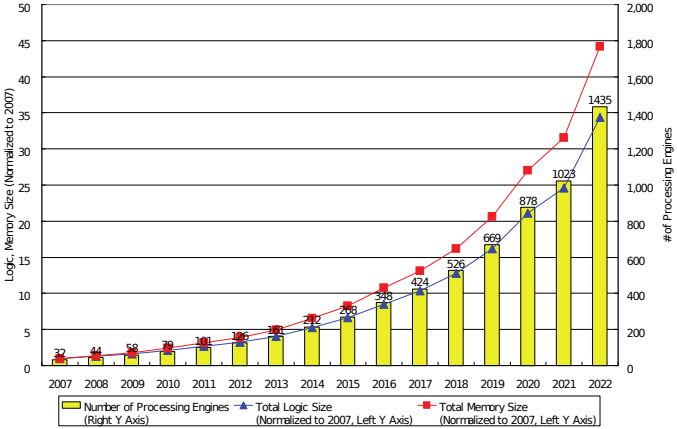
Simulation matériel/logiciel efficace et nouvelles techniques de virtualisation

Frédéric Pétrot, Laboratoire TIMA

tima.imag.fr/sls/people/petrot

frederic.petrot@univ-grenoble-alpes.fr

Tendances de l'intégration silicium



Feuille de route de l'ITRS [1] : nombre cœurs dans systèmes embarqués

1. International Roadmap on Semiconductors

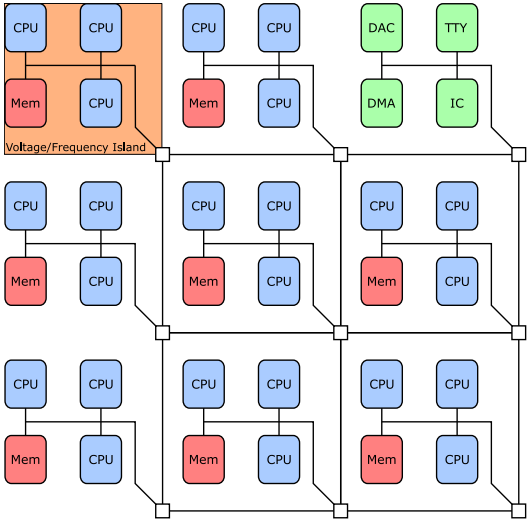
Architectures typiques de cette décennie

Caractéristiques

- massivement parallèle
- homogène ou hétérogène
- avec des hiérarchies mémoire souvent complexes

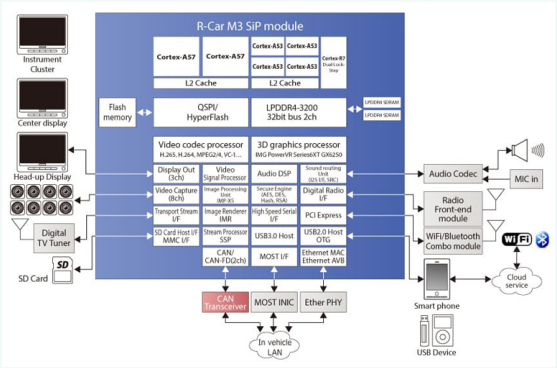
Propriétés

- flexible à très flexible
- fortes contraintes de coût
- fortes contraintes de temps de mise sur le marché

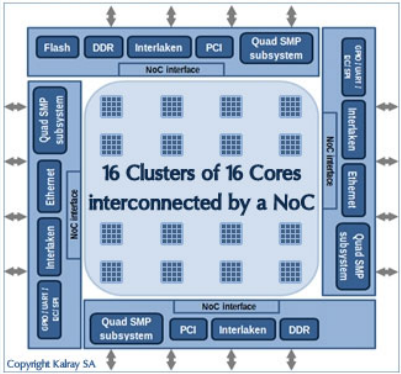


Architectures

Deux exemples industriels actuels



Renesas R-Car M3



Kalray

Sondage, scannez le qr code !

Plan

1. Introduction

2. Modélisation en vue de la simulation

3. Simulation matériel/logiciel

4. Co-simulation

5. Acteurs

6. À retenir

Simulation

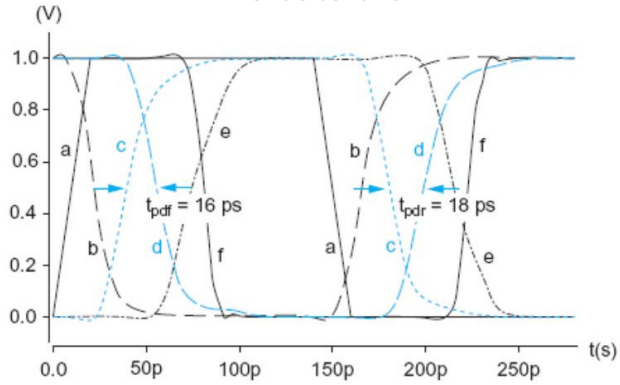
Une technologie qui couvre tous les aspects de la conception et de la validation des systèmes électroniques

Sujet de cette présentation

- simulation de systèmes électroniques numériques (et mixtes)
  - interconnectant de nombreux composants matériels
  - contenant un ou plusieurs processeurs
  - executant du logiciel
- plus abstrait que le niveau transfert de registres (RTL)
- focalisé sur exécution rapide du logiciel sur modèle de simulation du matériel

Simulation : niveaux d'abstraction

Monde continu



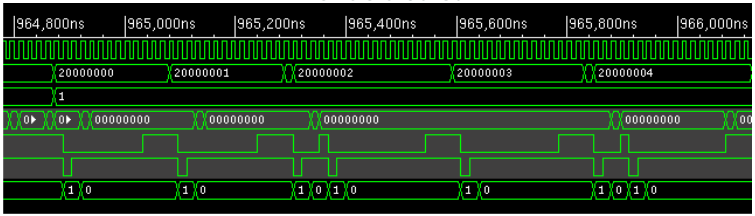
Résolution d'équations différentielles

(source *CMOS VLSI Design*, N. Weste)

Modélisation physique, pas (*step*) de simulation variables, extrêmement lent

Simulation : niveaux d'abstraction

Monde discret



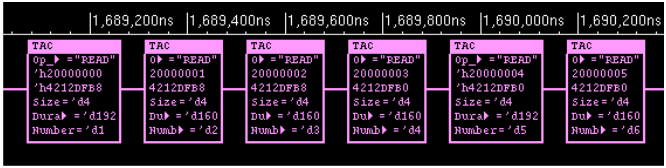
RTL (register transfer level), événements discrets, synthétisable

Communication par signal, pas fixe, δ-cycle après δ-cycle, très lent

CABA (cycle accurate bit accurate), à visée de simulation uniquement

Communication par signal, pas fixe, cycle après cycle, lent

Simulation : niveaux d'abstraction

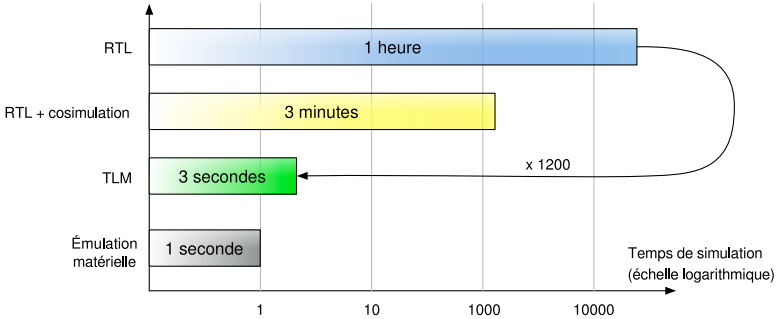


TLM (transaction level model), événements discrets, simulable

(source *STMicroelectronics*)

Communication par transactions gros grain, événements rares, rapide

Simulation : niveaux d'abstraction



Temps d'encodage puis décodage d'une image MPEG 4

(source *STMicroelectronics*)

Gain en temps de modélisation	
RTL	1
CABA	3
TLM	10

Motivations

Approches tournées vers le développement logiciel

- en avance de phase vis-à-vis disponibilité SoC ou PCB
- participe de la qualité logicielle car facilite les tests de non régression
- facilite le débug grâce à une grande observabilité

Et sa vérification fonctionnelle

En particulier du logiciel lié au matériel

- *firmware*, logiciels *bare metal*
- *drivers* des systèmes d'exploitation

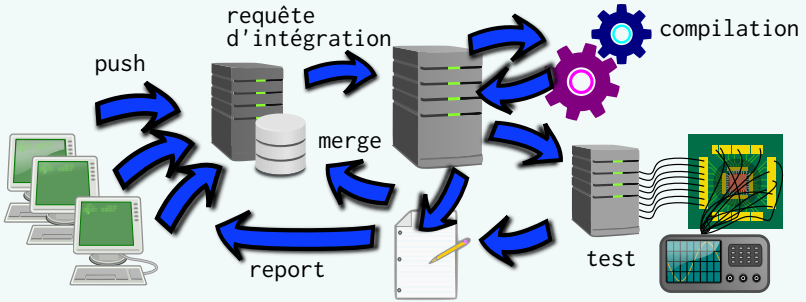
Mais aussi applications parallèles exploitant les plates-formes

Boot de Linux ARM11MPCORE	
hôte	2 s
TLM+DBT	10 s
CABA+ISS	3.3 h

(source TIMA)

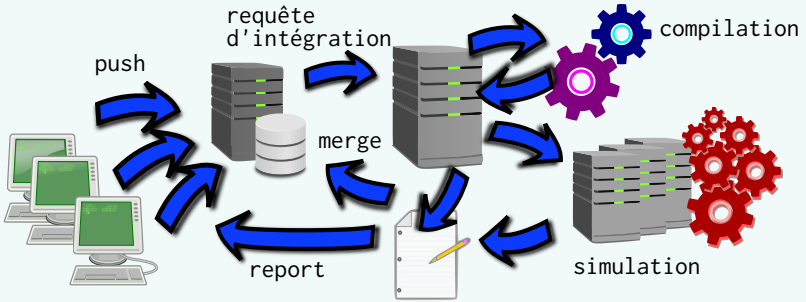
Motivations

Exemple emblématique : intégration continue



Motivations

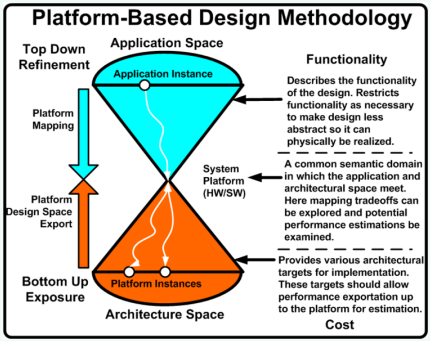
Exemple emblématique : intégration continue



Motivations

Exploration de l'espace de conception matériel et/ou logiciel

- spécification exécutable : sert de référence
- facilité de déploiement et de mise en œuvre
- support aux choix de dimensionnement
  - taille et nombre de bus
  - famille et nombre de processeurs
  - niveau de parallélisations des programmes
- permet l'injection précise de fautes et l'analyse des impacts
- remplace les feuilles excel des experts
- rationalise le processus de décision



(source A. Sangiovanni-Vincentelli, Berkeley)

Propriétés non fonctionnelles

Estimation réaliste délicate

- temps
- puissance consommée
- température

« Truth . . . is much too complicated to allow anything but approximations », John Von Neumann, 1947
« All models are wrong ; some models are useful », George E. P. Box, 2005

Vitesse de simulation

- précision influence grandement la vitesse de simulation
  modélisation des caches, des bus, etc, ralentissent fortement
- vitesse de simulation très critique pour mise au point du logiciel

Plan

1. Introduction

2. Modélisation en vue de la simulation

3. Simulation matériel/logiciel

4. Co-simulation

5. Acteurs

6. À retenir

Simulation materiel/logiciel

Clarification

hôte : machine sur laquelle la simulation s'exécute

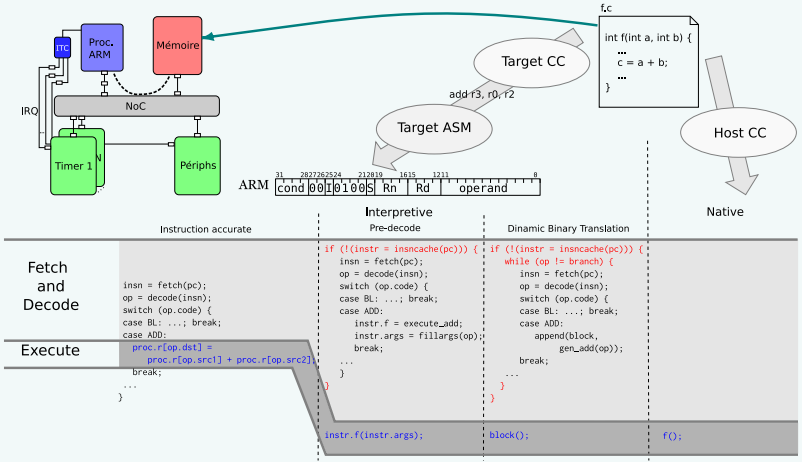
cible : machine simulée

Hypothèses

- simulation repose sur évènements pour exécution du matériel
  - haut niveau d'abstraction pour vitesse compatible avec developpement logiciel
- logiciel à une place à part
  - exécuté par un modèle de processeur

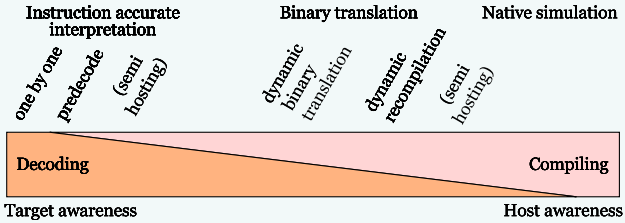
Panel des techniques

Panel des techniques de simulation du logiciel



Panel des techniques

Sur un axe de l'interprétation complète à l'exécution native

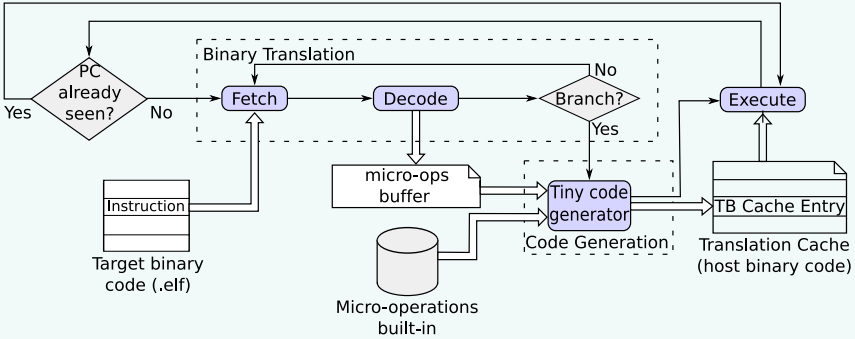


Caractéristiques gros grain des approches de simulation de logiciel

	sim. speed	simulation accuracy		development time		full soft. exec.
		ins. count	time	first	reuse	
IA	−−	++	++	+	+	yes
Predecode	−	++	++	−	+	yes
BT	++	−−	−−	−−	−	yes
Native	+ + +	−−	−−	−−	++	no

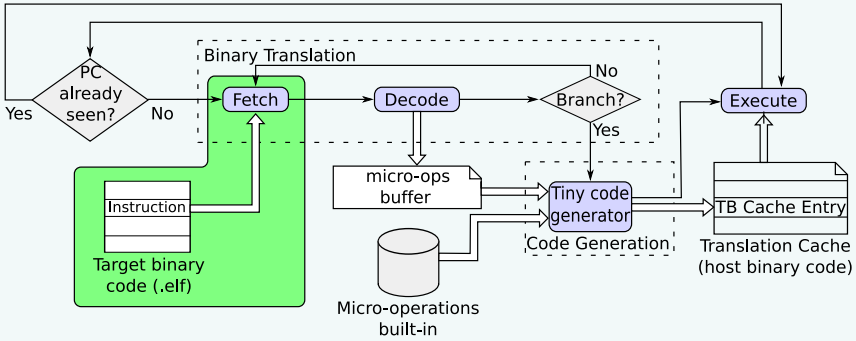
Mécanisme de simulation

Processus d'interprétation



Mécanisme de simulation

Processus d'interprétation

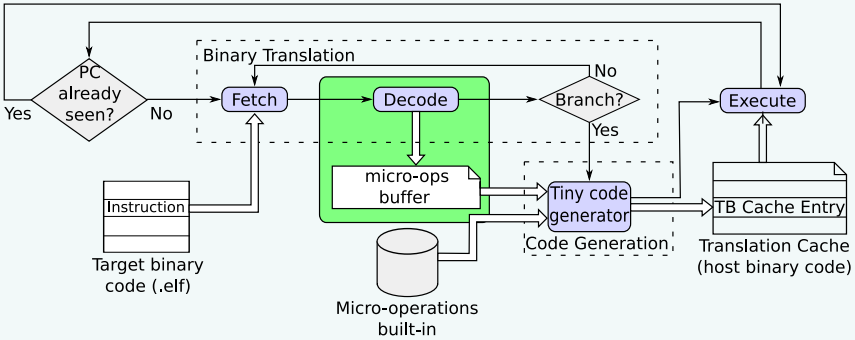


Exemple de génération de code

18 target_instrX

Mécanisme de simulation

Processus d'interprétation



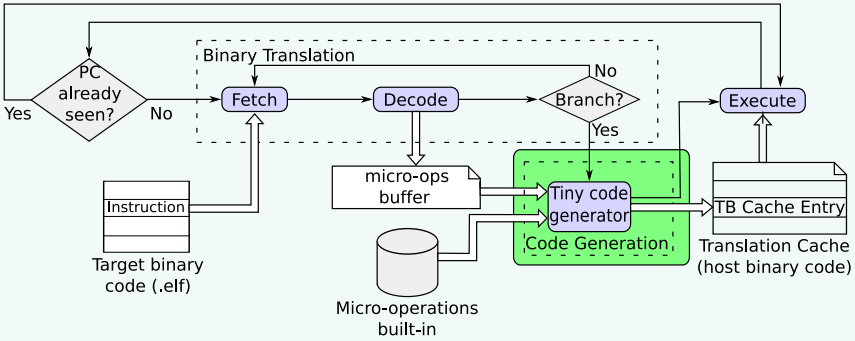
Exemple de génération de code

```
18  target_instrX          micro-op1_instrX

                           micro-op2_instrX
```

Mécanisme de simulation

Processus d'interprétation



Exemple de génération de code

```
18  target_instrX       micro-op1_instrX    host_instr1_micro-op1_instrX
                                            host_instr2_micro-op1_instrX
                                            host_instr3_micro-op1_instrX
                        micro-op2_instrX    host_instr1_micro-op2_instrX
                                            host_instr2_micro-op2_instrX
```

Mécanisme de simulation

Conséquences

- instructions exécutées par « blocs » en temps zéro
- exécution directe sur hôte avec chaînage des blocs
  impossible au simulateur de reprendre la main si pas prévu

$\Rightarrow$ synchronisation avec les modèles matériels à définir

Points de synchonisation

- opérations d'entrée/sortie (accès aux périphériques)
- gestion des interruptions de l'hôte
- après un nombre d'instructions prédéfini
  par insertion de code *ad-hoc* lors de la génération

Annotation du code généré : principes

Objectifs

Ajouter des synchronisations, quantifier, par ex., le temps d'exéction sur la cible

Insertion de micro-operations

- accumule un nombre de cycles fonction de la spécification
  $\Rightarrow$ prise en compte des dépendances entre registres, données, et contrôle
- modélisation micro-architecture (caches, prédicteurs de branchements, ...)

Exemple d'annotations

Instr address	Target code	Original translation	Annotated translation	Annotated generated code
addr_instr1	target_instrX	micro-op1_instrX	micro-op1_instrX	host_instr1_micro-op1_instrX
				host_instr2_micro-op1_instrX
				host_instr3_micro-op1_instrX
		micro-op2_instrX	micro-op_annotation	host_instr1_micro-op_annotation
				host_instr2_micro-op_annotation
			micro-op2_instrX	host_instr1_micro-op2_instrX

QEMU+SystemC/TLM : solution opérationnelle

QEMU

- traducteur binaire dynamique fiable et rapide (récemment multi-thread)
- simule de multiples architectures cibles : x86, ARMvX, Sparc, PPC, MIPS, RISCV, ...
- sur de multiples architectures hôtes : x86, ARMv7/v8, Sparc, PPC, MIPS
- de 10 à 30 instructions exécutées par instruction simulée

SystemC

- standard IEEE 1666-2011
- classes C++ permettant l'expression de la structure et de la concurrence
- à différent niveaux d'abstraction : TLM et CABA en particulier

Plan

1. Introduction

2. Modélisation en vue de la simulation

3. Simulation matériel/logiciel

4. Co-simulation

5. Acteurs

6. À retenir

Co-simulation avec autre solveurs

Monde du système au sens large : Matlab/Simulink, SystemC-AMS

Co-simulation avec *fixed-step* Matlab/Simulink

Plate-forme virtuelle SystemC appelé par Matlab/Simulink

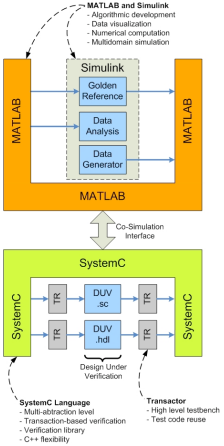
- support du *legacy*
- favorise l'adoption et la migration vers SystemC
- lent à très lent

Simulation SystemC-AMS

Modèles de calcul

- *timed data flow* : signal échantillonné à pas fixe
- *linear signal flow* : continu avec résultats discrets
- *electrical linear networks* : lois de Kirchoff

$\Rightarrow$ Problème très difficile si pris dans sa généralité



(source J.-F. Boland, Univ. Mc Gill, QC)

Acteurs du domaine

Membres du consortium Accelera

(source `http://www.accellera.org/about/members`)

Outils/PME

VLAB	Australie
Simics	USA
OVP	UK
QBox	France
Coside	Allemagne
...	...

Labos universitaires français

CEA-LIST - Saclay
IRISA - Rennes
Labsticc - Université de Bretagne Sud, Lannion
LIP6 - Sorbonne Université, Paris
LIRMM - Université de Montpellier, Montpellier
TIMA - Univ. Grenoble Alpes, Grenoble
Verimag - Univ. Grenoble Alpes, Grenoble
...

Prototypage virtuel à TIMA

Développement de la technologie

- sujet majeur depuis le milieu des années 90
- première démonstration intégration QEMU+SystemC en 2009
- insertion d'annotations non fonctionnelles pour métriques et synchronisations
- production et analyse de traces
- support SIMD et VLIW
- optimisation vitesse de simulation

$\Rightarrow$ Mise en open-source de la technologie Rabbits

Industrialisation

- création de la société

en 2016
- rachat d'Antfield par GreenSoCs en novembre 2017

Plan

1. Introduction

2. Modélisation en vue de la simulation

3. Simulation matériel/logiciel

4. Co-simulation

5. Acteurs

6. À retenir

Utilisation de plates-formes virtuelles : difficultés

Marché de niche, intérêt grandissant, mais difficile de « sauter le pas »

Utilité des modèles

Encore sujette à caution chez de nombreux industriels

Disponibilité des modèles

- très peu de modèles disponibles comparé au RTL
- modèles essentiellement internes aux compagnies

Compatibilité des modèles

- SystemC/TLM 2.0 permet l'abstraction, mais contenu échangé souvent personnalisé

Qualité des modèles

- difficile à établir sur la fonction (specs papier $\ggg$ 10 pages par IP)
- extrêmement difficile à garantir sur les métriques

Plates-formes virtuelles : pour conclure

Gains

Validation fonctionnelle d'architectures électronique/logiciel

- permet l'essentiel du développement en avance de phase
  ex. : logiciel quasi-prêt et débuggé lors de l'arrivée circuit/carte
- temps de *bring-up* des circuits/cartes réduits significativement
  ex. : d'une semaine à une journée
- modèles participent de la qualité du logiciel
  vérifiant par ex. des valeurs min ou max sur sources de tension ou horloges
- tests de non-régression facilement automatisables
  ex. : intégration continue basée sur plates-formes virtuelles

Remerciements

Merci aux différents collègues ayant travaillé sur ce sujet :

Aimen Bouchhima (ENSI Tunis),
Clément Deschamps (GreenSoCs, précédemment Antfield),
Antoine Faravelon (PhD UGA),
Nicolas Fournel (Nov'in),
Patrice Gerin (Kalray),
Marius Gligor (Kalray),
Omayma Matoussi (CEA),
Mian-Muhammad Hamayun (Univ. Islamabad),
Luc Michel (GreenSoCs, précédemment Antfield),
Hao Shen (Synopsys, CA).

Sondage, scannez le qr code !