



Prototypage virtuel de système sur puce pour une simulation rapide et fidèle

Séminaire Collège de France, 29 Janvier 2014

Laurent Maillet-Contoz

STMicroelectronics

Laurent.Maillet-Contoz@st.com

Matthieu Moy

Grenoble INP – Verimag

Matthieu.Moy@grenoble-inp.fr





Prototypage virtuel de système sur puce pour une simulation rapide et fidèle

Séminaire Collège de France, 29 Janvier 2014

Laurent Maillet-Contoz

System Design Solutions – System Level Modeling

laurent.maillet-contoz@st.com

- Contexte industriel
- Prototypage virtuel de système sur puce
- Emergence de nouveaux besoins

Un exemple de produit

4

- **ARM® Cortex™-A53 and Cortex™-A57: processeur 64-bit**
- **Haute performance and fiabilité**
- **Efficacité énergétique exceptionnelle**
- **Codage video 10-bit ultra efficace (HEVC)**
- **Ultra Haute Définition à 60 frames par seconde**
- **Systèmes domestiques connectés ultra-haute bande passante (par ex 8 flux HD)**
- **HDCP 2.2 (sécurité)**
- **HDMI 2.0: 6 GBps / canal**
- **DisplayPort 1.2**

The screenshot shows the STMicroelectronics website with a press release titled "Groundbreaking Architecture from STMicroelectronics Leads Industry Transition to 64-bit Computing for the Digital Home". The page includes a navigation menu, a sidebar with company information, and a main content area with a list of bullet points and a paragraph. The right sidebar features "Upcoming Events" and "Featured Products".

Company Information
Media Center
Investor Relations
Events
Innovation and Technology
Sustainability
Customer Satisfaction
Careers

Groundbreaking Architecture from STMicroelectronics Leads Industry Transition to 64-bit Computing for the Digital Home

- Builds on successful deployment of current ST ARM-based SoCs, including its 'Cannes,' 'Monaco,' and 'Alicante' product families
- Leverages latest 64-bit ARM cores to support the increasing performance and memory requirements of future SoCs

Geneva / 06 janv. 2014

STMicroelectronics (NYSE: STM), a global semiconductor leader serving customers across the spectrum of electronics applications and a leading manufacturer of High-Definition (HD) and Ultra-High-Definition (UHD) set-top-box ICs, has released details of its innovative STi8K™ architecture addressing future Systems-on-Chips (SoCs) for the Digital Home.

Based on the latest ARMv8-A cores, STi8K™ SoCs will further enhance the available Cannes/STiH3 (FHD/UHD HEVC Client-Box), Monaco/STiH4 (FHD/UHD HEVC Server/Gateways), and Alicante/STiD12 (DOCSIS®) product families, providing, in combination with them, a smooth and future-proof industry transition from 32-bit to 64-bit computing.

"The consumer industry has initiated the transition from 32- to 64-bit computing in the mobile market," said [Gian Luca Bertino](#), Executive Vice President and General Manager Digital Convergence Group, STMicroelectronics. "New upcoming, highly demanding technologies, such as DOCSIS 3.1 or full-featured 10-bit High-Efficiency Video Coding (HEVC) in Ultra High Definition at 60 frames per second, will drive this same transition in

Upcoming Events

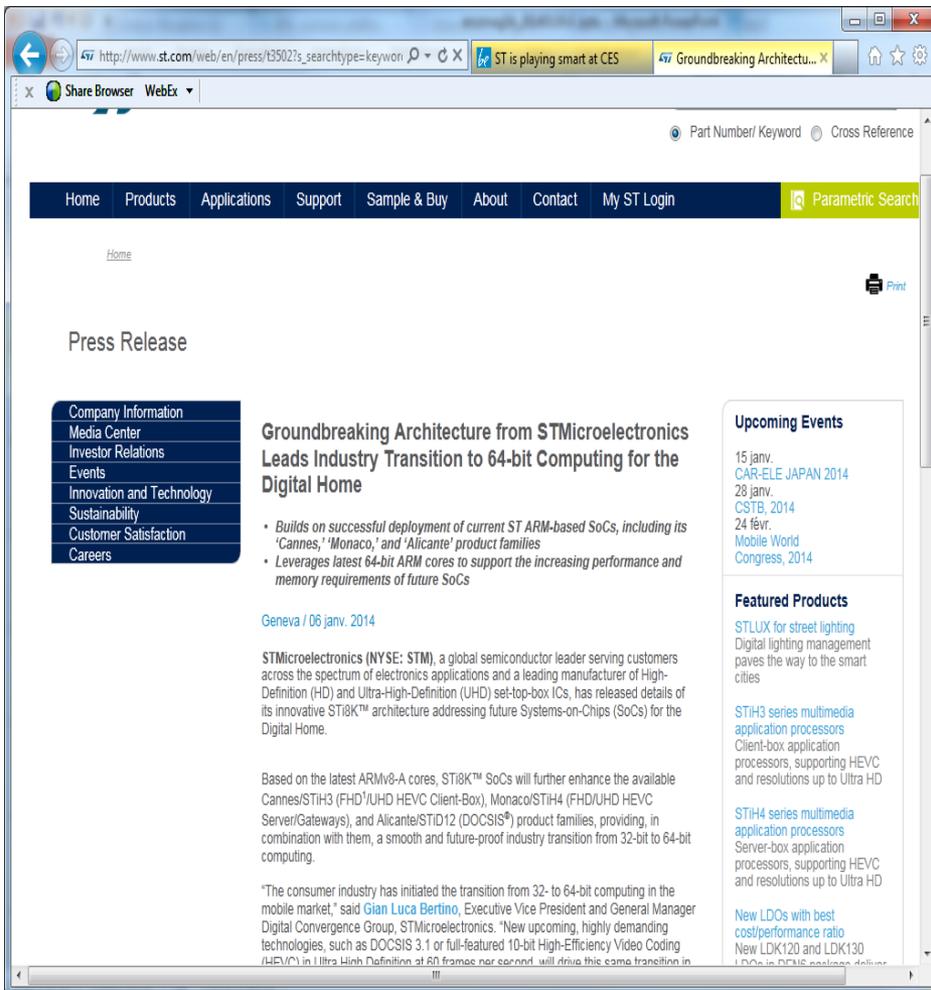
- 15 janv. CAR-ELE JAPAN 2014
- 28 janv. CSTB, 2014
- 24 févr. Mobile World Congress, 2014

Featured Products

- STLUX for street lighting
Digital lighting management paves the way to the smart cities
- STiH3 series multimedia application processors
Client-box application processors, supporting HEVC and resolutions up to Ultra HD
- STiH4 series multimedia application processors
Server-box application processors, supporting HEVC and resolutions up to Ultra HD
- New LDCs with best cost/performance ratio
New LDK120 and LDK130 LDCs in CMOS technology deliver

Un exemple de produit

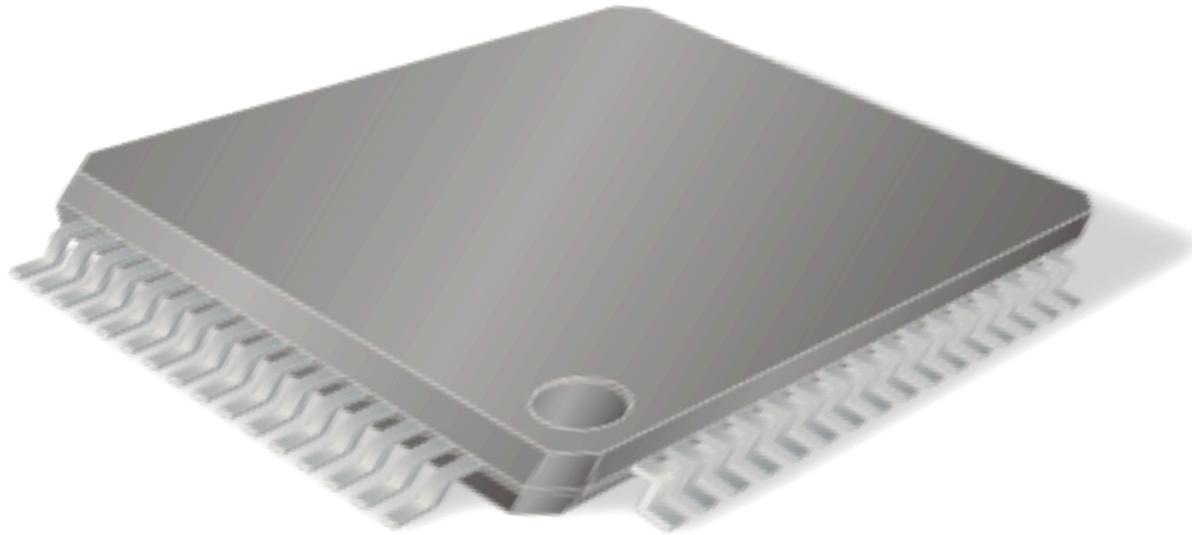
4



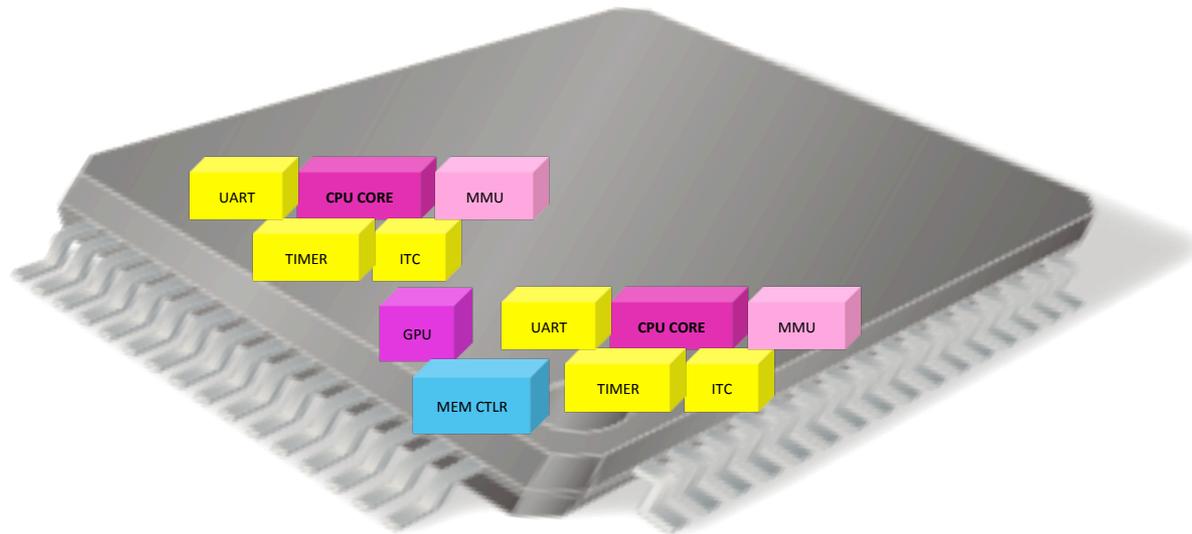
- **ARM® Cortex™-A53 and Cortex™-A57: processeur 64-bit**
- **Haute performance and fiabilité**
- **Efficacité énergétique exceptionnelle**
- **Codage video 10-bit ultra efficace (HEVC)**
- **Ultra Haute Définition à 60 frames par seconde**
- **Systèmes domestiques connectés ultra-haute bande passante (par ex 8 flux HD)**
- **HDCP 2.2 (sécurité)**
- **HDMI 2.0: 6 GBps / canal**
- **DisplayPort 1.2**

- **Linux**
- **OpenGL**
- **Video4Linux**
- **Gstreamer**
- **Advanced Linux Sound Architecture (Alsa)**
- **Pilotes bas niveau**
- **Logiciel enfoui dans les sous-systèmes**

Les produits à développer : des systèmes sur puce complexes



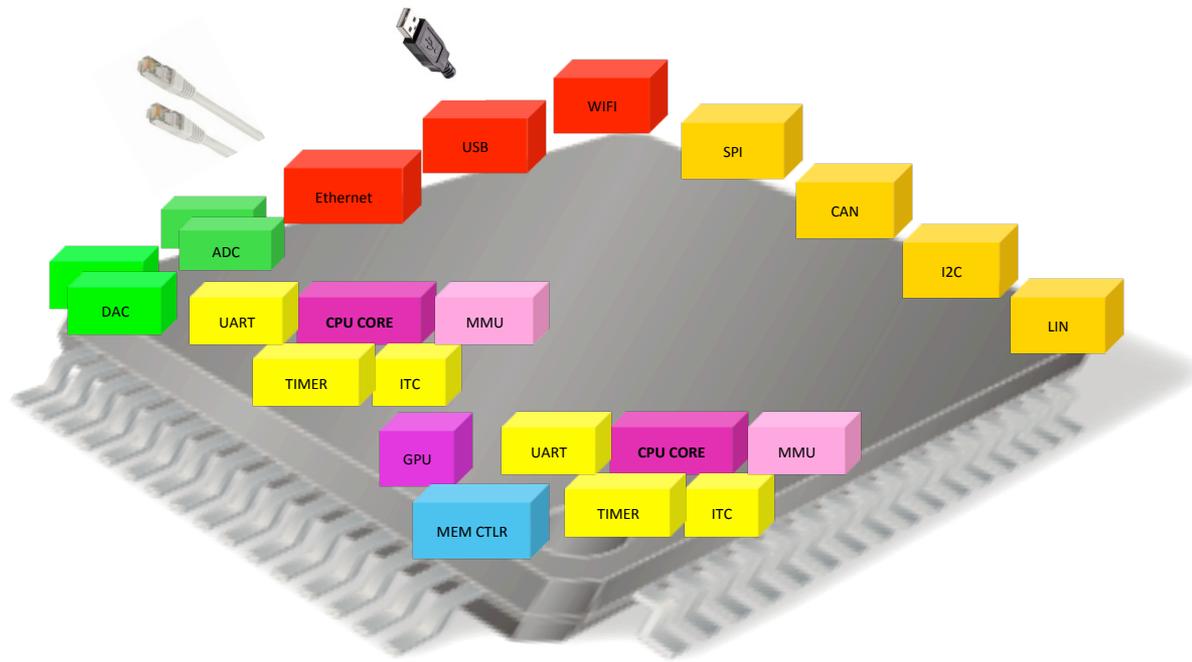
Les produits à développer : des systèmes sur puce complexes



Multi-processeurs

Les produits à développer : des systèmes sur puce complexes

5

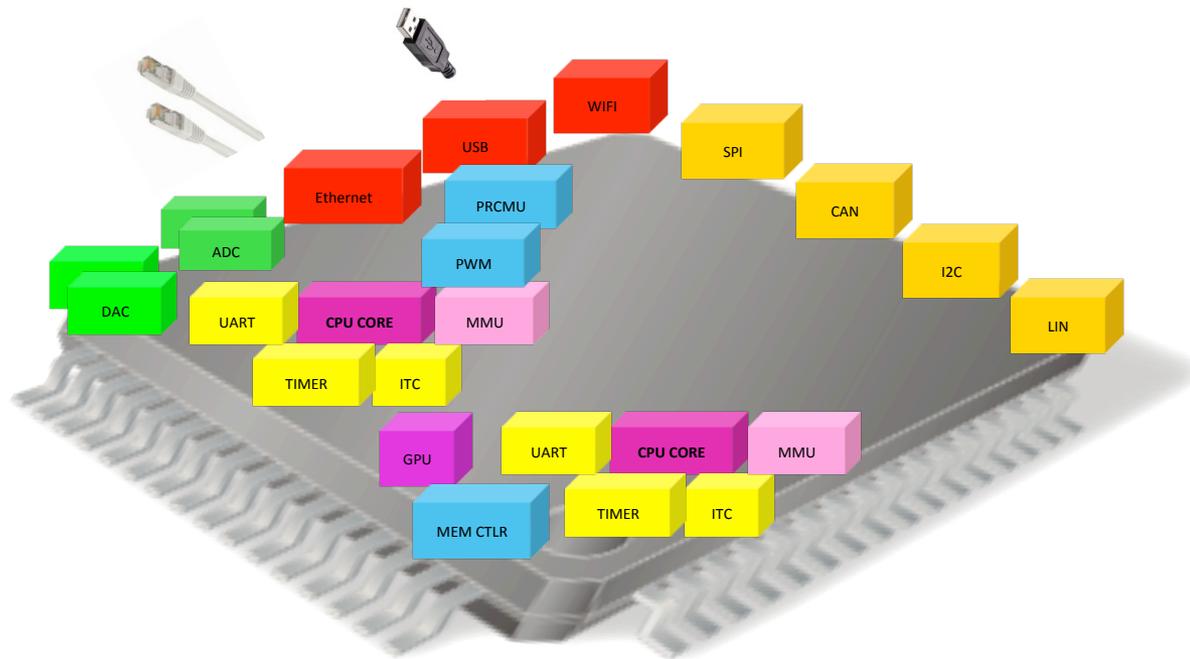


Multi-processeurs

Nombreuses interfaces E/S

Les produits à développer : des systèmes sur puce complexes

5



Multi-processeurs

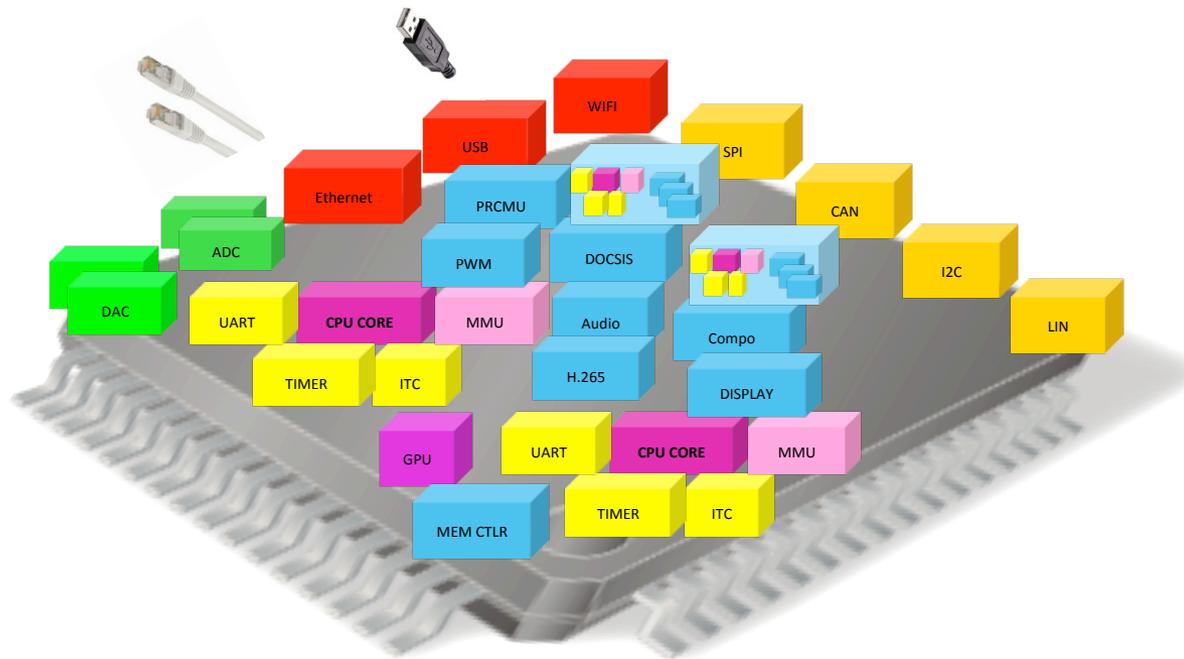
Nombreuses interfaces E/S

Nombreux arbres d'horloges

Plusieurs domaines d'alimentation

Les produits à développer : des systèmes sur puce complexes

5



Multi-processeurs

Accélérateurs matériels

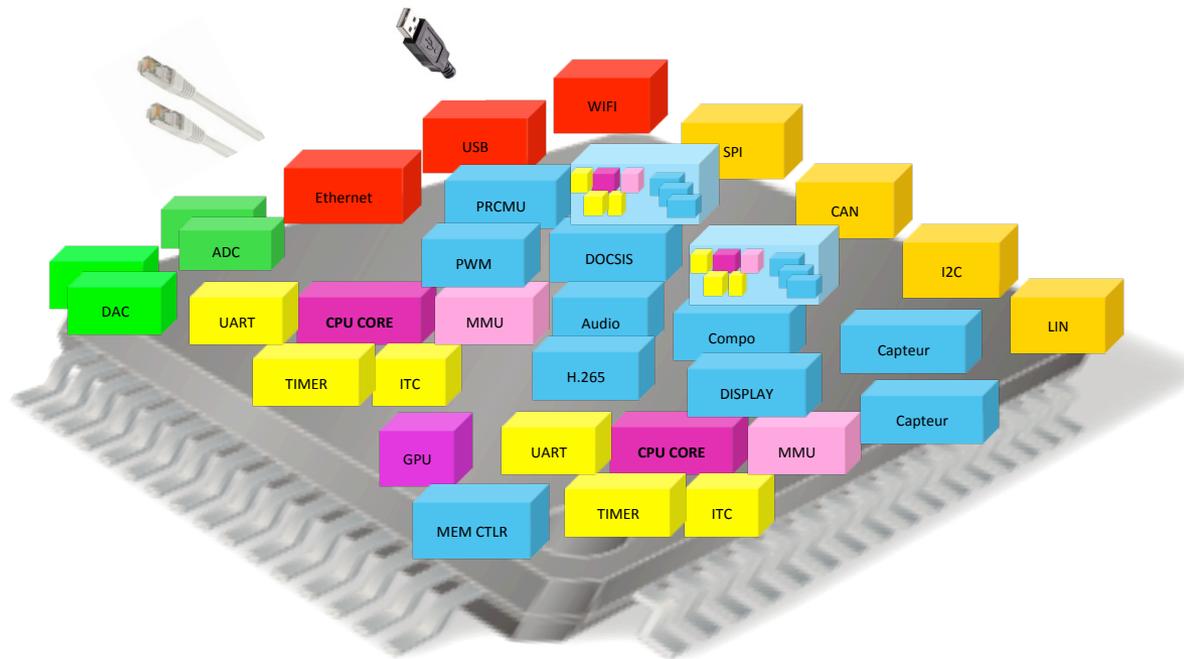
Nombreuses interfaces E/S

Nombreux arbres d'horloges

Plusieurs domaines d'alimentation

Les produits à développer : des systèmes sur puce complexes

5



Multi-processeurs

Accélérateurs matériels

Nombreuses interfaces E/S

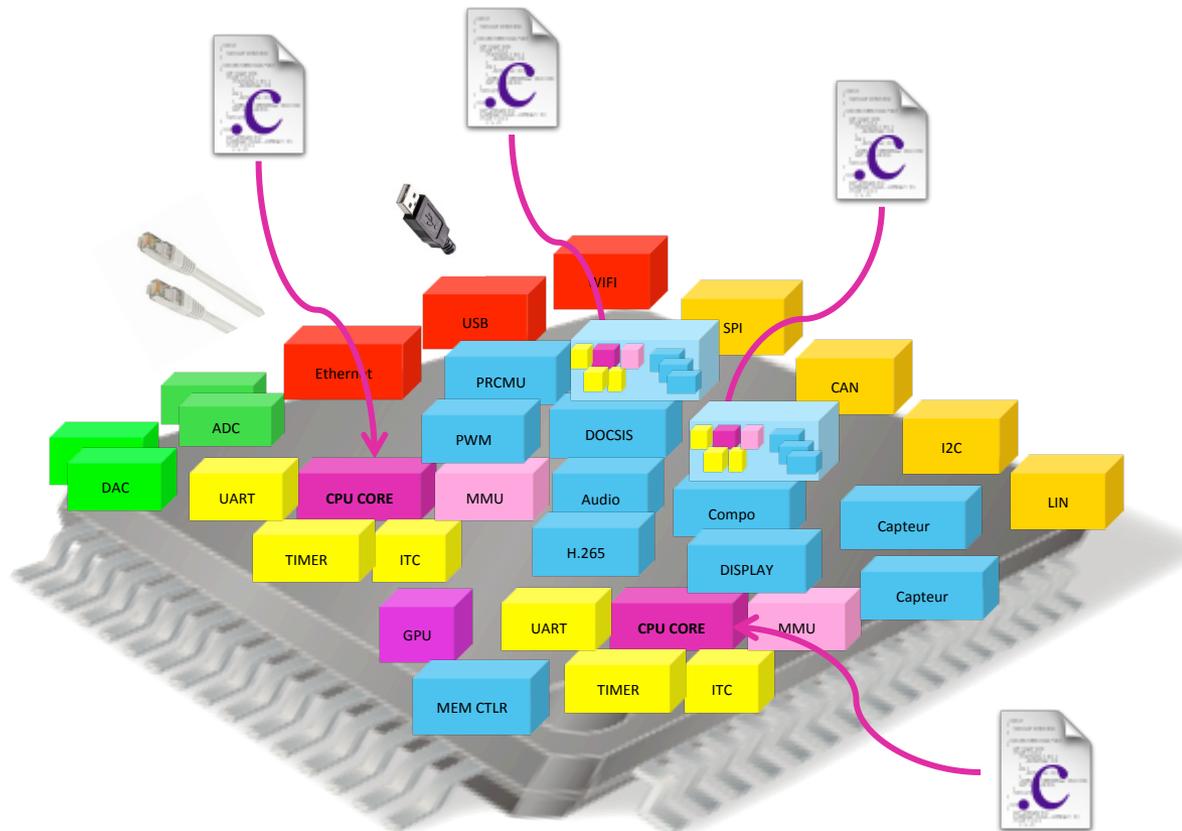
Capteurs

Nombreux arbres d'horloges

Plusieurs domaines d'alimentation

Les produits à développer : des systèmes sur puce complexes

5



Multi-processeurs

Nombreuses interfaces E/S

Nombreux arbres d'horloges

Plusieurs domaines d'alimentation

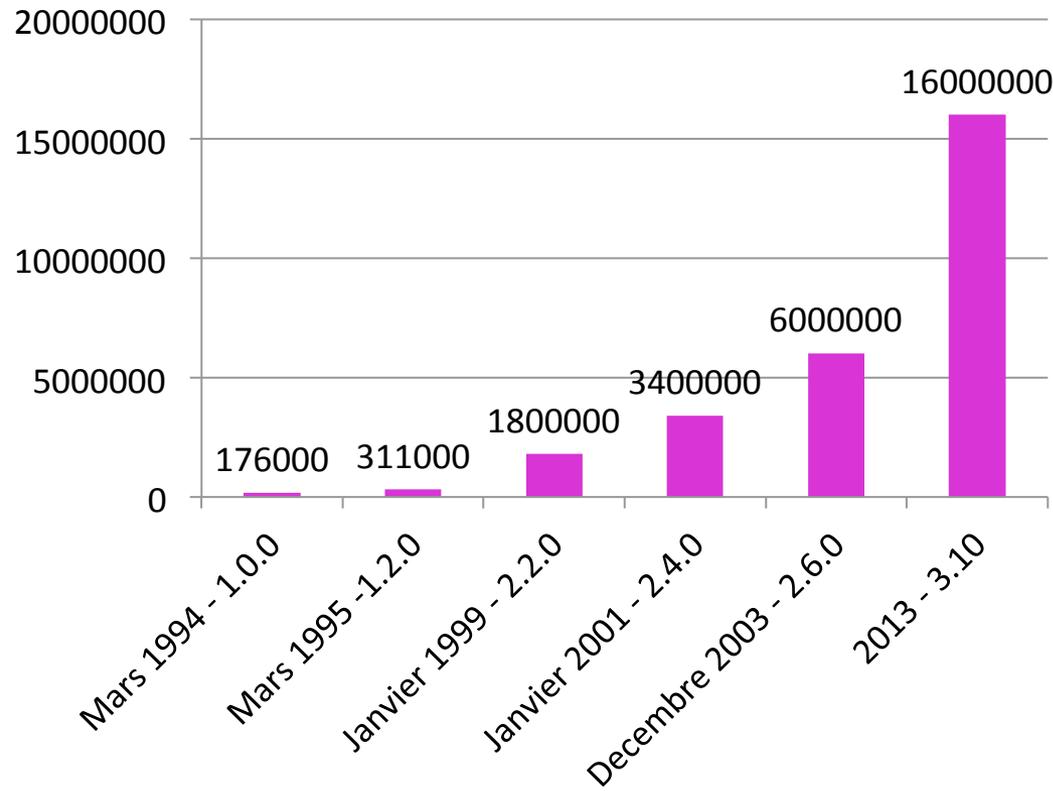
Accélérateurs matériels

Capteurs

Logiciel embarqué

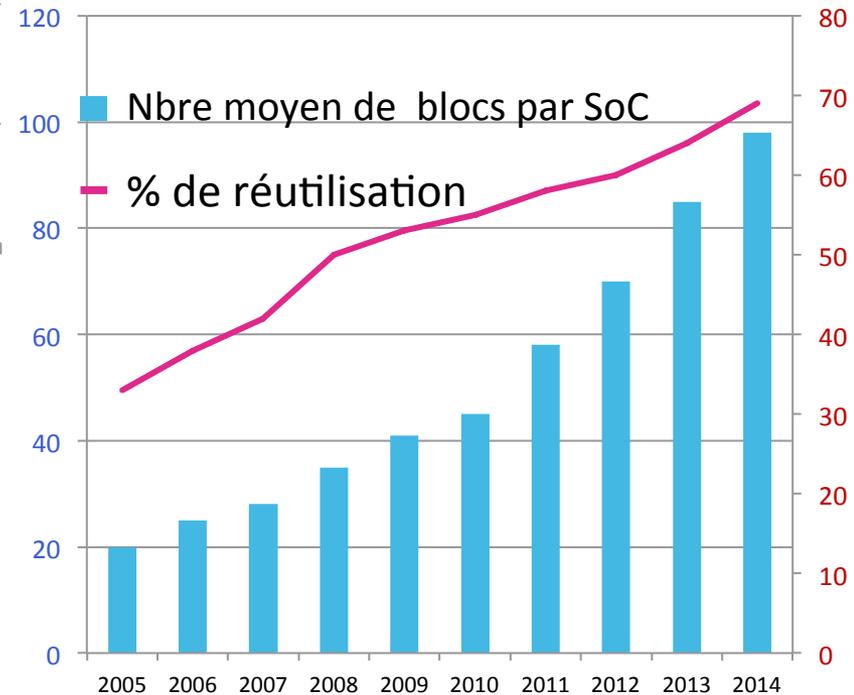
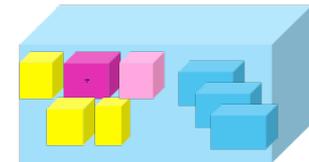
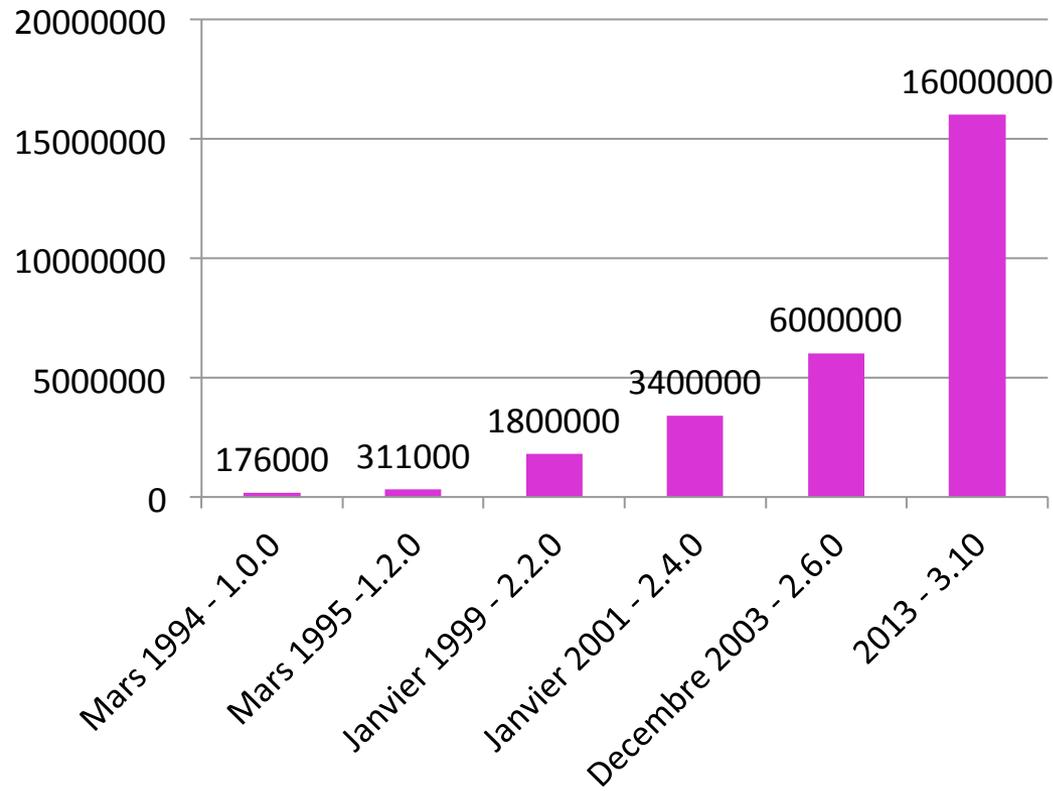
Une complexité exponentielle

Nombre de lignes de code
Noyau Linux



Une complexité exponentielle

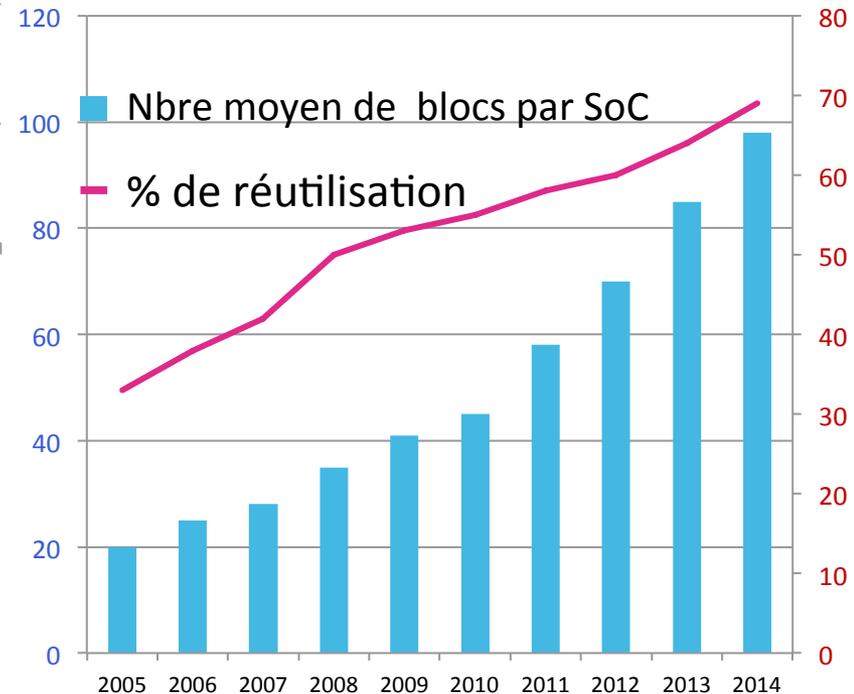
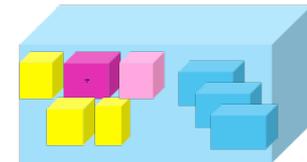
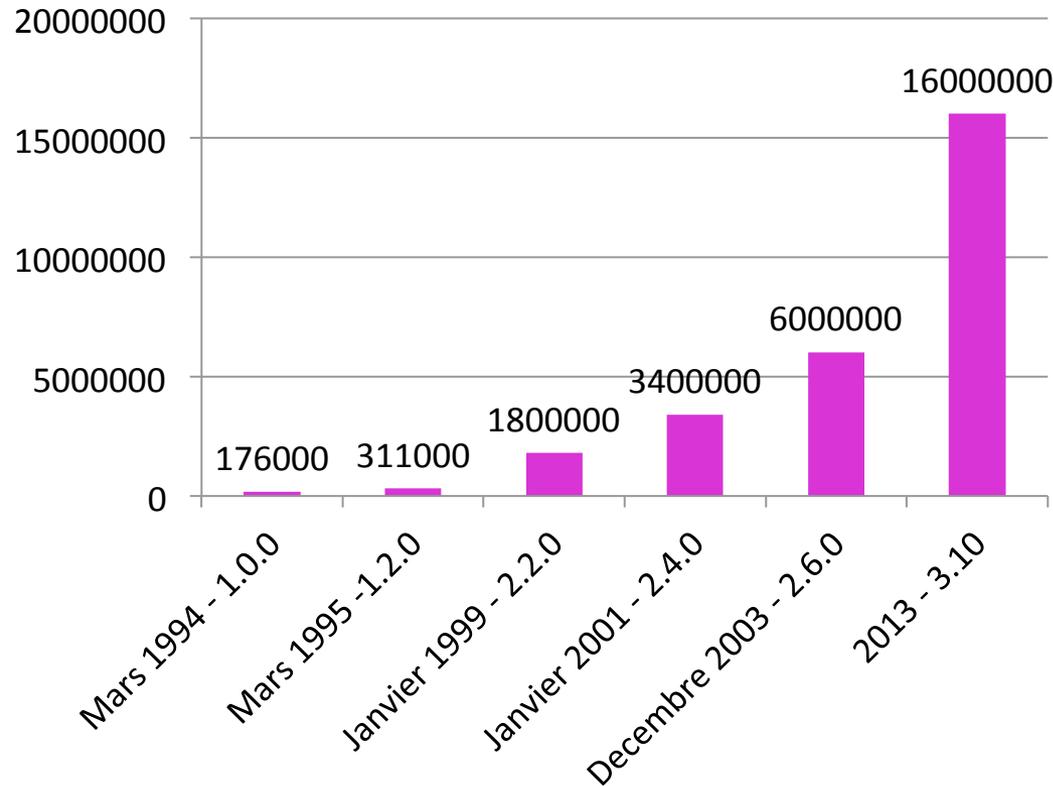
Nombre de lignes de code Noyau Linux



Source: Semico - octobre 2010

Une complexité exponentielle

Nombre de lignes de code
Noyau Linux

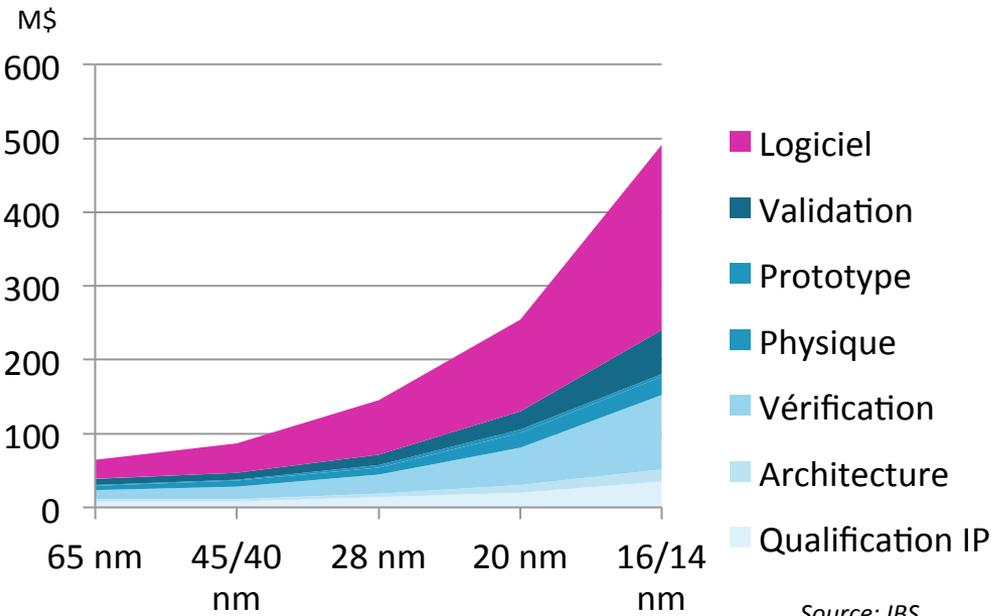


Source: Semico - octobre 2010

- ✔ De plus en plus de fonctions
- ✔ Intégration logiciel/matériel
- ✔ Fiabilité à maximiser

Une concurrence exacerbée

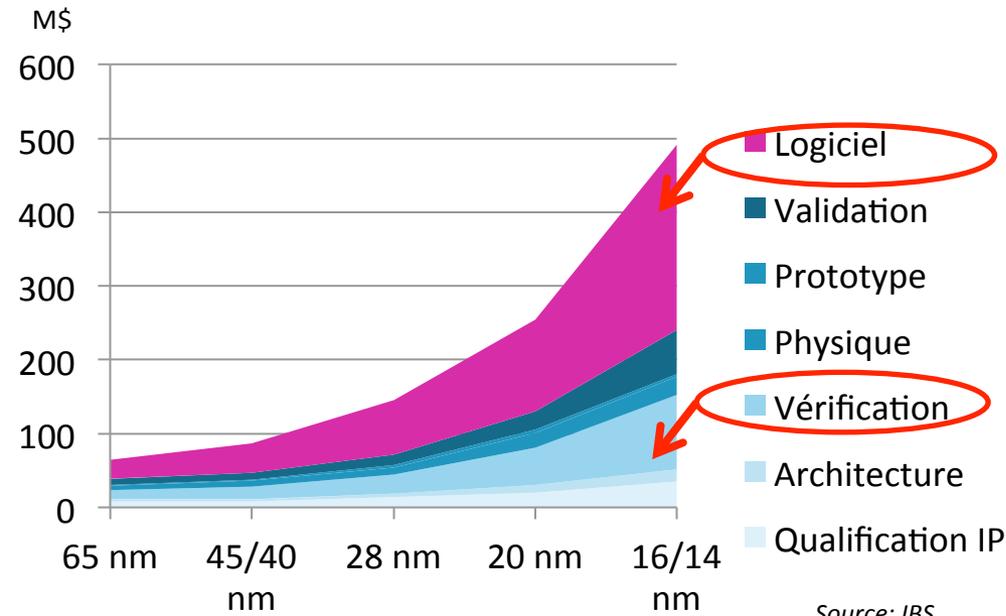
Augmentation des coûts de conception



Source: IBS

Une concurrence exacerbée

Augmentation des coûts de conception

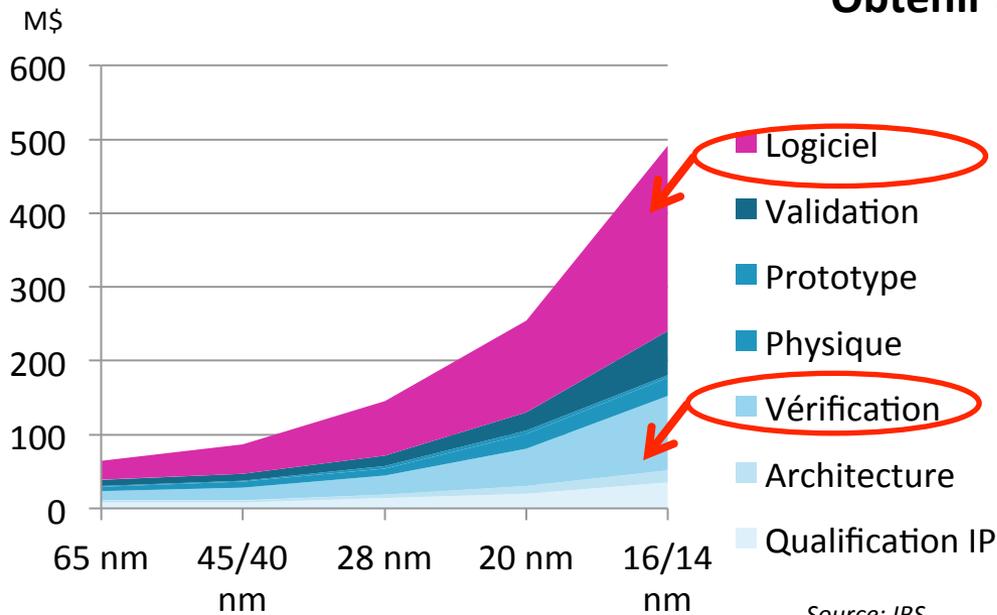


Source: IBS

Une concurrence exacerbée

Augmentation des coûts de conception

Obtenir des revenus pour supporter ces coûts

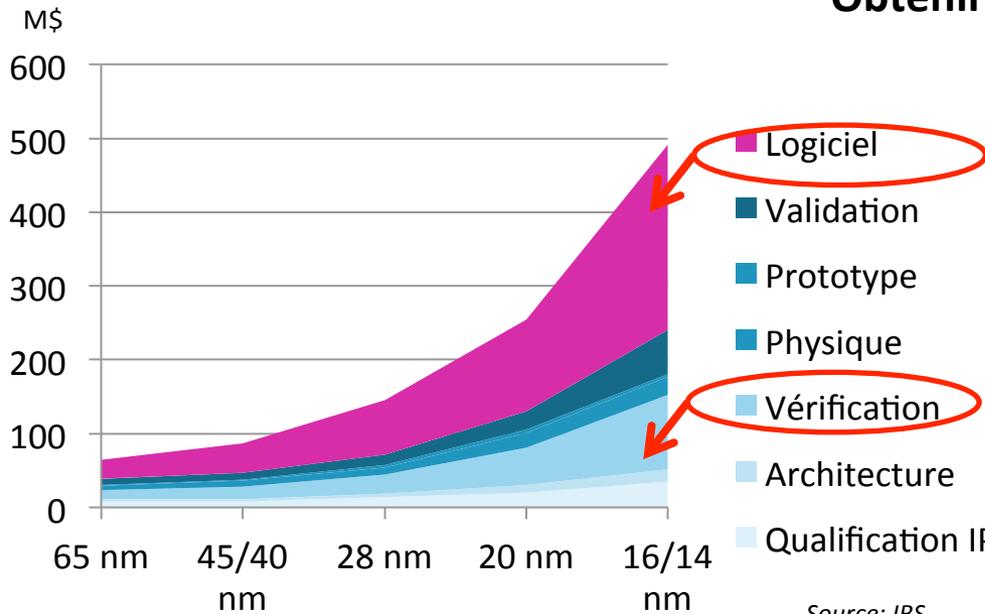


Source: IBS

Une concurrence exacerbée

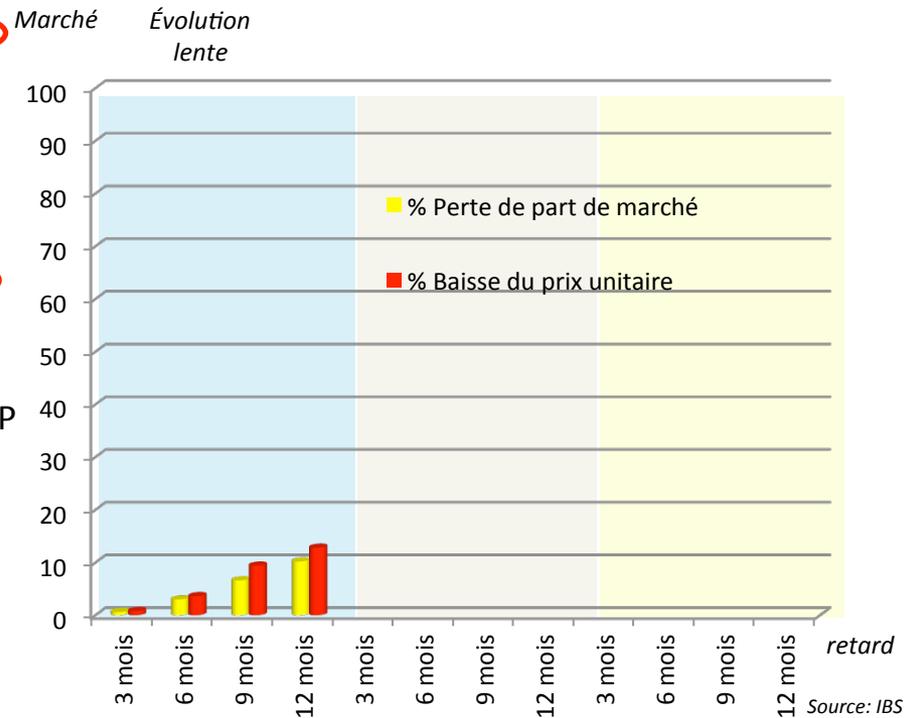
Augmentation des coûts de conception

Obtenir des revenus pour supporter ces coûts



Source: IBS

Impact du retard sur les revenus

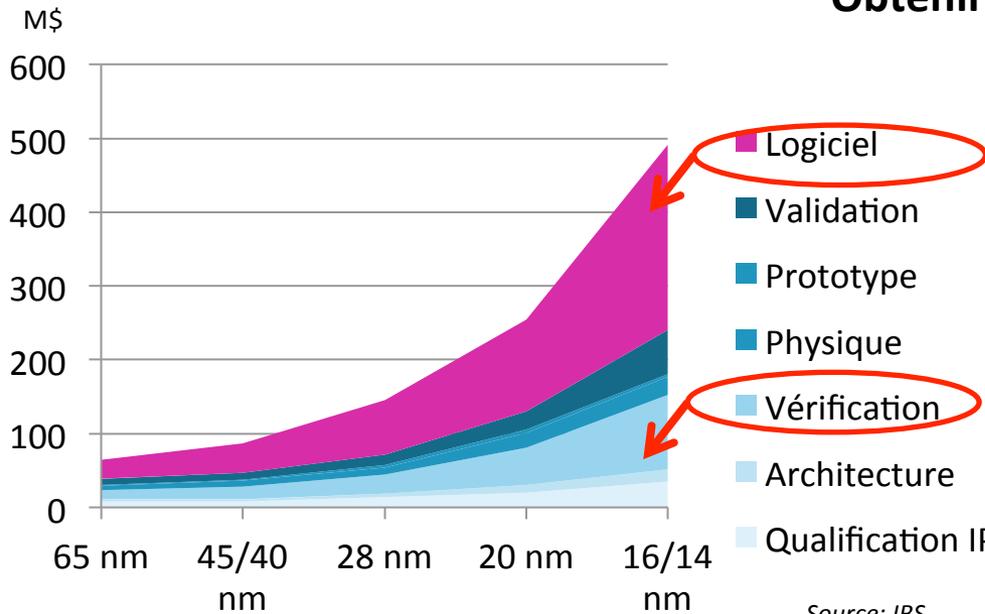


Source: IBS

Une concurrence exacerbée

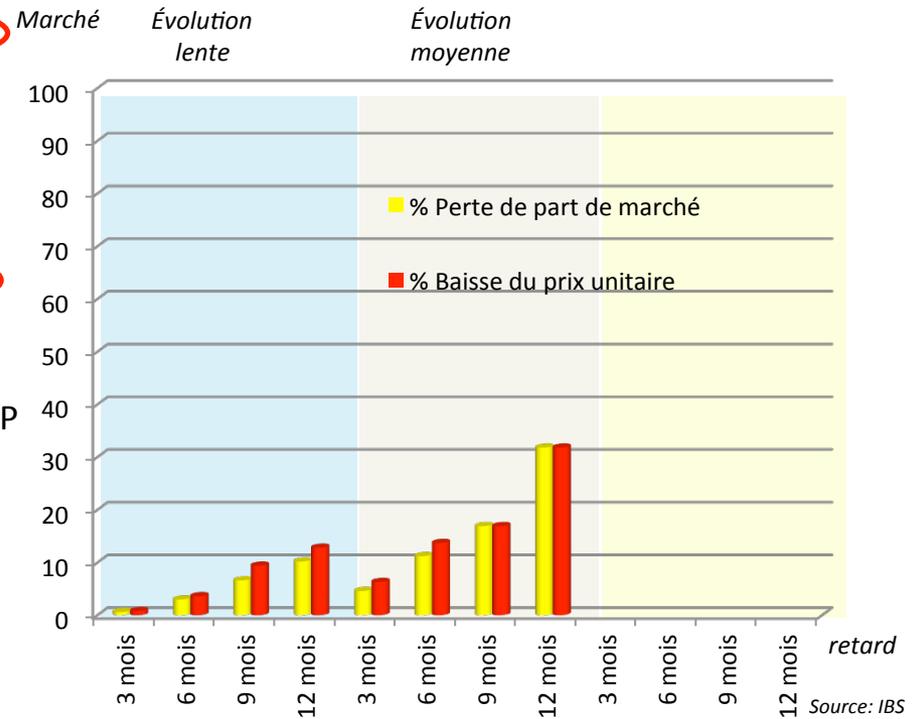
Augmentation des coûts de conception

Obtenir des revenus pour supporter ces coûts



Source: IBS

Impact du retard sur les revenus

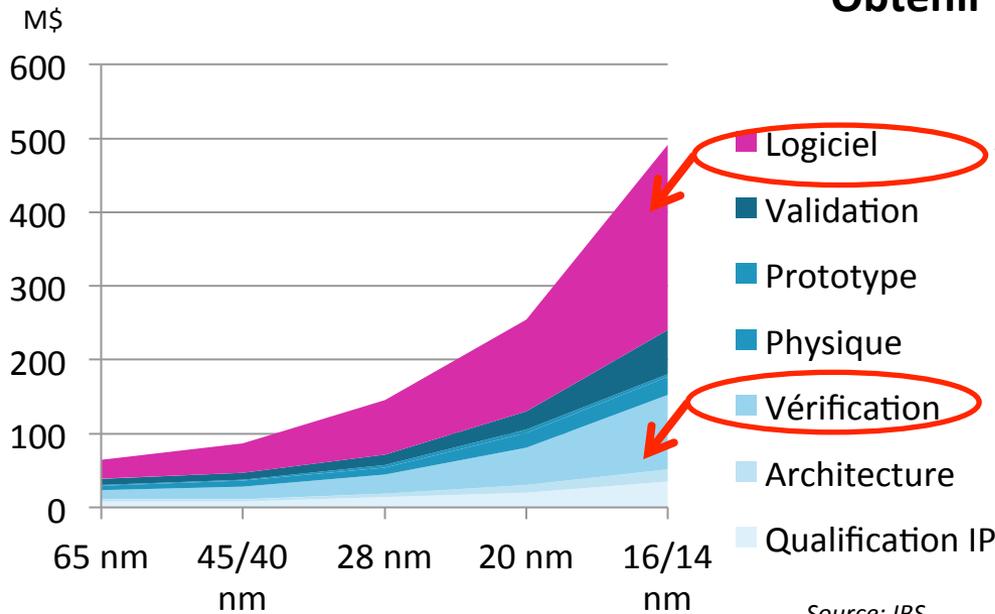


Source: IBS

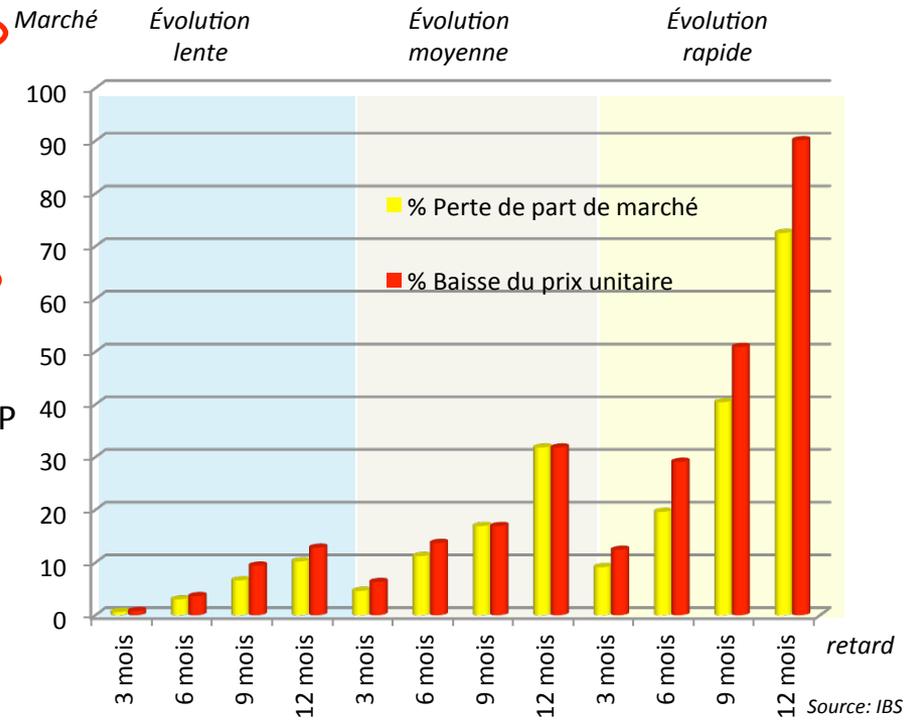
Une concurrence exacerbée

Augmentation des coûts de conception

Obtenir des revenus pour supporter ces coûts



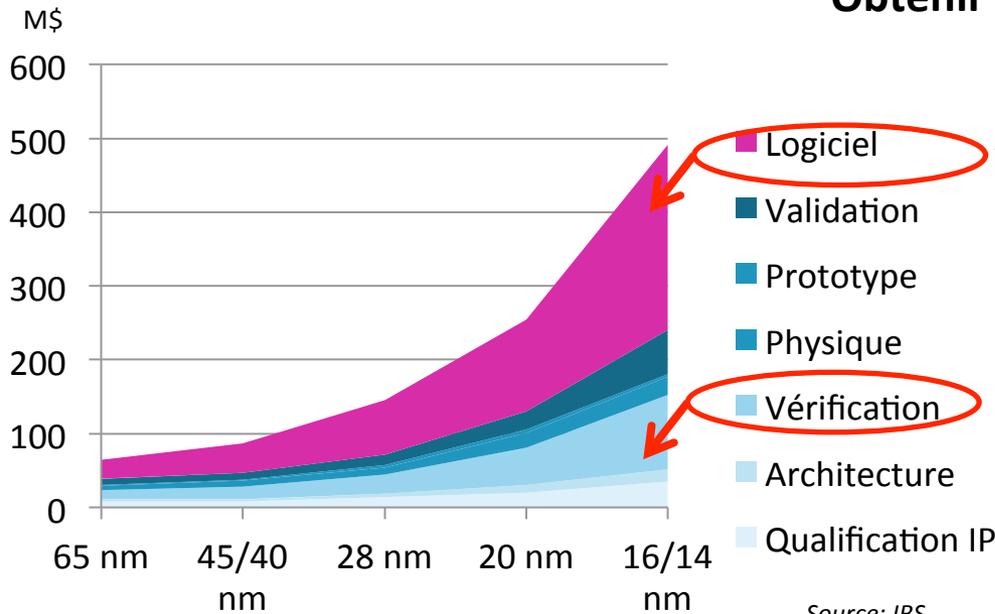
Impact du retard sur les revenus



Une concurrence exacerbée

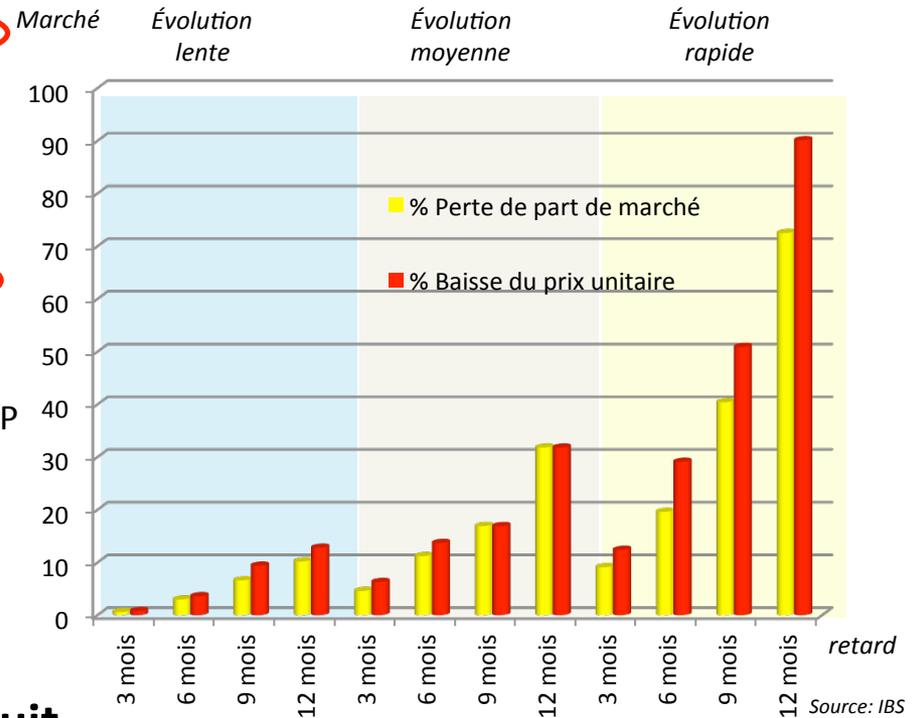
Augmentation des coûts de conception

Obtenir des revenus pour supporter ces coûts



Source: IBS

Impact du retard sur les revenus



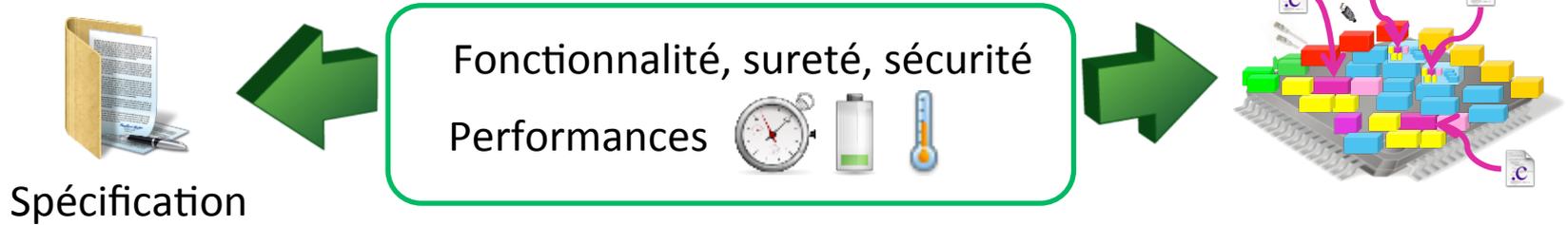
Source: IBS

- ▶ Proposer le bon produit
- ▶ Fournir toutes les fonctionnalités attendues
- ▶ Arriver au plus tôt sur le marché

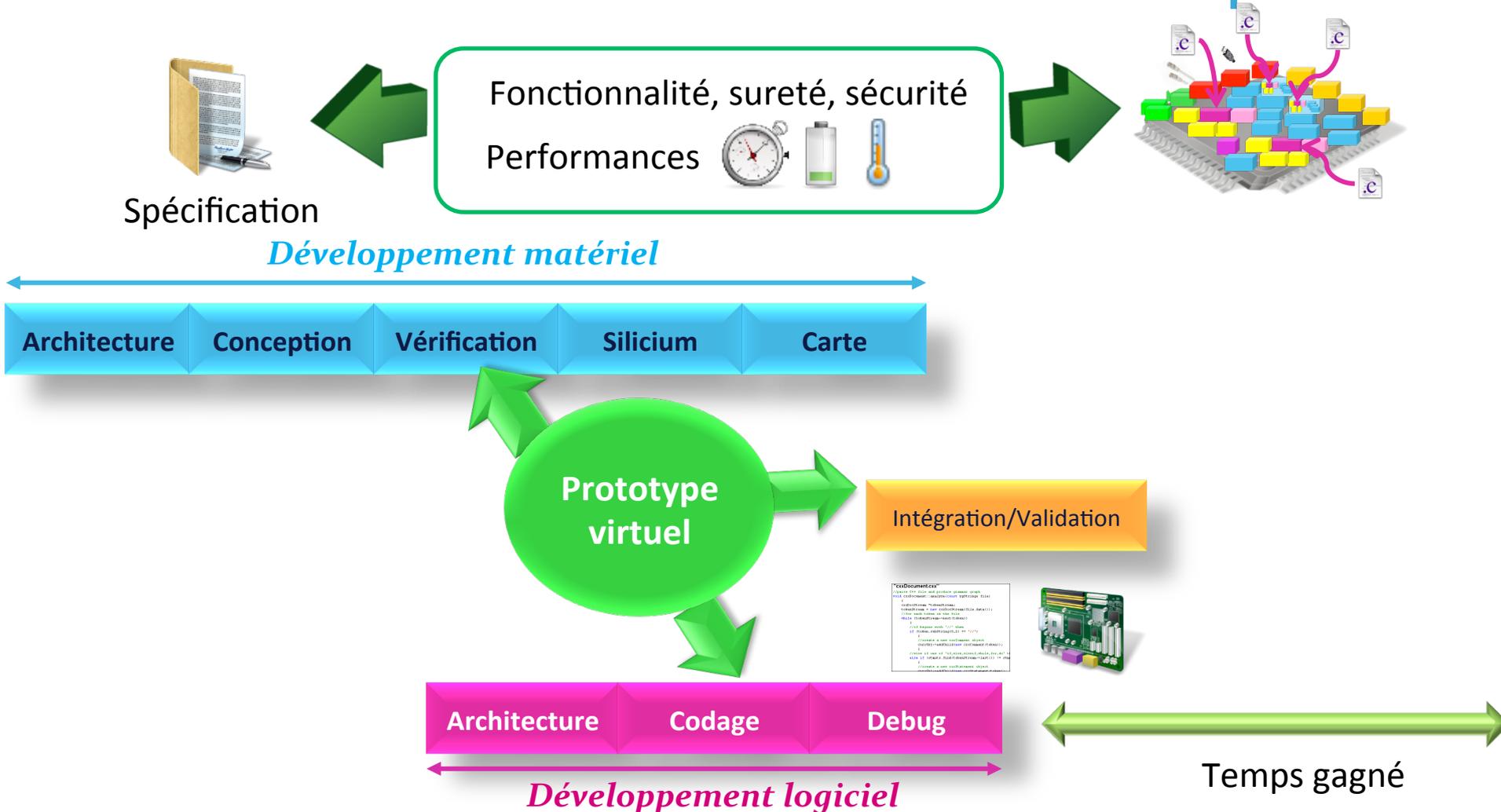


Prototypage virtuel

Evolution des flots de conception



Evolution des flots de conception

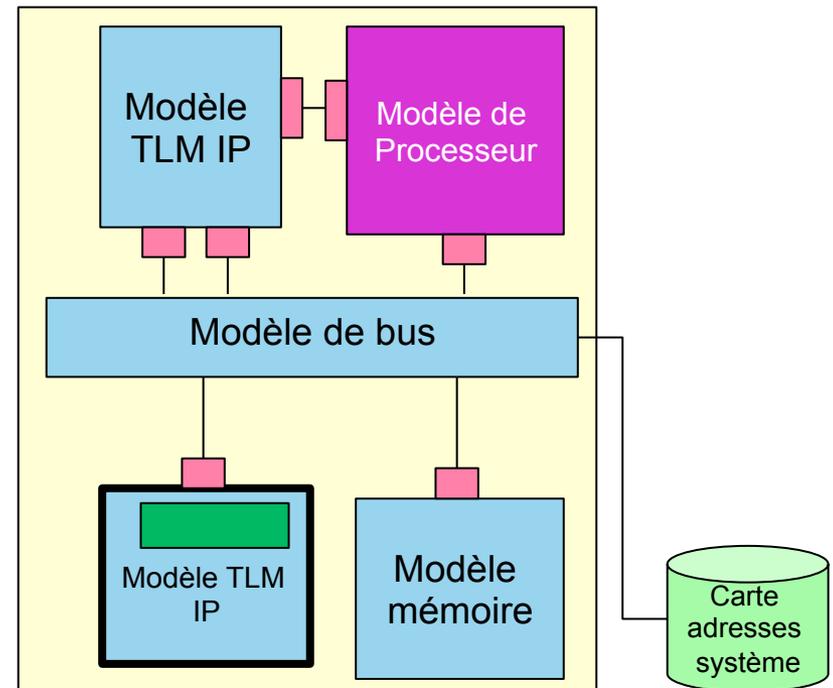


- ✔ Validation en avance de phase avec des modèles
- ✔ Interactions logiciel/matériel
- ✔ Référence commune aux différents métiers

Vérification fonctionnelle de bloc matériel

Anticiper la mise au point de l'environnement de test

- ✔ Modèle de **référence**
 - ✔ Fonctionnalité
 - ✔ Registres
 - ✔ Interfaces de communication
- ✔ Désambiguïsation de la spécification
- ✔ Modèle disponible avant le RTL
- ✔ Mise au point des tests
- ✔ Réutilisabilité des modèles et des tests

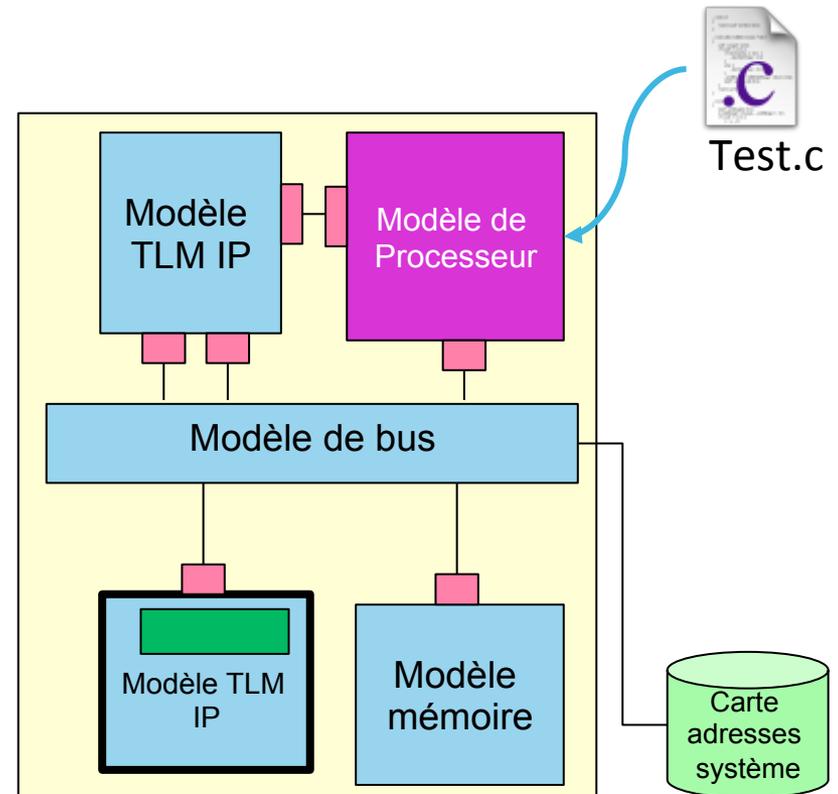


Vérification fonctionnelle de bloc matériel

10

🔗 Anticiper la mise au point de l'environnement de test

- ✔ Modèle de **référence**
 - ✔ Fonctionnalité
 - ✔ Registres
 - ✔ Interfaces de communication
- ✔ Désambiguïsation de la spécification
- ✔ Modèle disponible avant le RTL
- ✔ Mise au point des tests
- ✔ Réutilisabilité des modèles et des tests

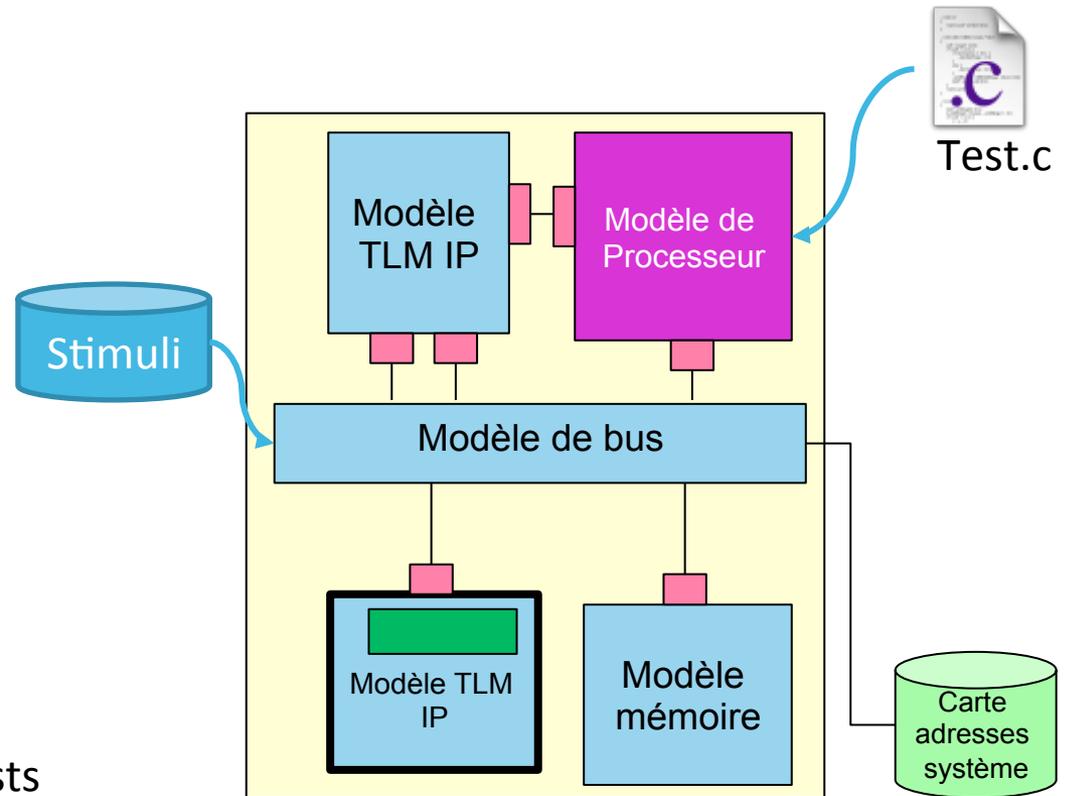


Vérification fonctionnelle de bloc matériel

10

🔗 Anticiper la mise au point de l'environnement de test

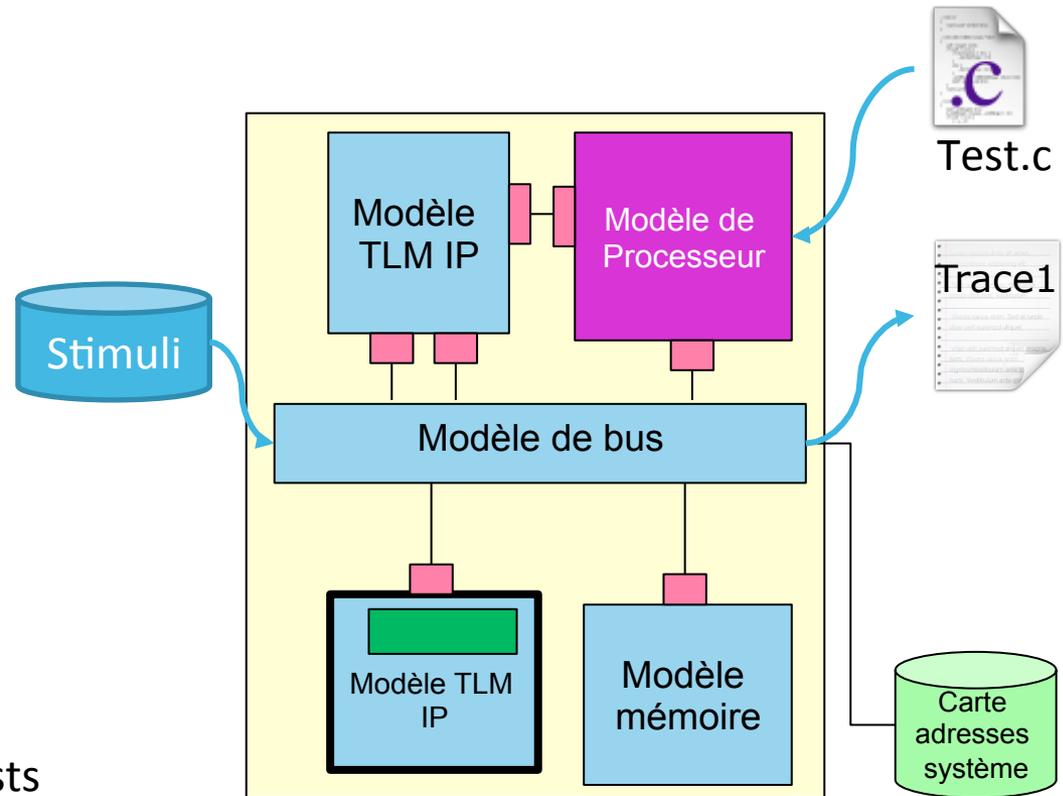
- ✔ Modèle de **référence**
 - ✔ Fonctionnalité
 - ✔ Registres
 - ✔ Interfaces de communication
- ✔ Désambiguïsation de la spécification
- ✔ Modèle disponible avant le RTL
- ✔ Mise au point des tests
- ✔ Réutilisabilité des modèles et des tests



Vérification fonctionnelle de bloc matériel

Anticiper la mise au point de l'environnement de test

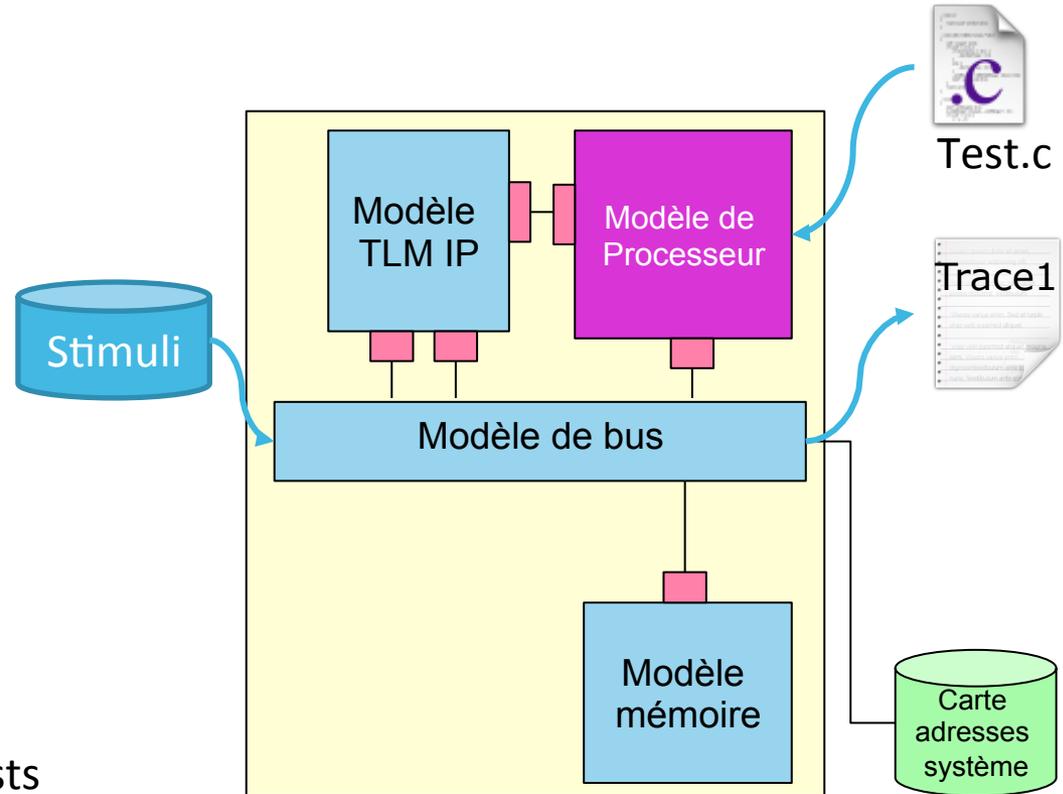
- ✔ Modèle de **référence**
 - ✔ Fonctionnalité
 - ✔ Registres
 - ✔ Interfaces de communication
- ✔ Désambiguïisation de la spécification
- ✔ Modèle disponible avant le RTL
- ✔ Mise au point des tests
- ✔ Réutilisabilité des modèles et des tests



Vérification fonctionnelle de bloc matériel

Anticiper la mise au point de l'environnement de test

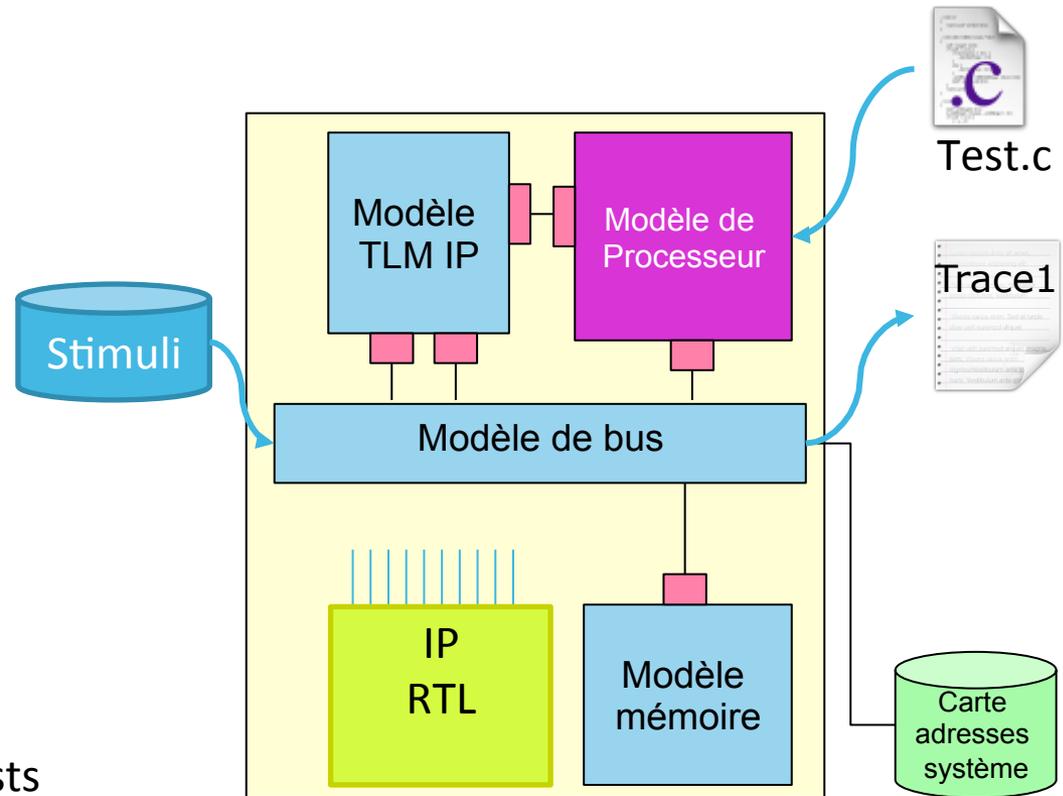
- ✔ Modèle de **référence**
 - ✔ Fonctionnalité
 - ✔ Registres
 - ✔ Interfaces de communication
- ✔ Désambiguïisation de la spécification
- ✔ Modèle disponible avant le RTL
- ✔ Mise au point des tests
- ✔ Réutilisabilité des modèles et des tests



Vérification fonctionnelle de bloc matériel

Anticiper la mise au point de l'environnement de test

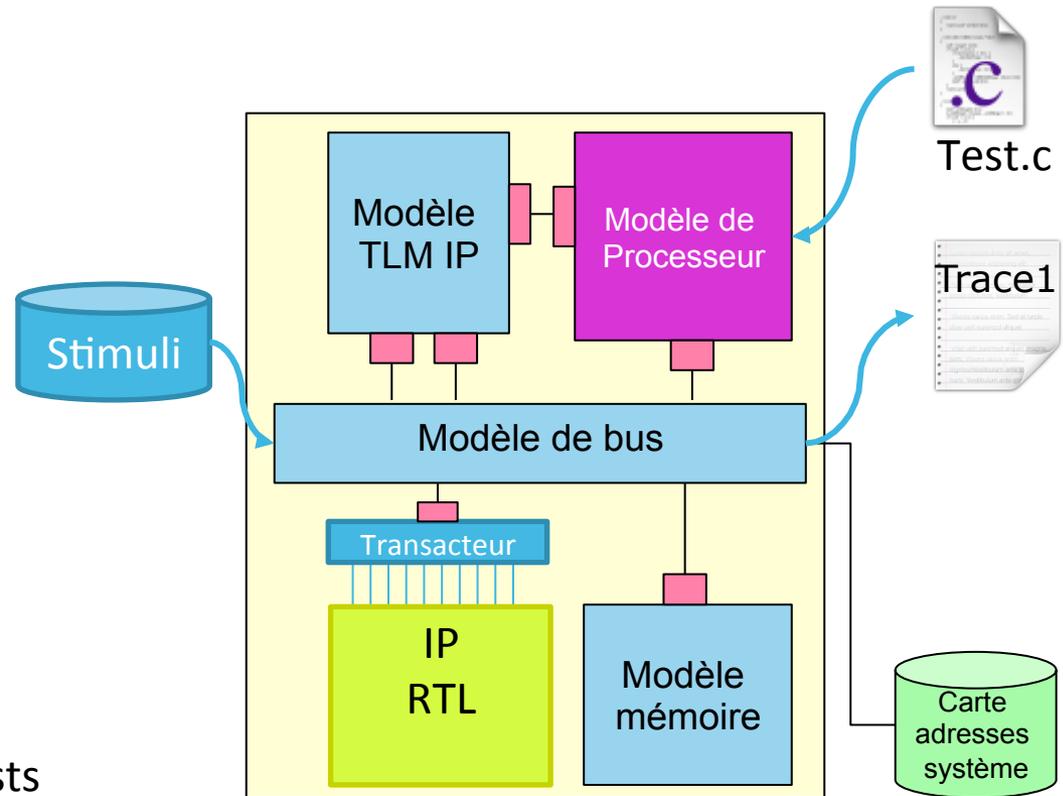
- ✔ Modèle de **référence**
 - ✔ Fonctionnalité
 - ✔ Registres
 - ✔ Interfaces de communication
- ✔ Désambiguïisation de la spécification
- ✔ Modèle disponible avant le RTL
- ✔ Mise au point des tests
- ✔ Réutilisabilité des modèles et des tests



Vérification fonctionnelle de bloc matériel

Anticiper la mise au point de l'environnement de test

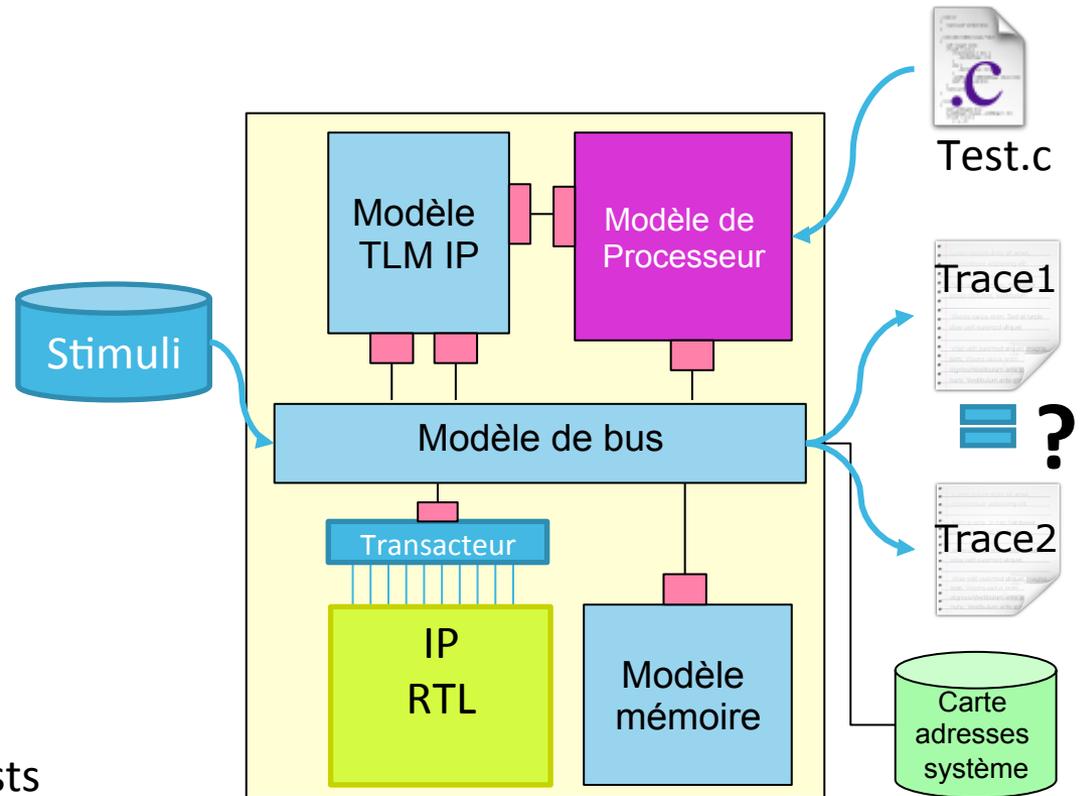
- ✔ Modèle de **référence**
 - ✔ Fonctionnalité
 - ✔ Registres
 - ✔ Interfaces de communication
- ✔ Désambiguïisation de la spécification
- ✔ Modèle disponible avant le RTL
- ✔ Mise au point des tests
- ✔ Réutilisabilité des modèles et des tests



Vérification fonctionnelle de bloc matériel

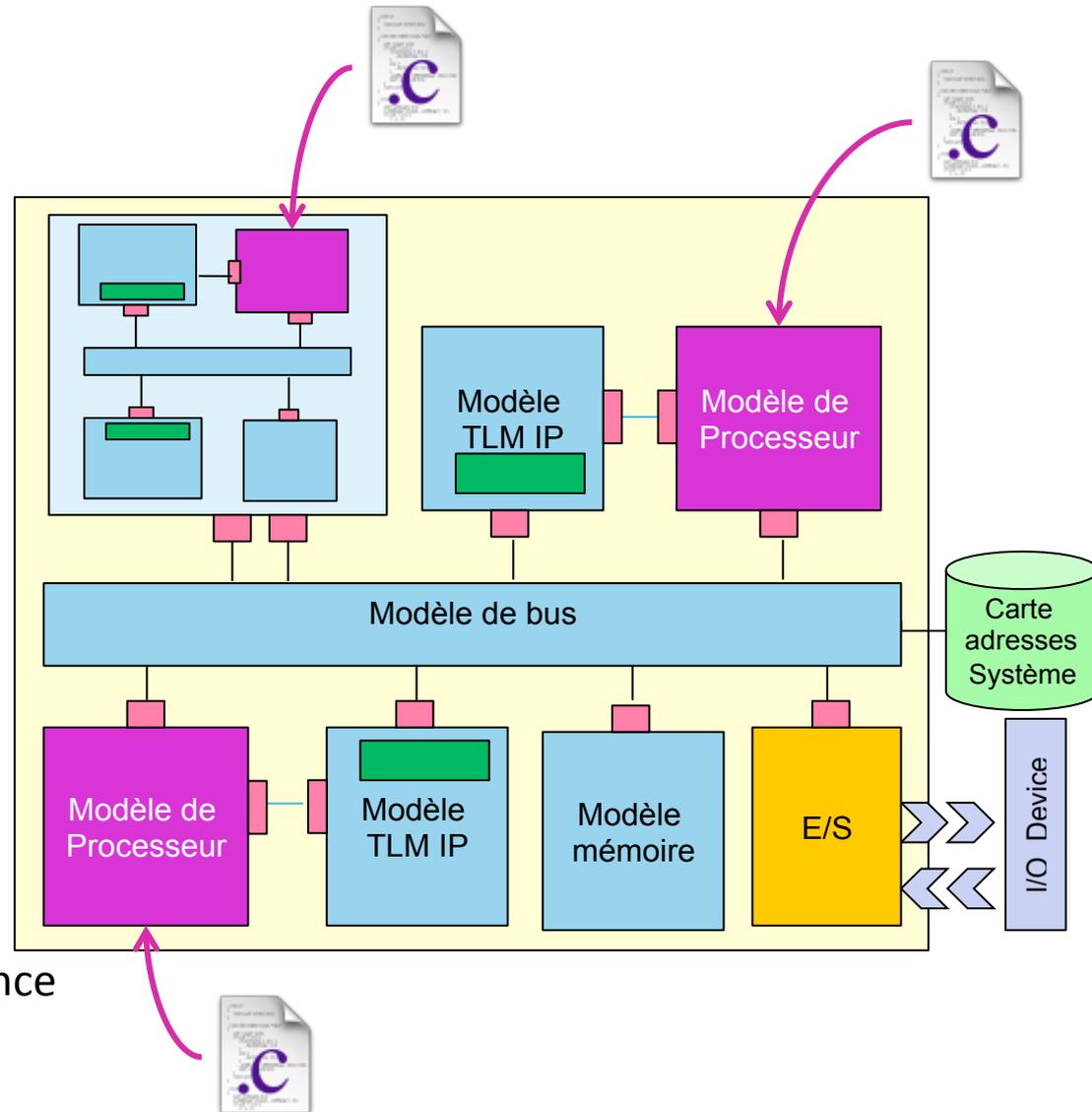
Anticiper la mise au point de l'environnement de test

- ✔ Modèle de **référence**
 - ✔ Fonctionnalité
 - ✔ Registres
 - ✔ Interfaces de communication
- ✔ Désambiguïisation de la spécification
- ✔ Modèle disponible avant le RTL
- ✔ Mise au point des tests
- ✔ Réutilisabilité des modèles et des tests



Développement et intégration logiciel

- ✓ Modèles
 - ✓ Fonctionnalité
 - ✓ Registres
 - ✓ Interfaces de communication
- ✓ Debug interactif
 - ✓ Vitesse d'exécution
 - ✓ Observabilité
- ✓ Représentativité
 - ✓ Pas de modification du logiciel sur le silicium
- ✓ Réutilisation des modèles de référence



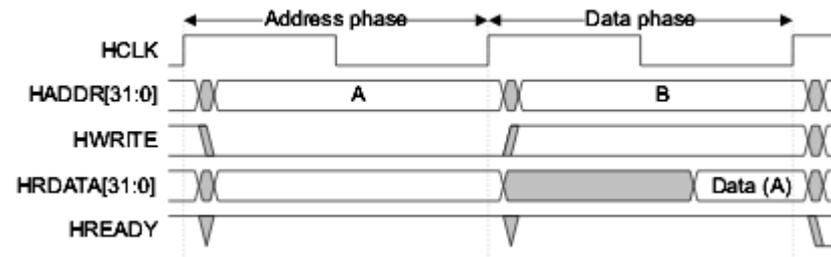
Modèles et niveaux d'abstraction

Algorithmique

C, C++ Cycle accurate
modèle de micro-architecture, cycle à cycle

VHDL, Verilog Transfert de registre (RTL)
logique combinatoire et registres ; horloges

Porte logique



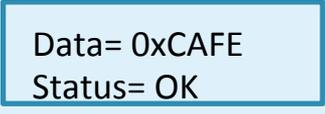
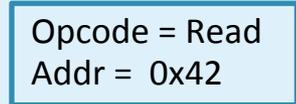
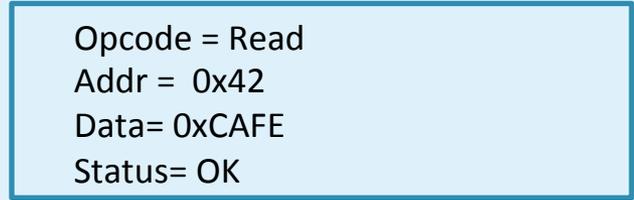
Modèles et niveaux d'abstraction

Algorithmique

SystemC/TLM

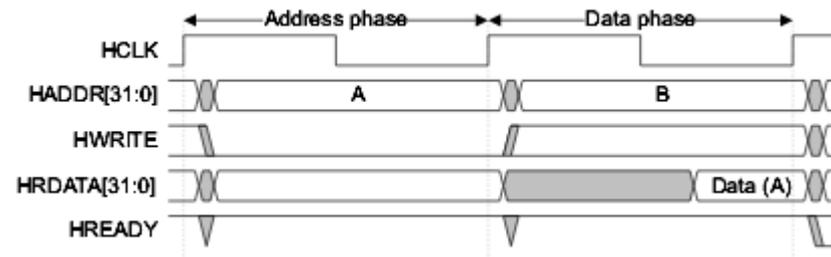
Transactionnel, faible couplage temporel
modèle d'architecture avec un temps imprécis

Transactionnel, fort couplage temporel
modèle d'architecture avec un temps précis



C, C++ **Cycle accurate**
modèle de micro-architecture, cycle à cycle

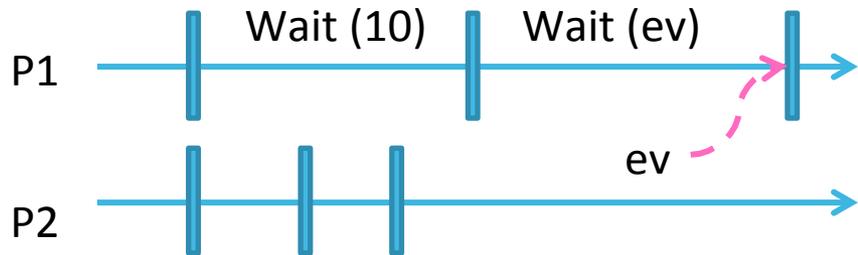
VHDL, Verilog **Transfert de registre (RTL)**
logique combinatoire et registres ; horloges



Porte logique



Synchroniser les modèles

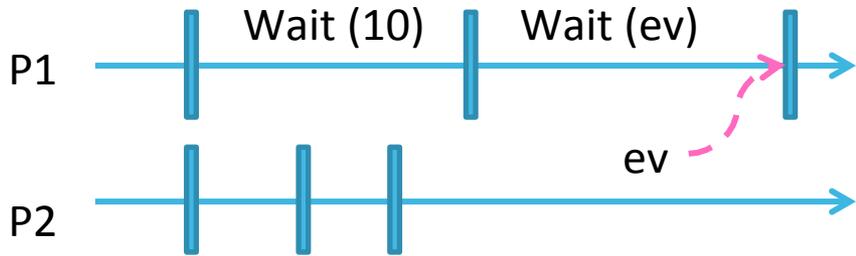


SystemC : Ordonnanceur coopératif

TLM : Pas d'horloges pour se synchroniser

❓ **Quand rendre la main à l'ordonnanceur ?**

Synchroniser les modèles



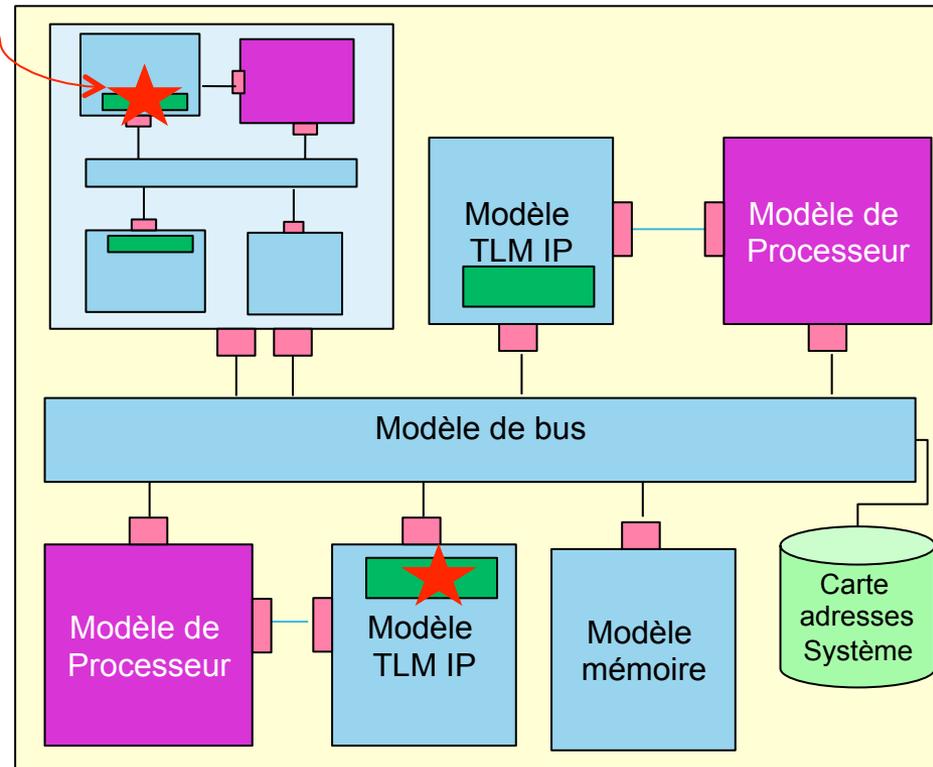
SystemC : Ordonnanceur coopératif

TLM : Pas d'horloges pour se synchroniser

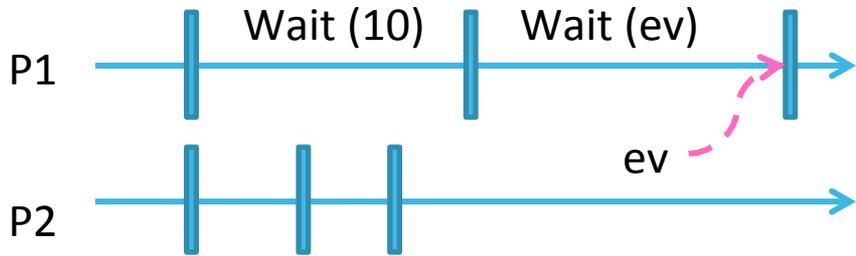
🔍 Quand rendre la main à l'ordonnanceur ?

Synchronize()

- ✔ Synchronisations explicites
 - ✔ Nécessaires pour la fonctionnalité
 - ✔ Actions qui impactent les autres blocs
 - ✔ Registres de contrôle, status



Synchroniser les modèles



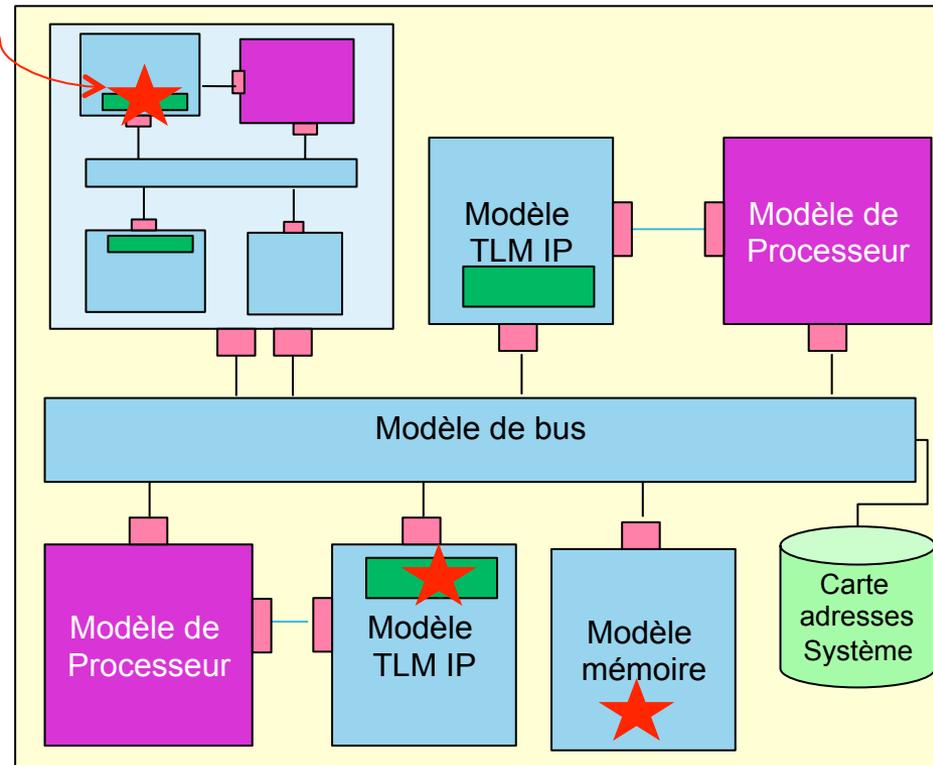
SystemC : Ordonnanceur coopératif

TLM : Pas d'horloges pour se synchroniser

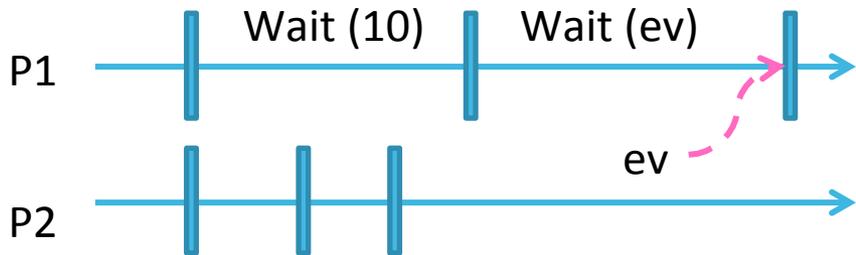
❓ Quand rendre la main à l'ordonnanceur ?

Synchronize()

- ✔ Synchronisations explicites
 - ✔ Nécessaires pour la fonctionnalité
 - ✔ Actions qui impactent les autres blocs
 - ✔ Registres de contrôle, status
 - ✔ Adresses pour synchro inter cpu



Synchroniser les modèles



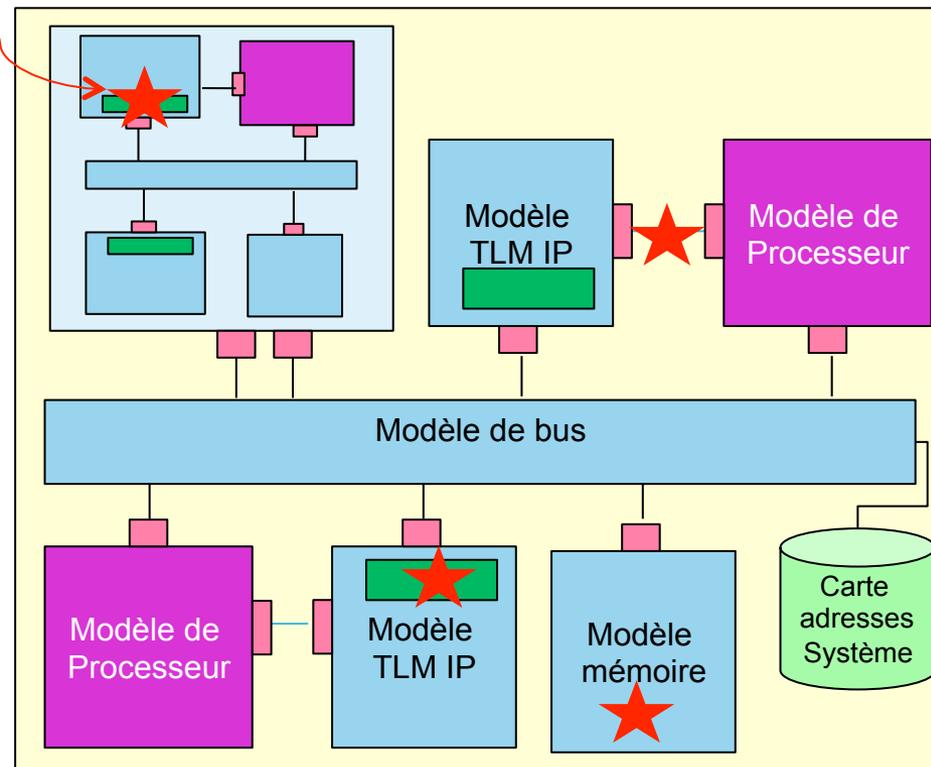
SystemC : Ordonnanceur coopératif

TLM : Pas d'horloges pour se synchroniser

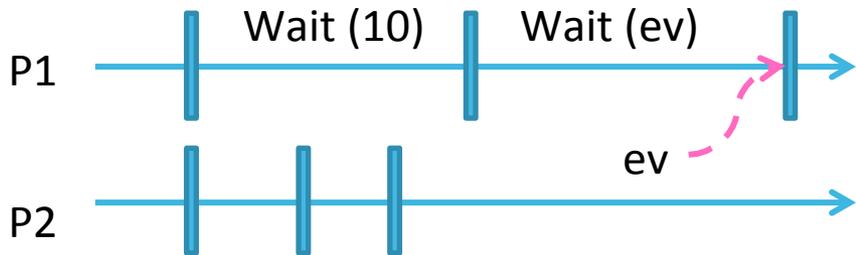
❓ **Quand rendre la main à l'ordonnanceur ?**

Synchronize()

- ✔ Synchronisations explicites
 - ✔ Nécessaires pour la fonctionnalité
 - ✔ Actions qui impactent les autres blocs
 - ✔ Registres de contrôle, status
 - ✔ Adresses pour synchro inter cpu
 - ✔ Génération d'interruptions



Synchroniser les modèles



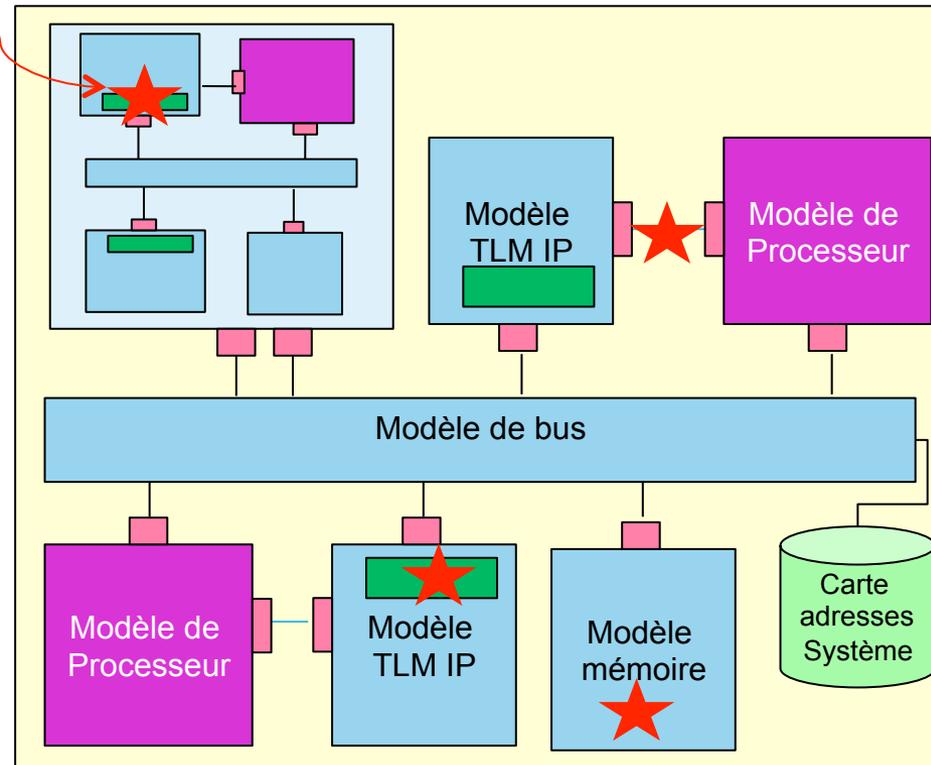
SystemC : Ordonnanceur coopératif

TLM : Pas d'horloges pour se synchroniser

❓ **Quand rendre la main à l'ordonnanceur ?**

Synchronize()

- ✔ Synchronisations explicites
 - ✔ Nécessaires pour la fonctionnalité
 - ✔ Actions qui impactent les autres blocs
 - ✔ Registres de contrôle, status
 - ✔ Adresses pour synchro inter cpu
 - ✔ Génération d'interruptions
- ✔ Réduire le nombre de synchronisations au strict minimum

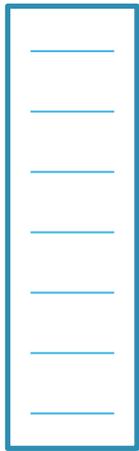


Le temps dans les prototype virtuels

- ❓ *Exécution en temps nul ? Peu réaliste pour modéliser les timers*
- ▶ *Faible couplage temporel : Cumuler des annotations de temps*

Le temps dans les prototype virtuels

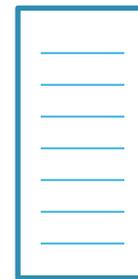
- ❓ *Exécution en temps nul ? Peu réaliste pour modéliser les timers*
- ▶ *Faible couplage temporel : Cumuler des annotations de temps*



Initiateur



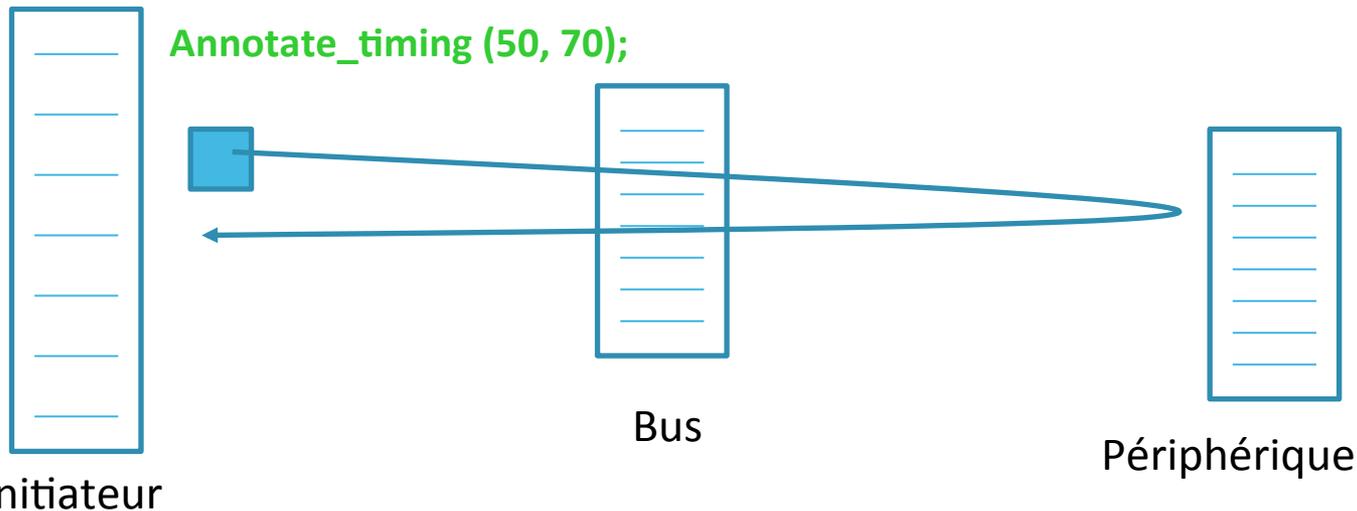
Bus



Périphérique

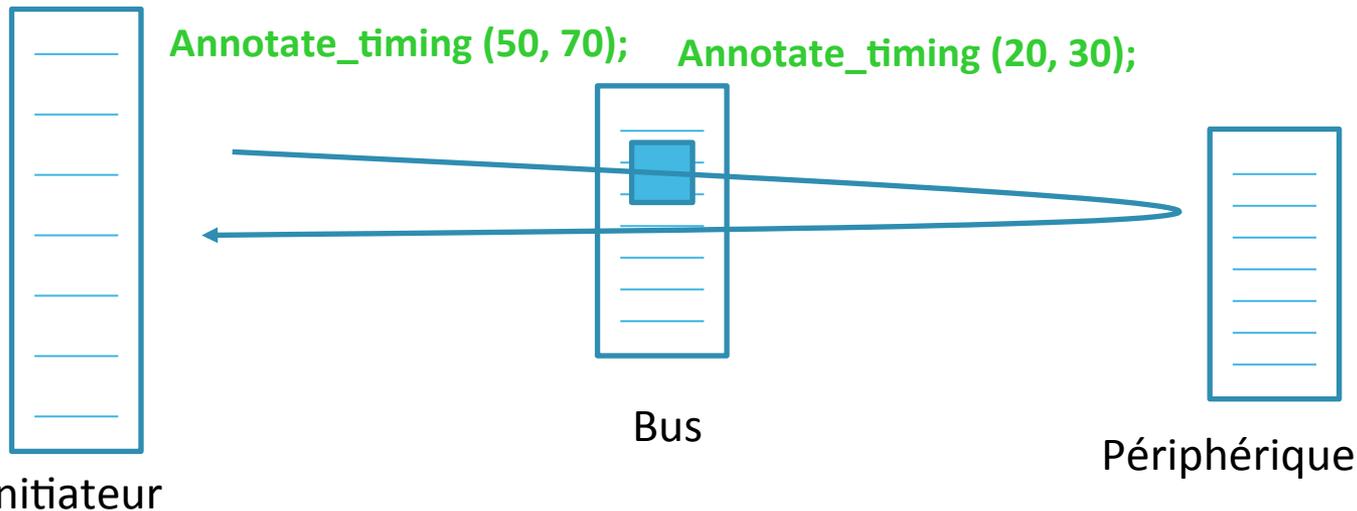
Le temps dans les prototype virtuels

- ❓ *Exécution en temps nul ? Peu réaliste pour modéliser les timers*
- ▶ *Faible couplage temporel : Cumuler des annotations de temps*



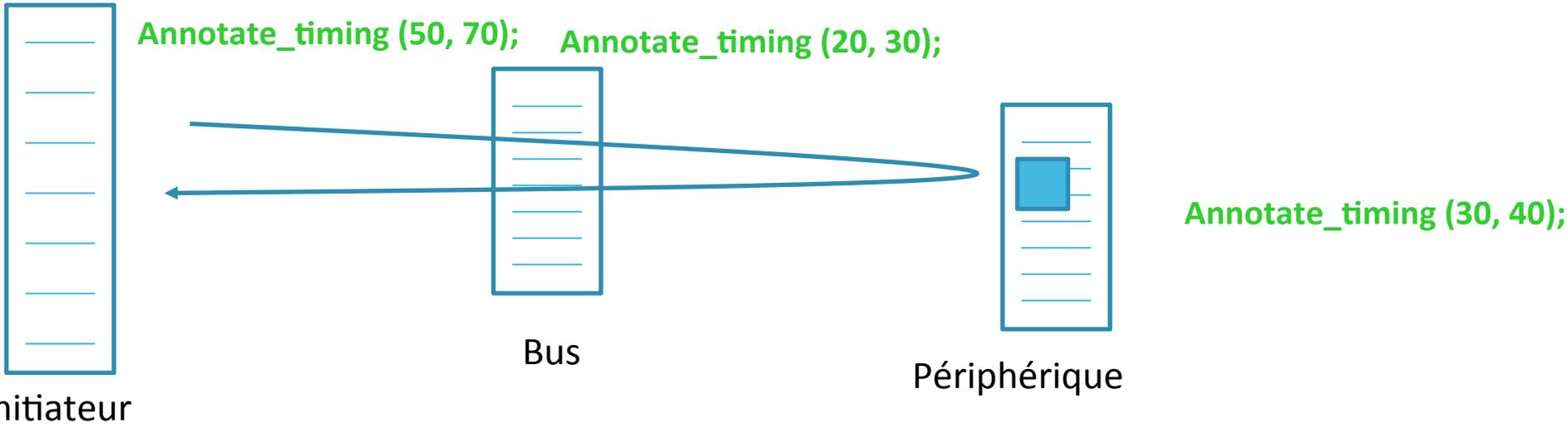
Le temps dans les prototype virtuels

- ❓ *Exécution en temps nul ? Peu réaliste pour modéliser les timers*
- ▶ *Faible couplage temporel : Cumuler des annotations de temps*



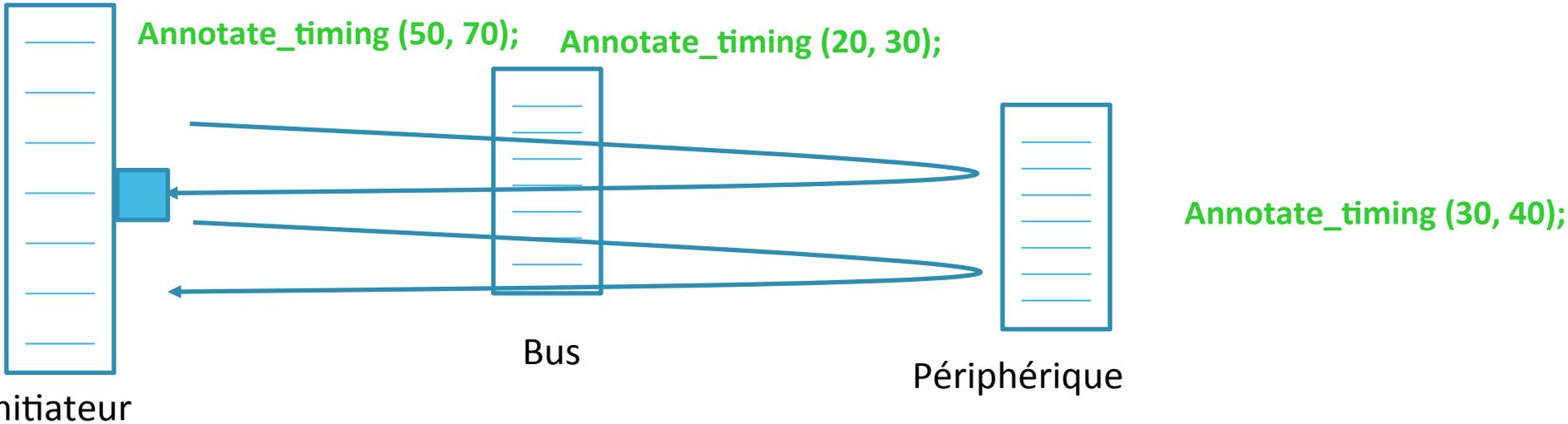
Le temps dans les prototype virtuels

- ❓ *Exécution en temps nul ? Peu réaliste pour modéliser les timers*
- ▶ *Faible couplage temporel : Cumuler des annotations de temps*



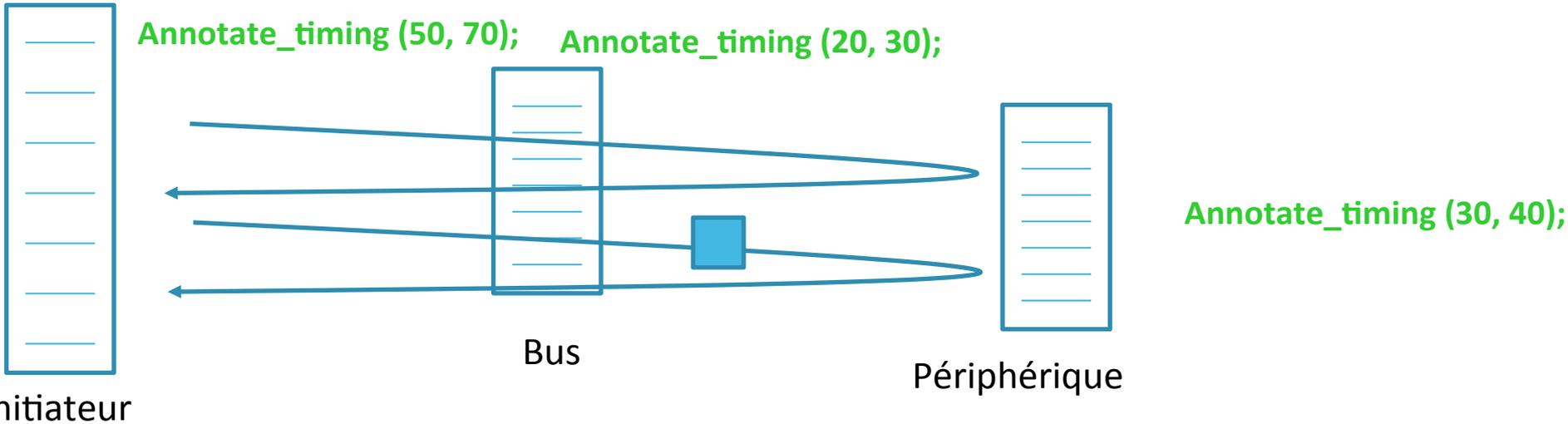
Le temps dans les prototype virtuels

- ❓ *Exécution en temps nul ? Peu réaliste pour modéliser les timers*
- ▶ *Faible couplage temporel : Cumuler des annotations de temps*



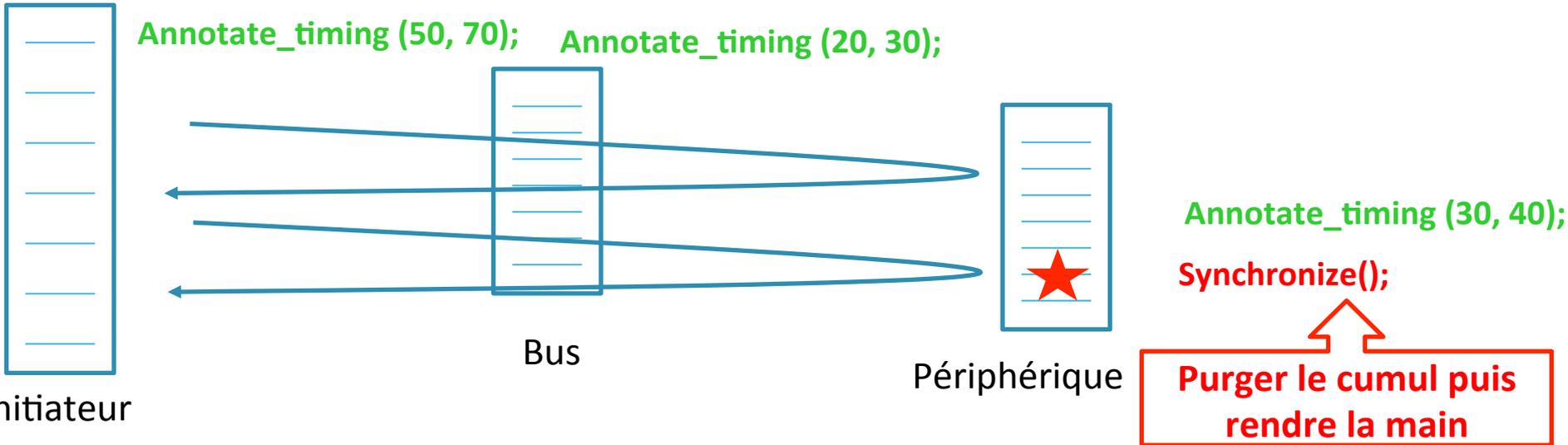
Le temps dans les prototype virtuels

- ❓ *Exécution en temps nul ? Peu réaliste pour modéliser les timers*
- ▶ *Faible couplage temporel : Cumuler des annotations de temps*



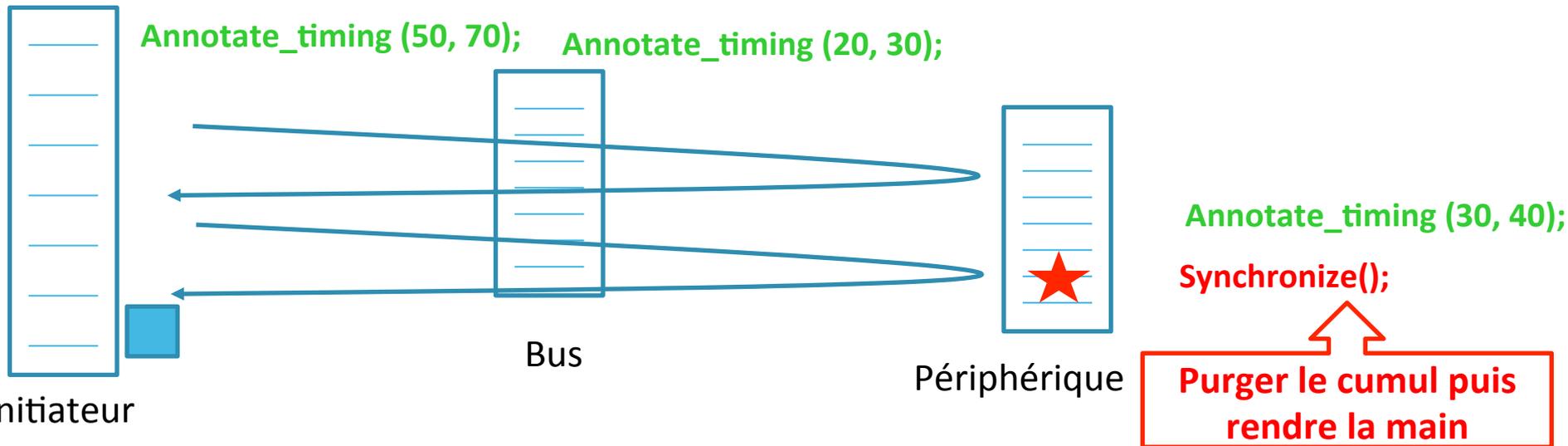
Le temps dans les prototype virtuels

- ❓ *Exécution en temps nul ? Peu réaliste pour modéliser les timers*
- ▶ *Faible couplage temporel : Cumuler des annotations de temps*



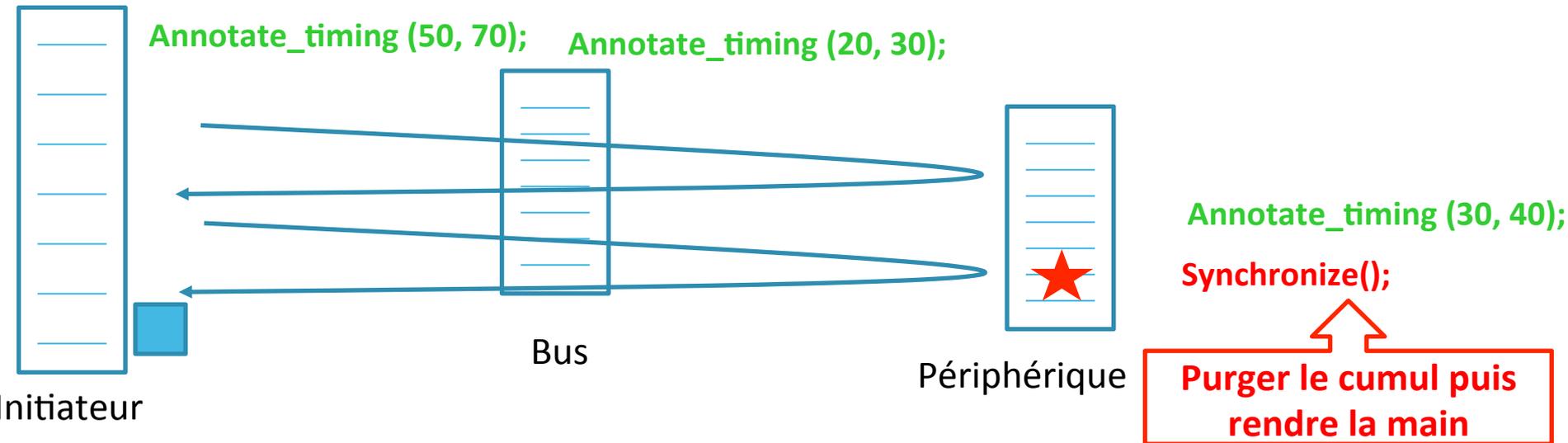
Le temps dans les prototype virtuels

- ❓ *Exécution en temps nul ? Peu réaliste pour modéliser les timers*
- ▶ *Faible couplage temporel : Cumuler des annotations de temps*



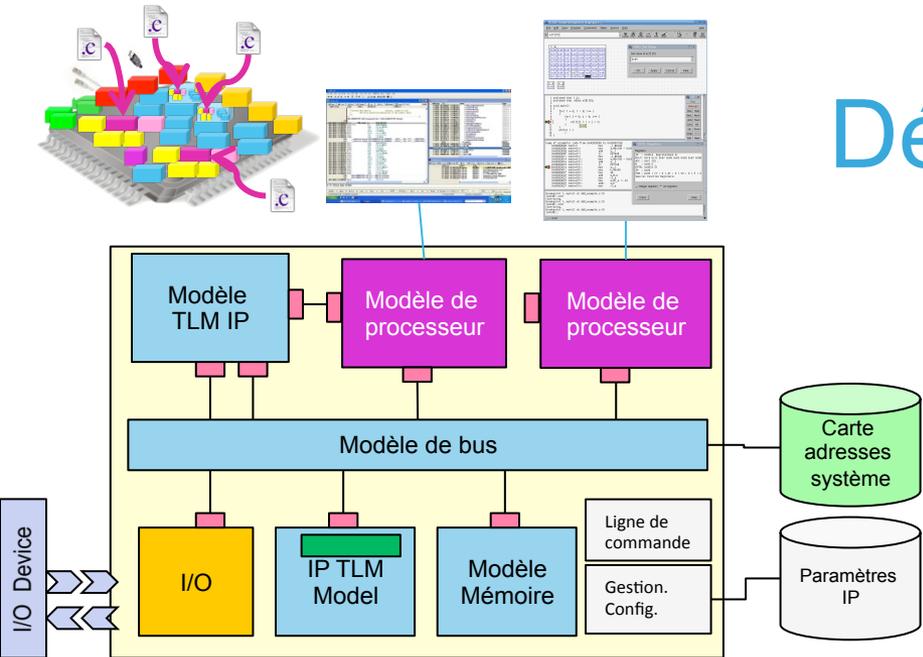
Le temps dans les prototype virtuels

- ❓ *Exécution en temps nul ? Peu réaliste pour modéliser les timers*
- ▶ *Faible couplage temporel : Cumuler des annotations de temps*

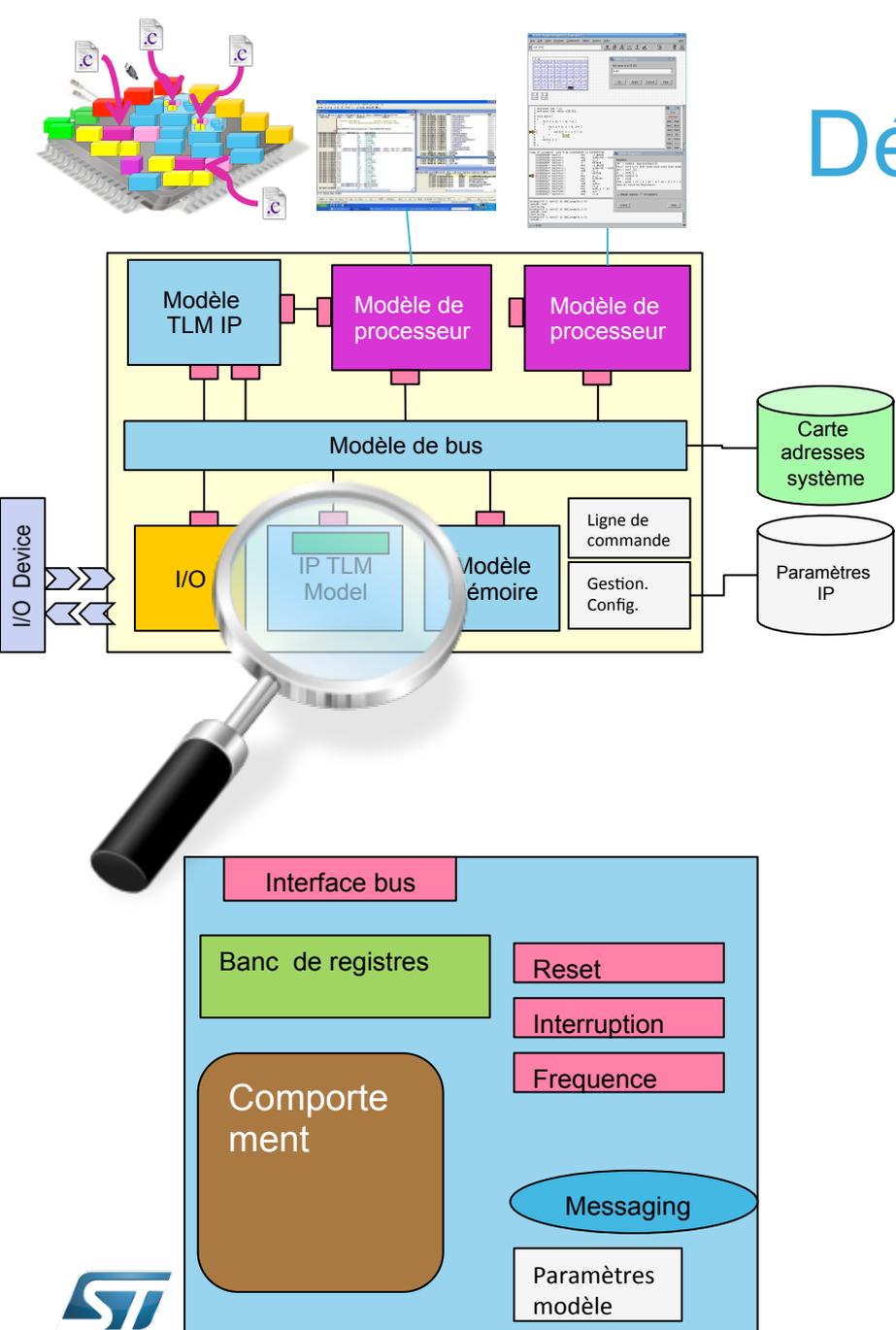


- ✔ *Cumul borné par les points de synchronisation*
- ✔ *Raisonner sur des intervalles, pas sur des valeurs absolues*

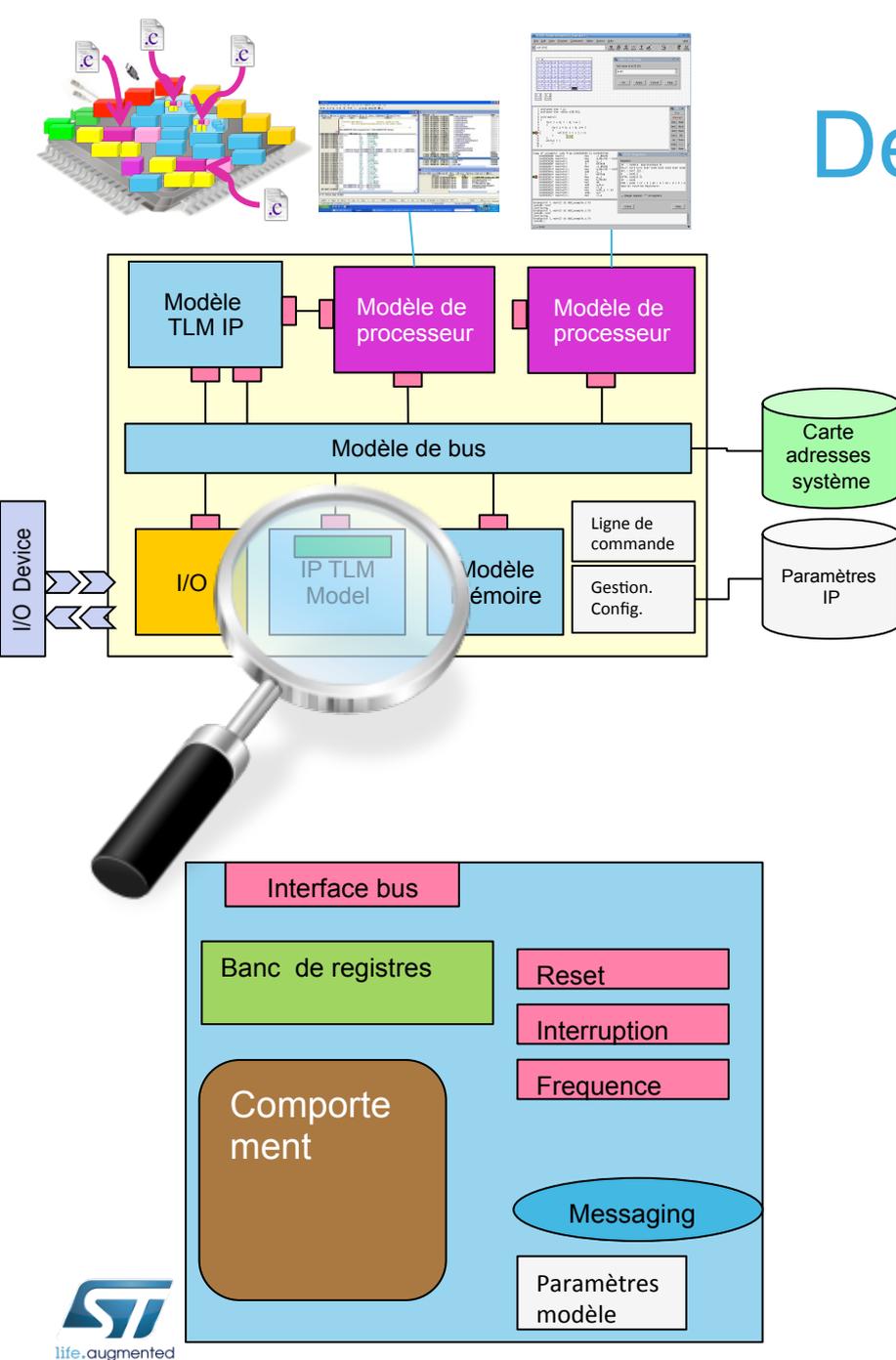
Déploiement industriel



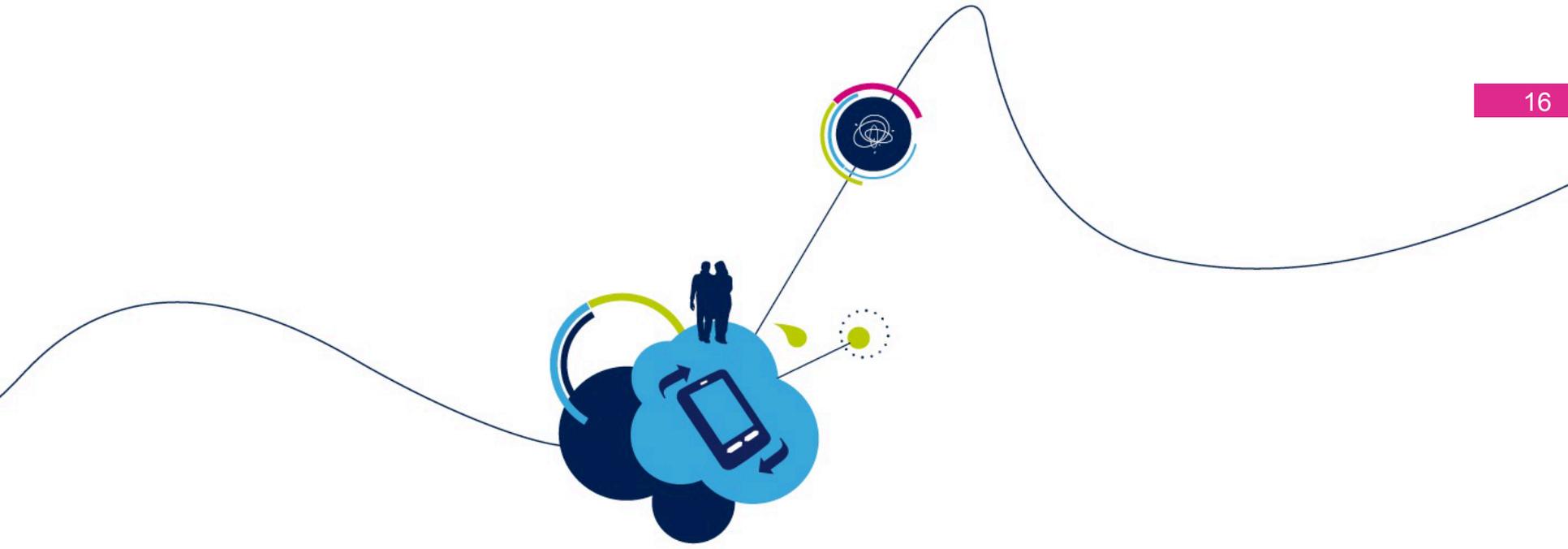
Déploiement industriel



Déploiement industriel



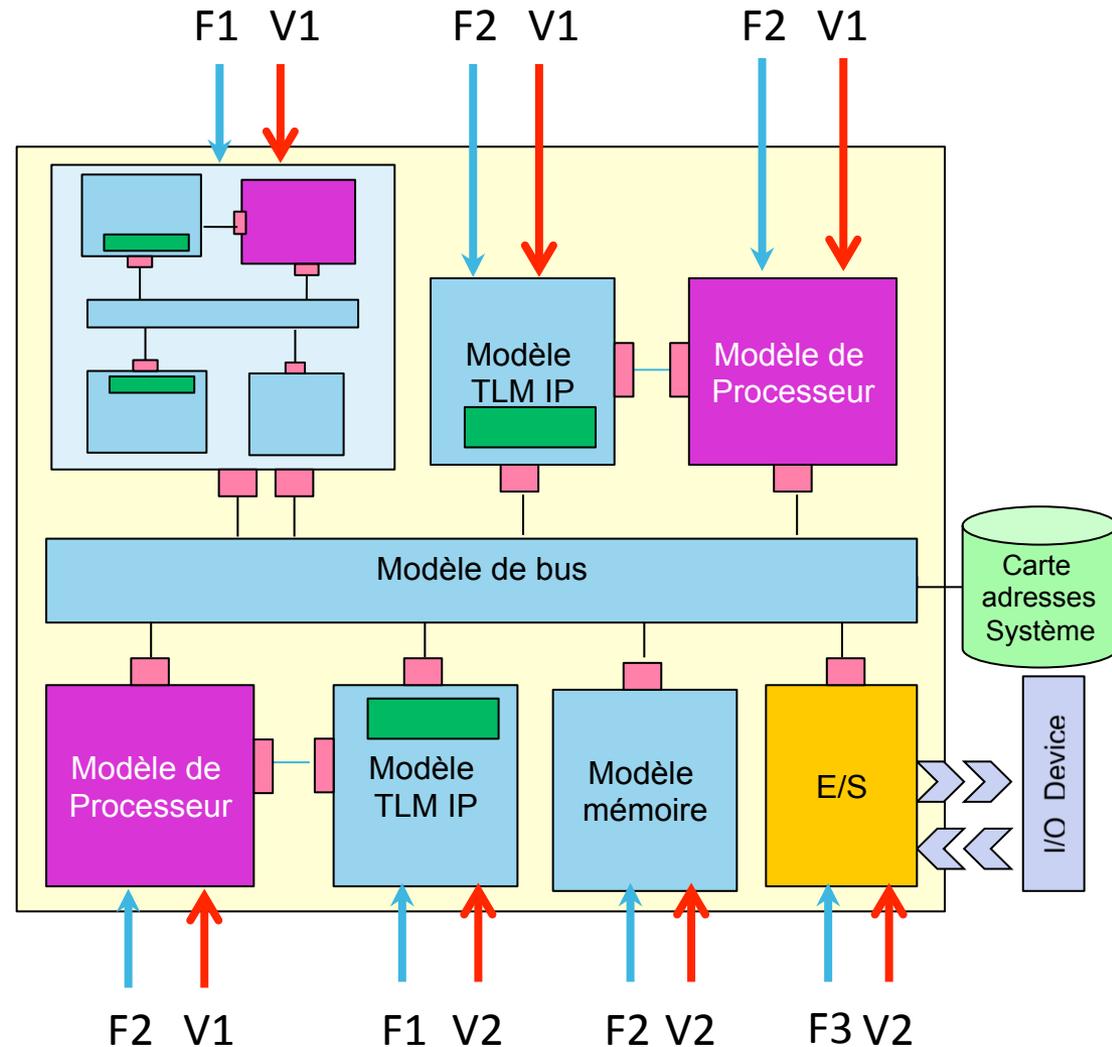
- Kit de modélisation
 - Primitives de modélisation
 - Protocoles de communication
 - Expertise et recommandations
- Bibliothèque de modèles génériques
- Déploiement sur les groupes produits
 - Depuis 2005
 - Nombreux prototypes virtuels développés
 - Grand nombre d'utilisateurs
- Expertise reconnue et contribution aux standards
 - Donation de propositions de standards
 - Participation active à la standardisation
 - OSCI TLM1 et TLM2, IEEE 1666



Emergence de nouveaux besoins

Modéliser horloges et alimentation

- Fréquences de fonctionnement
- Domaines d'alimentation
- Validation du logiciel
 - Modes fonctionnels
 - Points de fonctionnement



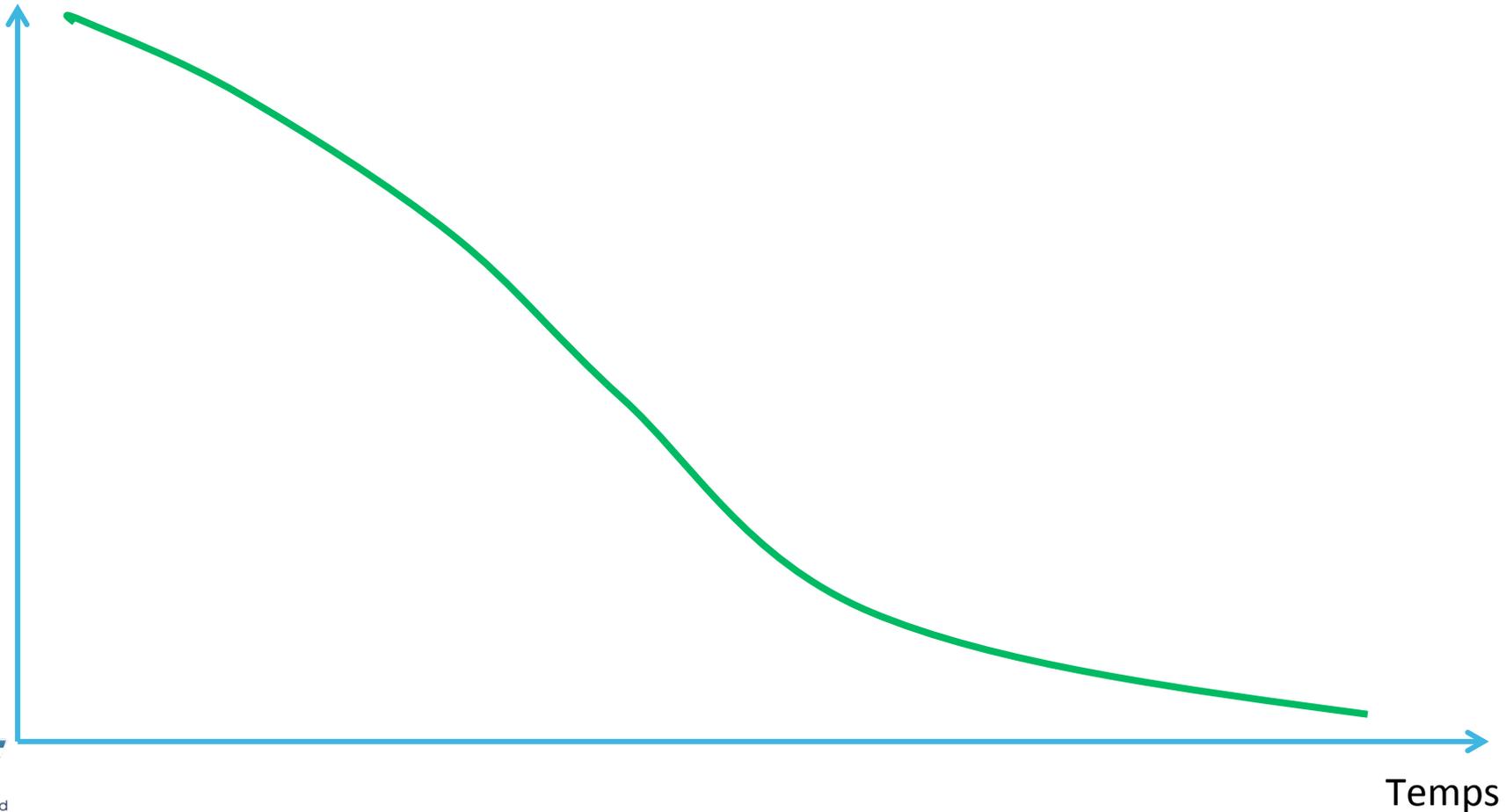
Interdépendance fonctionnel/extra-fonctionnel

Niveau de batterie



Gestion de la qualité de service

Gestion dynamique de la consommation



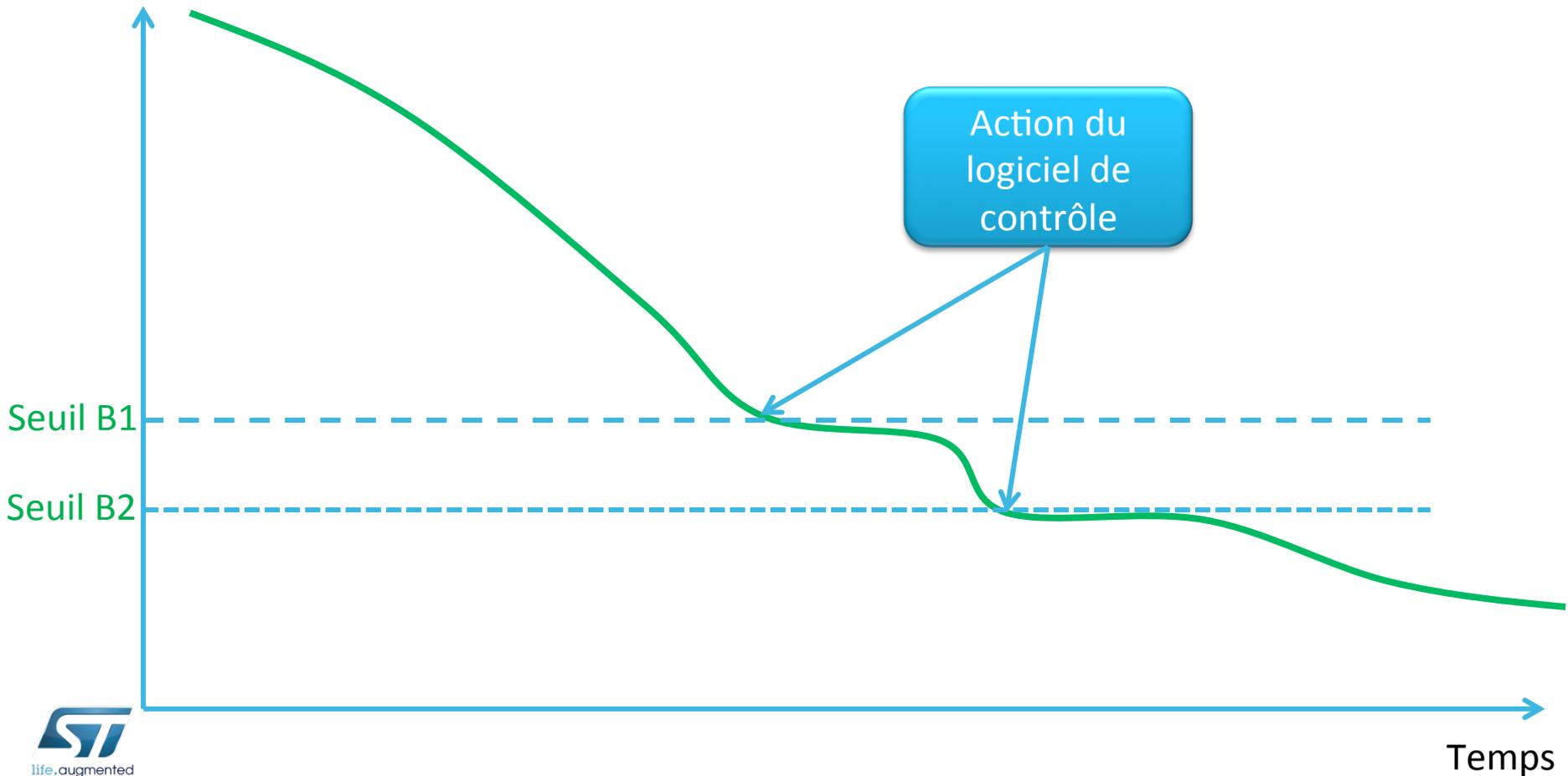
Interdépendance fonctionnel/extra-fonctionnel



Gestion de la qualité de service

Gestion dynamique de la consommation

Niveau de batterie



Interdépendance fonctionnel/extra-fonctionnel



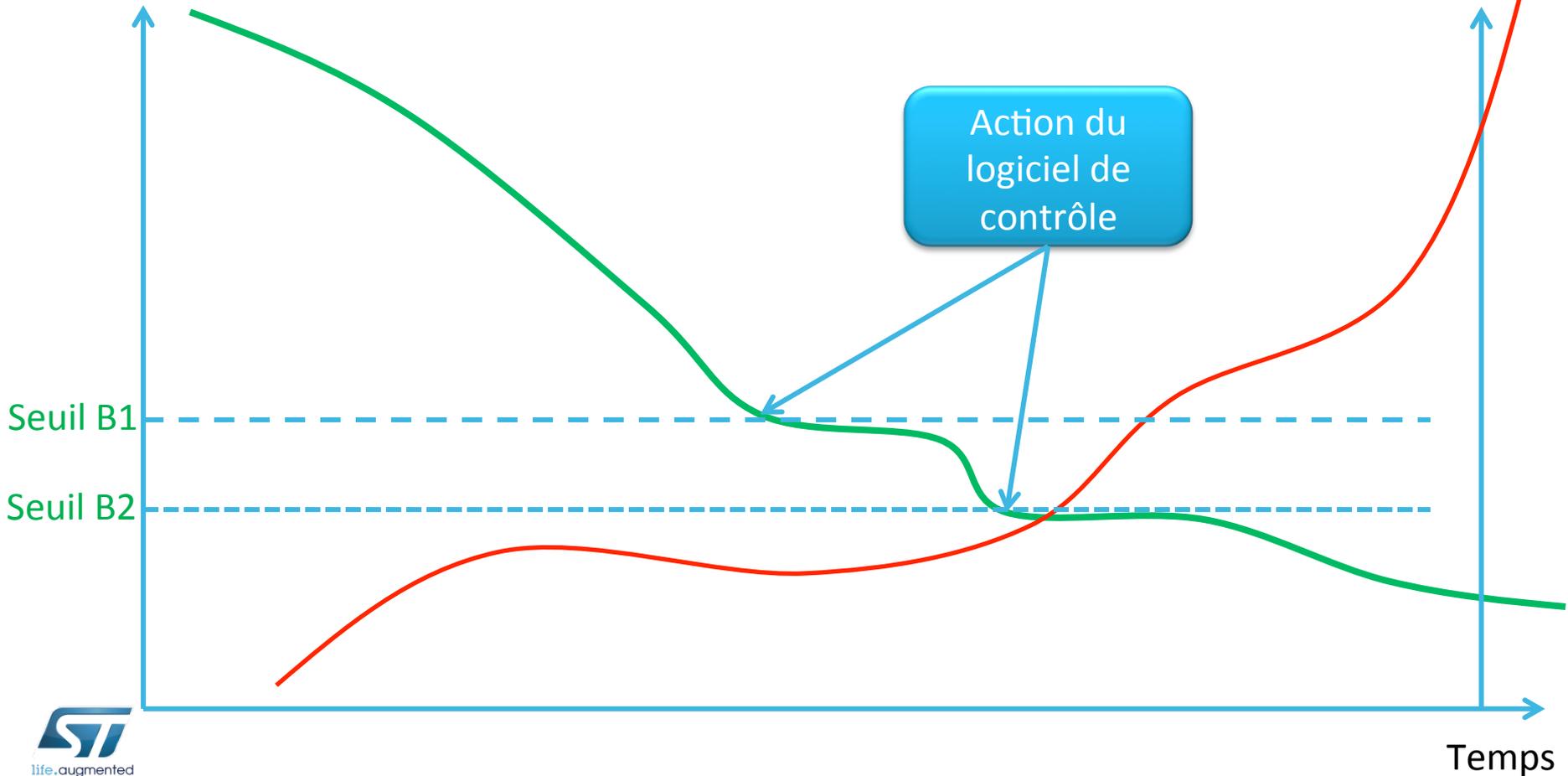
Gestion de la qualité de service

Gestion dynamique de la consommation

Niveau de batterie

Température

Action du logiciel de contrôle



Interdépendance fonctionnel/extra-fonctionnel

Niveau de batterie



Gestion de la qualité de service

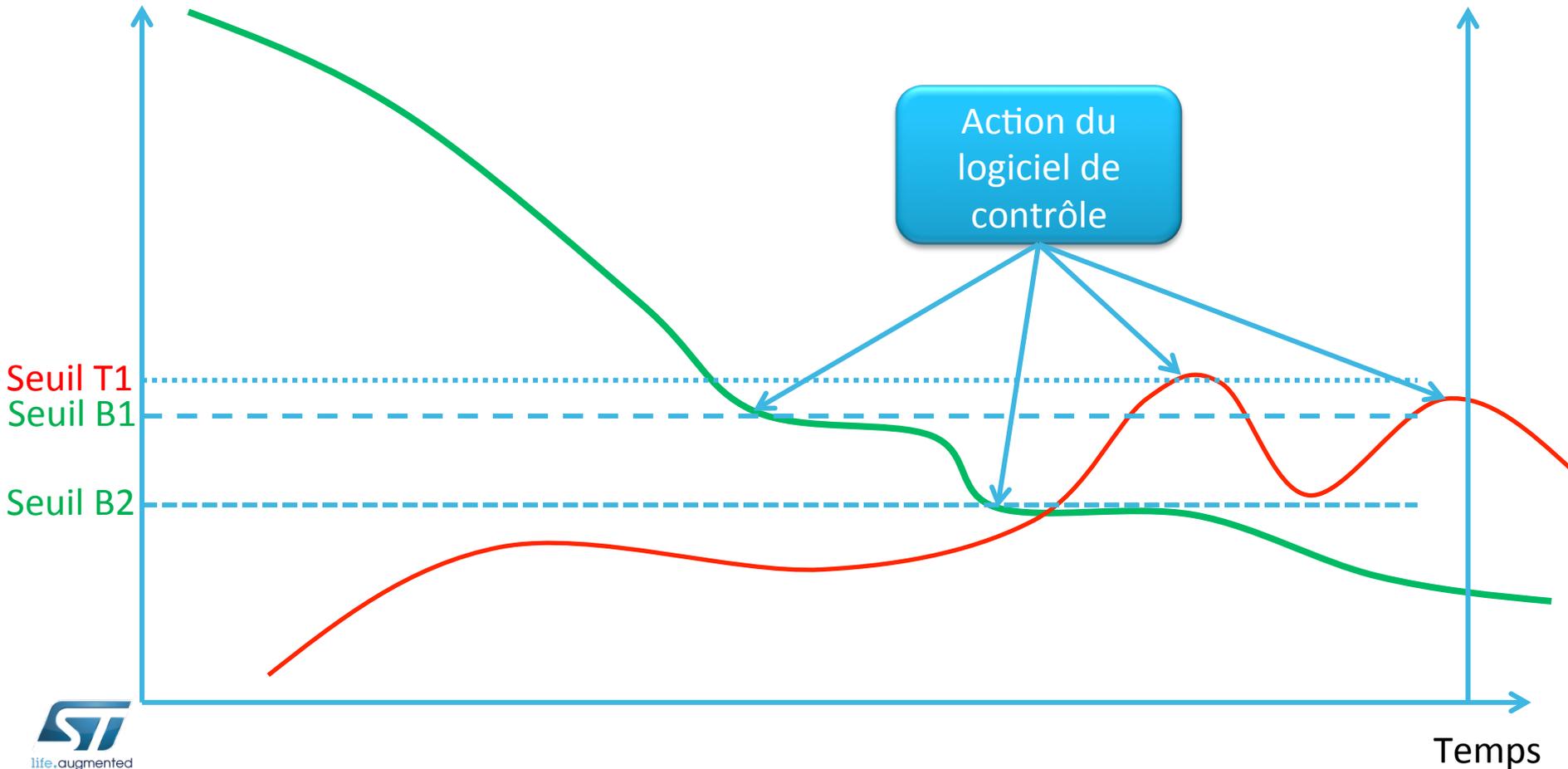
Gestion dynamique de la consommation



Gestion thermique dynamique

Température

Action du logiciel de contrôle



Interdépendance fonctionnel/extra-fonctionnel

Niveau de batterie



Gestion de la qualité de service

Gestion dynamique de la consommation

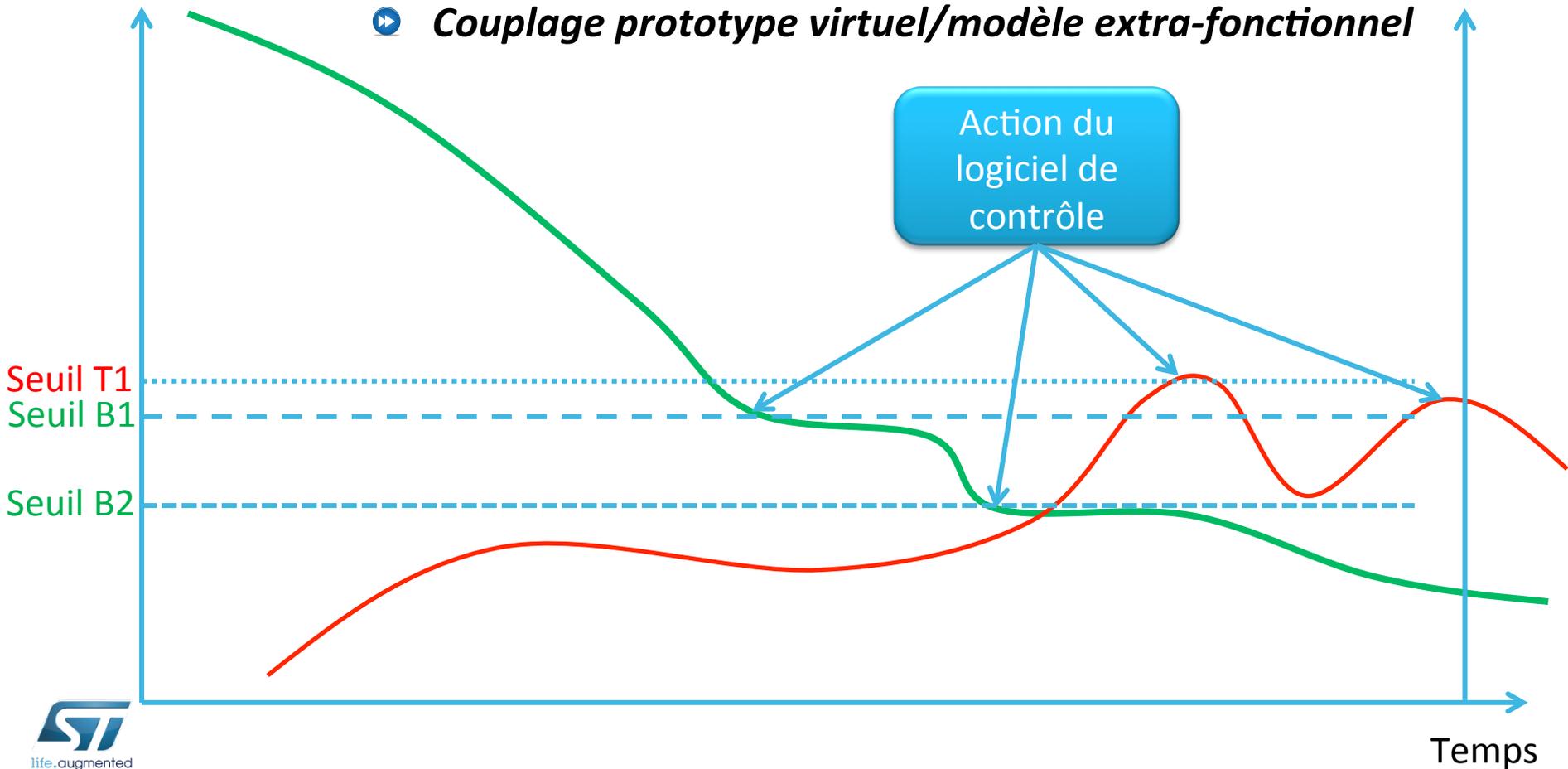


Gestion thermique dynamique

Temperature

▶ *Couplage prototype virtuel/modèle extra-fonctionnel*

Action du logiciel de contrôle



- Evolution des flots de conception
 - Approche de modélisation transactionnelle
- Déploiement industriel effectif
 - Retour d'expérience depuis 2005
 - Intégré aux projets de SoC complexes
- Coopération académie/industrie à STMicroelectronics depuis 10 ans
 - Avec l'écosystème français : ANR, Pôles de compétitivité, ...
 - Avec l'écosystème européen : MEDEA, CATRENE, ...
- Emergence de nouveaux besoins
 - Accélération des simulations
 - Modélisation des interactions fonctionnel/extra-fonctionnel