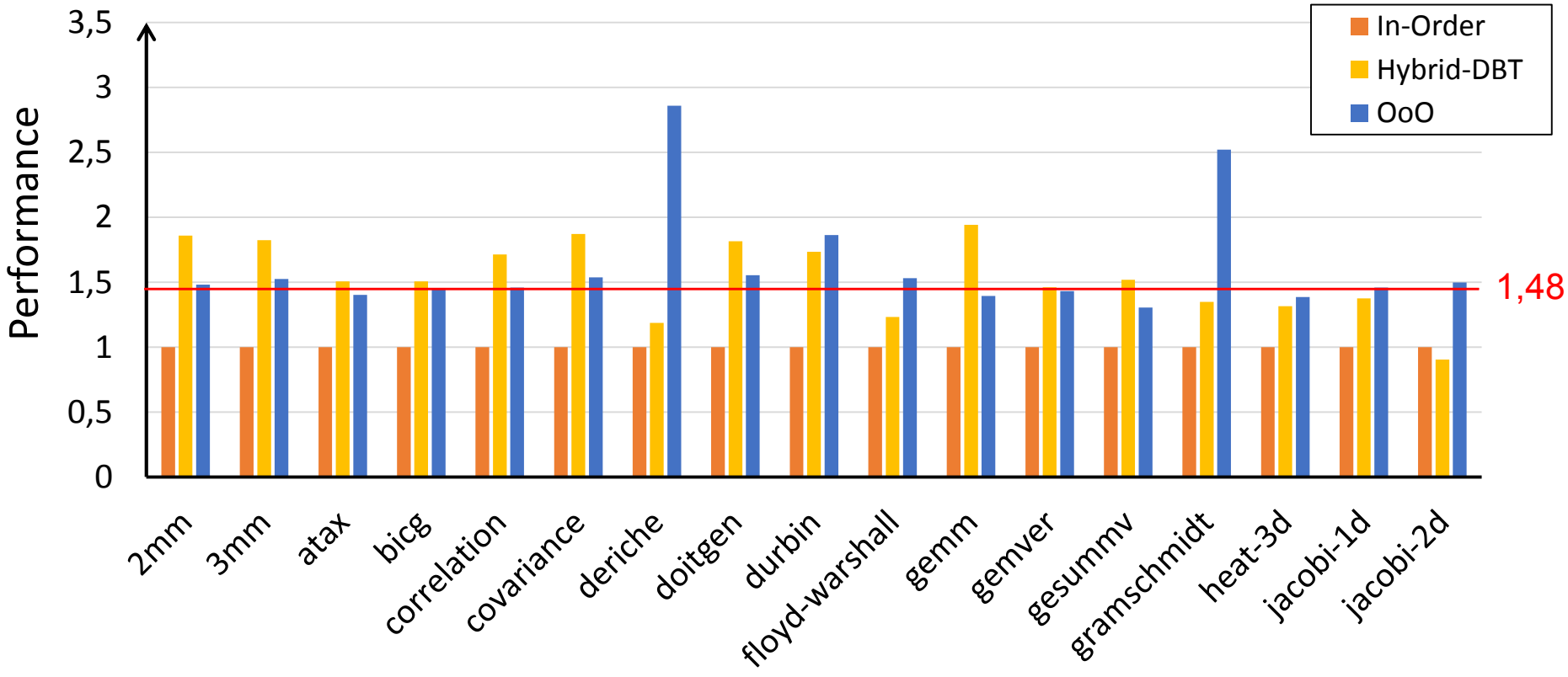
Intérêt de l'accélération matérielle

- Impact des accélérateurs sur l'**énergie consommée** par une application



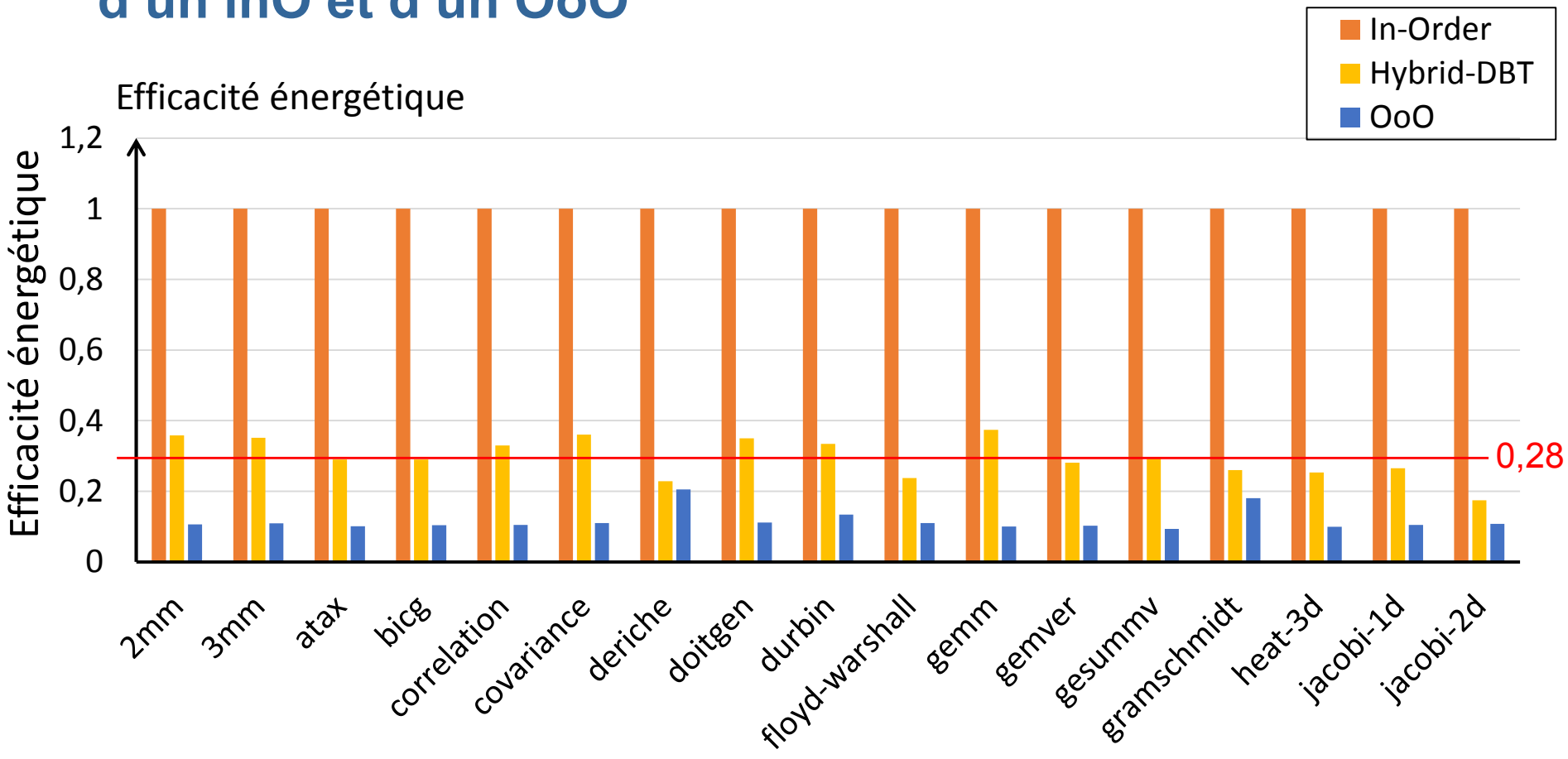
Performance du système

- **Comparaison des performances de Hybrid-DBT, d'un inO et d'un OoO**



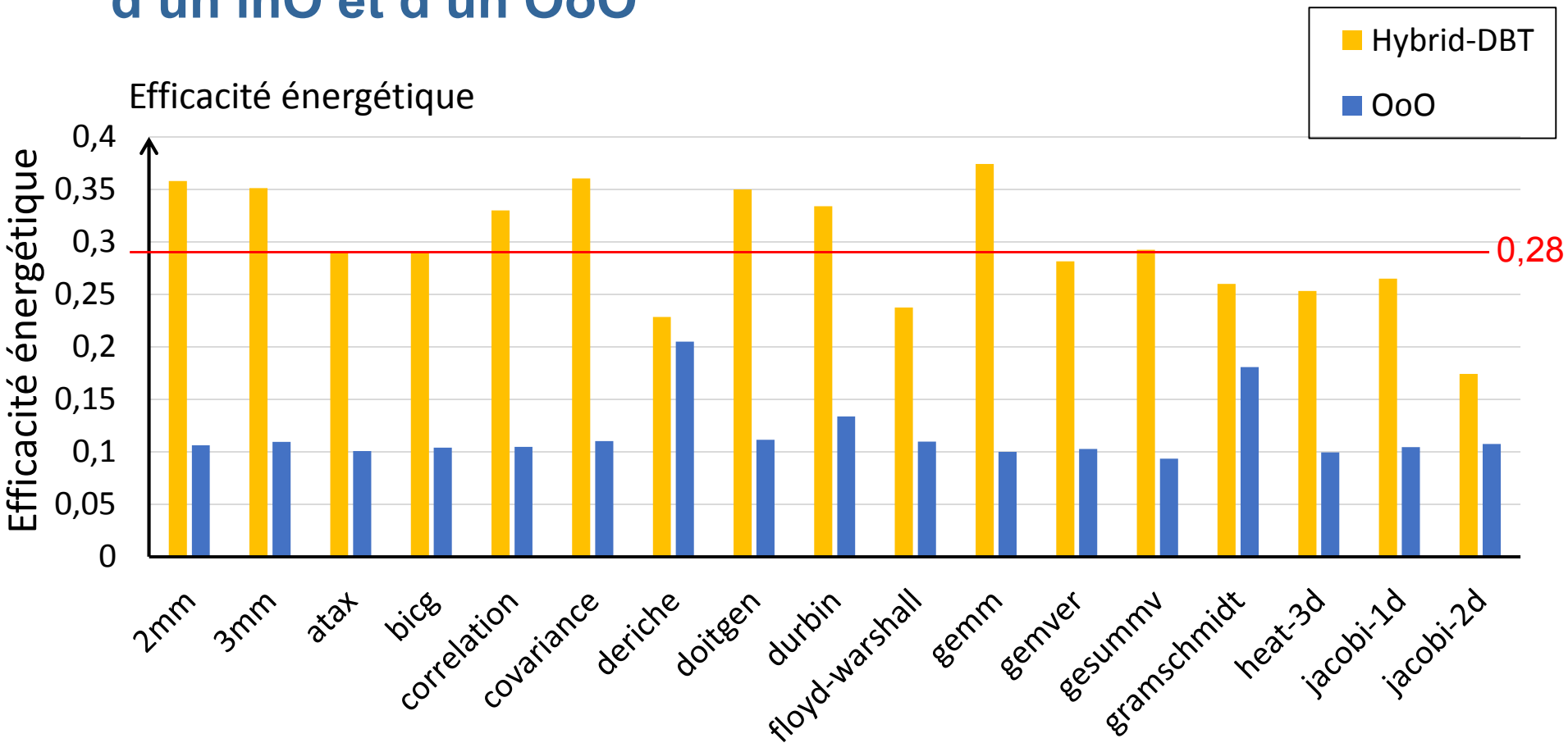
Efficacité énergétique du système

- **Comparaison de l'efficacité énergétique de Hybrid-DBT, d'un inO et d'un OoO**



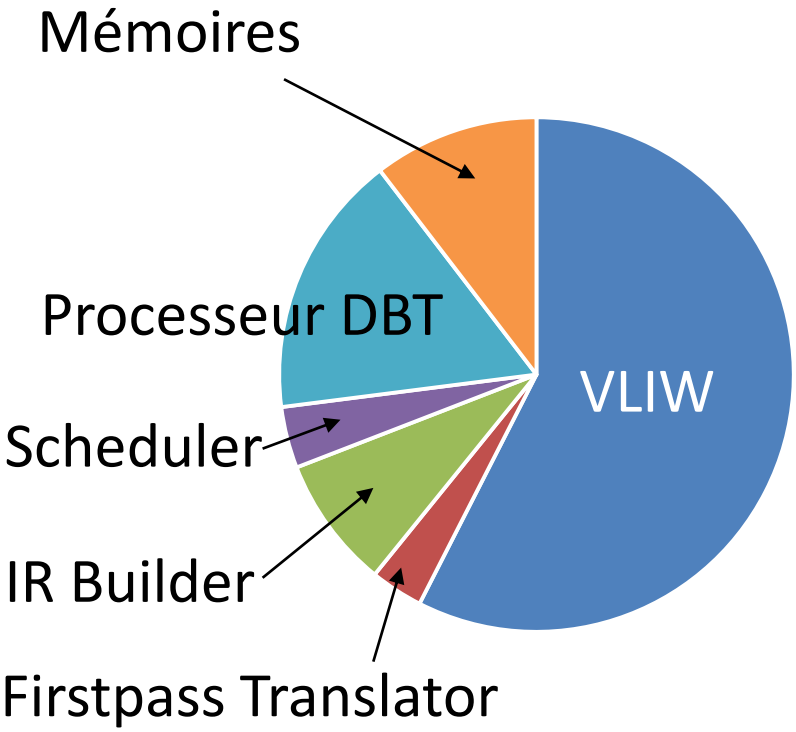
Efficacité énergétique du système

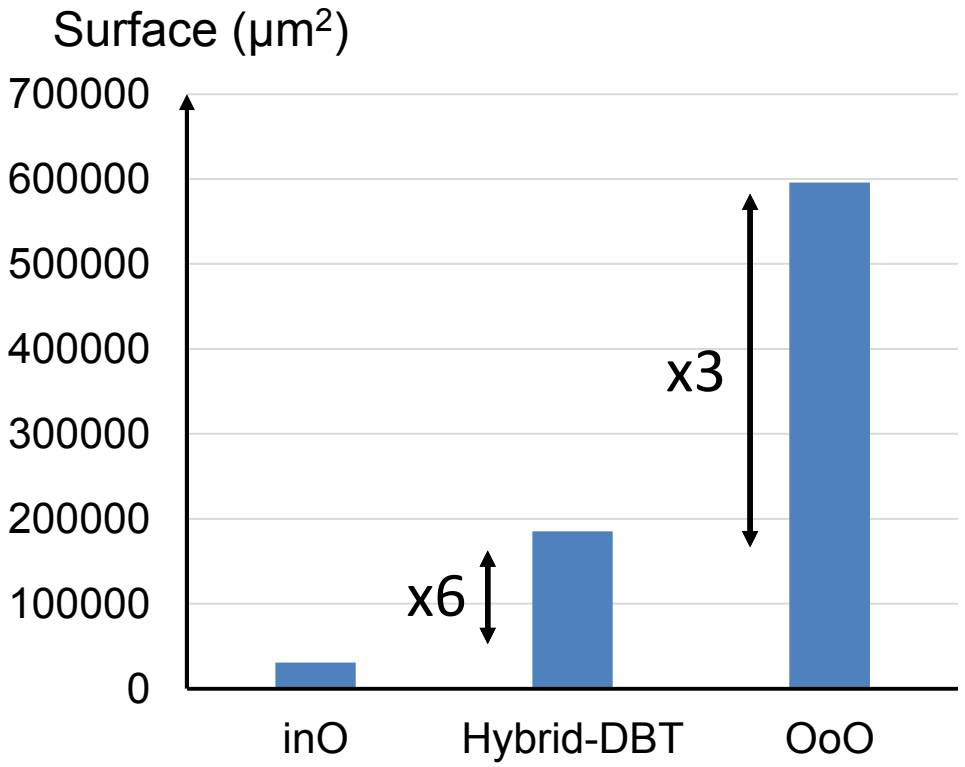
- **Comparaison de l'efficacité énergétique de Hybrid-DBT, d'un inO et d'un OoO**



Surcoût matériel

- **Coût matériel de notre approche**





Plan de la présentation

- **Etat de l'art**
  - Architecture des processeurs
  - Traduction Dynamique de Binaires
- **Accélération matérielle de la DBT**
- **Conclusion**

Conclusion

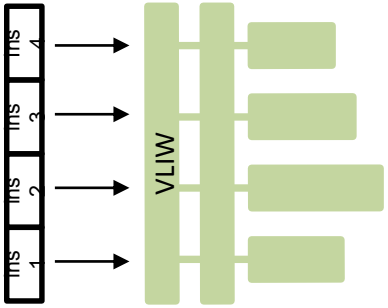
- **Conception d'une *HW/SW Co-Designed Machine***
  - Mise en place du flot d'optimisation
  - Définition de l'IR
  - Conception conjointe de quatre accélérateurs
  - Conception du processeur VLIW en HLS

- **Optimisation continue**
  - Spécialisation dynamique du VLIW
  - Spéculation de dépendances mémoire
    - Conception d'une LSQ partitionnée
    - Gestion de la spéculation dans l'ordonnanceur

Perspectives – Construction de traces

- **Réduction du coût de la construction de traces**
  - Mécanisme similaire aux prédicteurs de branchements
  - Ordonnanceur avec support pour la spéculation

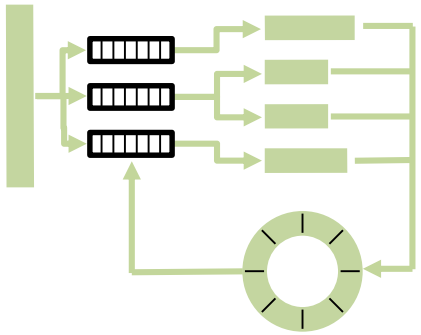
Dynamicité



- **Hybrid-DBT**

Réutilisation des traces

& génération rapide



- **Processeur VLIW :**

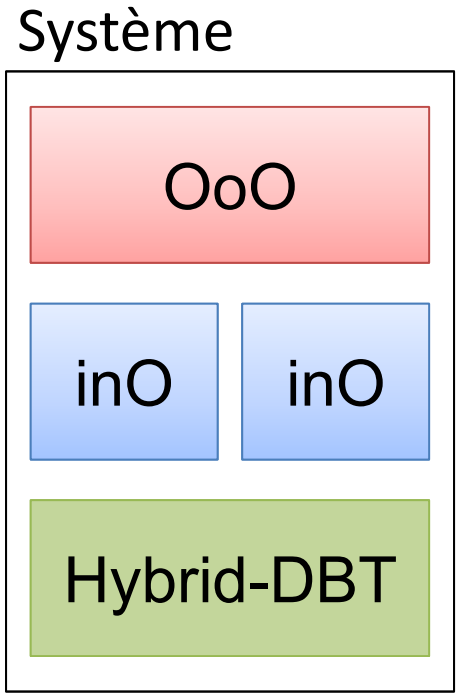
Trace déterminée à la compilation

- **Processeur OoO :**

Trace recalculée à chaque exécution

Perspectives – Système hétérogène

- **Intégration dans un système hétérogène**
  - Processeurs OoO
  - Processeurs inO
  - Hybrid-DBT
- **Politique de migration de tâches**
- **Utilisation des compteurs de performances**



Perspectives - Sécurité

- **Spéculation lors des phases d'optimisation**
- **Sensible aux failles spectres**
  - Construction de traces : *Bounds Check Bypass* (v1)
  - Spéculation mémoire : *Speculative Store Bypass* (v4)
- **Modification du flot de DBT pour prévenir ces attaques**