

Définition des critères pour plate-forme TLM-T

1. A1) Vue générale de la plate-forme SOCLIB
2. A2) Niveaux de modélisation SOCLIB
3. A3) Objectifs du document
4. B1) Introduction
5. B2) Exploration d'architecture
6. B3) Mise en place des éléments logiciels plus tôt dans le flot de ?
7. B4) Structuration logicielle
8. B5) Evaluation des performances
9. B6) Estimation de la consommation
10. C1) Introduction
11. C2) Pré-requis
12. C3) Vitesse d'exécution / accélération de la simulation
13. C4) Précision de l'évaluation et fidélité au modèle final
14. C5) Charge de développement du modèle (facilité d'écriture)
15. C6) Interopérabilité
16. C7) Facilité de réutilisation et d'intégration
17. C8) Disponibilité du modèle au plus tôt
18. C9) Parallélisation

A) Introduction

A1) Vue générale de la plate-forme SOCLIB

La plate-forme SoCLib adresse une classe particulière de systèmes embarqués : les systèmes multiprocesseurs intégrés sur puce (MP-SoC), qui sont utilisés dans pratiquement tous les secteurs industriels : télécommunications, vidéo et multimédia, automobile, transports, etc. Dans de nombreux secteurs industriels, les puces qui contrôlent les équipements sont les principaux facteurs de différenciation entre les produits, et le temps de conception et de développement d'un nouveau produit (time to market) joue un rôle décisif dans la compétition entre les industriels concernés. Ces puces sont désormais de véritables systèmes sur silicium fournissant une gamme étendue de fonctionnalités sur une surface de plus en plus petite.

Le projet SOCLIB a pour vocation de fournir les différents éléments nécessaires à une plate-forme de modélisation des systèmes sur silicium. Celle-ci doit inclure, outre des modèles de processeurs, tous les composants assurant la communication entre processeurs : micro-réseaux sur puce, système d'exploitation embarqué, intergiciels de communication, etc.

A2) Niveaux de modélisation SOCLIB

Pour rappel, la plate-forme SoCLib fournira **deux types de modèles de simulation**. Les modèles de niveau **CABA** (Cycle-Accurate & Bit-Accurate), permettent une évaluation précise des performances.

Les modèles de niveau **TLM-T** (Transaction Level Model with Timing) permettent une réduction sensible des temps de simulation (un ou deux ordres de grandeur) au prix d'une perte de précision mineure.

Ces modèles de simulation utiliseront le **langage SystemC** (<http://www.systemc.org>), dérivé du C++, bien adapté aux deux niveaux de modélisation retenus, qui est de plus en plus largement utilisé dans l'industrie européenne.

A3) Objectifs du document

Ce document a pour objectifs :

- de lister et de mettre en évidence les utilisations des différents partenaires du projet
- définir les critères retenus par les partenaires à partir de ces utilisations pour modéliser en TLM-T les composants.

B) Attentes par rapport aux modèles TLM-T

B1) Introduction

Il s'agit de définir pour l'ensemble des partenaires les différentes utilisations dans l'ordre d'importance pour le prototypage des SoCs en TLM-T.

B2) Exploration d'architecture

La première utilisation retenue est l'exploration d'architecture. Ces modèles de composants représentés avec le niveau TLMT doivent en faciliter l'exploration d'architecture.

Lors de l'exploration d'architecture, les concepteurs doivent disposer de différentes variantes de plates-formes de manière à déterminer la configuration architecturale la plus adaptée aux besoins du produit. Cela permet d'évaluer de façon fiable les performances d'une architecture matérielle par simulation de l'exécution effective du code binaire de l'application logicielle embarquée ou encore par l'utilisation de générateur de trafic pour ce qui concerne certaines fonctionnalités de type système (exemple : bus)

Par exemple, l'étude d'architecture inclue la mesure de la bande passante sur les bus, l'estimation de la taille des différentes files d'attente et buffers.

La modélisation TLMT doit fournir les mêmes informations (exemple : temps et consommation), mais en réduisant les temps de simulation d'un ordre de grandeur pour une perte de précision inférieure à 5%.

Cette utilisation a été mentionnée par le LIP6, Thales, Thomson, CEA-LIST et STMicroelectronics et l'IRISA.

B3) Mise en place des éléments logiciels plus tôt dans le flot de développement

Une plate-forme composée de briques programmées au niveau TLM-T doit fournir un ensemble fonctionnel (exemple : adressage mémoire, registres, mise en œuvre programmation) proche de la réalité. Cet environnement doit être suffisamment complet de manière à pouvoir développer puis implanter les différentes applications à mettre en œuvre. Cette plate-forme virtuelle devra par exemple permettre la mise en place des processus de synchronisation entre les différents éléments matériels et logiciels composant l'applicatif. Au niveau TLM-T, les estimations de temps ne sont évidemment pas aussi précises qu'au niveau CABA, mais elles doivent tout de même permettre la mise en place du logiciel.

Ce pré-développement du logiciel avec le modèle hardware doit inclure des modèles de processeur, les périphériques, les interfaces et modèles mémoires, les modèles d'interconnexion locale et globale.

Cette utilisation a été mentionnée par Thales, Thomson, CEA-LIST, STMicroelectronics et le CEA-LETI.

B4) Structuration logicielle

La grande diversité des plates-formes existant aujourd'hui oblige les développeurs logiciels à spécialiser leur code de manière à exploiter au maximum les capacités de la plate-forme. L'exploration logicielle doit ainsi permettre aux développeurs de structurer l'application de manière à utiliser au mieux les ressources disponibles: parallélisation, pipeline, utilisation des mémoires à différents niveaux et autres. Une plate-forme développée au niveau TLM-T doit donc disposer d'une structure fidèle à la plate-forme physique finale pour fournir aux développeurs les services nécessaires à la structuration de l'application. Il s'agit aussi d'avoir un même modèle de programmation ou le plus unifié possible pour les modèles TLM et TLMT afin de permettre une transition simple et naturelle entre les deux

Cette utilisation a été mentionnée par THALES, STMicroelectronics, TIMA, l'ENST, CEA-LIST, le CEA-LETI et TURBOCONCEPT.

B5) Evaluation des performances

Ce point découle du 1er point sur l'exploration d'architecture qui inclut généralement l'évaluation des performances comme un élément essentiel. Une plate-forme constituée de composants programmés au niveau TLMT doit permettre de travailler sur l'analyse des performances du circuit, sans trop pénaliser les temps de simulation. Si cette évaluation ne peut évidemment pas aussi précise qu'au niveau CABA, elle doit tout de même permettre aux développeurs de déterminer si la plate-forme sélectionnée est adaptée à leurs besoins. Il doit par exemple être possible d'évaluer la charge imposée sur les processeurs et d'évaluer la contention sur les éléments de communication. Cette première évaluation des performances permettra ainsi de retenir ou non la plate-forme et de vérifier si les contraintes temporelles de l'application peuvent être respectées. Il est possible dans ce cas de comparer différentes architectures, mais aussi à architecture constante différents déploiements de l'application logicielle sur la cible.

Cette utilisation a été mentionnée par Thales, Thomson, STMicroelectronics et le CEA-LETI.

B6) Estimation de la consommation

Ce point peut aussi découler du 1er point sur l'exploration d'architecture qui peut inclure l'évaluation de la consommation comme un élément déterminant dans le choix de l'architecture.

Cette utilisation a été mentionnée par le LIP6, le CEA-LETI et STMicroelectronics.

C) Critères d'écriture et de fonctionnement des modèles TLM-T

C1) Introduction

Il s'agit de définir pour l'ensemble des partenaires les différents critères retenus dans l'ordre d'importance et découlant de l'utilisation pour le prototypage des SoCs en TLM-T.

C2) Pré-requis

L'ensemble des partenaires a demandé un pré-requis sur la capacité d'exécution d'un même code sur différents niveaux de modélisation. Ainsi, les méthodologies et solutions mises en œuvre pour le TLM-T doivent permettre d'exécuter le même code applicatif ou autre (y compris système d'exploitation avec migration de tâches, etc.) qu'avec une représentation utilisant un autre niveau de modélisation (exemple : de type CABA).

C3) Vitesse d'exécution / accélération de la simulation

La vitesse d'exécution des modèles de composants au niveau TLM-T doit être suffisamment élevée pour permettre aux concepteurs de logiciels et de matériels (exemple : développeurs d'IPs et d'application) de faire exécuter fréquemment de nombreux tests.

Justification : il s'agit d'offrir des vitesses de simulation bien plus élevées que celles obtenues avec les autres niveaux de modélisation (exemple, CABA).

Ce critère a été mentionné par Thales, LIP6, STMicroelectronics, Turboconcept, TIMA, l'ENST, l'IRISA, SILICOMP, UNIVERSITE BRETAGNE SUD, CEA-LETI, LISIF.

C4) Précision de l'évaluation et fidélité au modèle final

Les modèles TLM-T doivent pouvoir servir de référence fonctionnel dans le flot de conception des systèmes, du moins jusqu'à ce que le choix se porte sur un autre niveau de modélisation. Par référence fonctionnel, nous entendons un niveau et une précision de représentation vraie au niveau « bit » lors d'une écriture en mémoire (écriture fonctionnellement juste par rapport à des attendus). Il ne s'agit pas d'être exact au niveau cycle d'un point de vue temporel donc les modèles de composants de niveau TLM-T ne peuvent très certainement pas être aussi précis que les modèles de niveau CABA. Cependant, il est important que la perte de précision temporelle soit la plus faible possible et que ces modèles permettent de structurer et de dimensionner les parties logicielles (c'est-à-dire de s'assurer que certaines contraintes de taille d'application, de fidélité à des résultats escomptés et à des temps d'exécution soit respectées avec une erreur de moins de 20 %). La perte de précision ne devrait pas induire plus de 5 % sur un modèle et plus de 20% sur un SOC.

Note : ce point est à rediscuter car une perte de un 1 % sur un seul modèle peut se traduire par une perte de précision importante (> 20%) sur le SOC.

Ce critère a été mentionné par Thales, LIP6, STMicroelectronics, Turboconcept, TIMA, l'ENST, l'IRISA, UNIVERSITE BRETAGNE SUD, CEA-LETI, LISIF.

C5) Charge de développement du modèle (facilité d'écriture)

Les règles de modélisation du modèle TLM-T doivent permettre de limiter le plus possible la charge de développement des modèles TLM-T, c'est-à-dire faciliter l'écriture rapide des modèles. Une dérivation d'un niveau de modélisation vers un autre peut être appréciable aussi.

Ce critère a été mentionné par TIMA, ENST, MAGILLEM DESIGN SERVICES, CEA-LIST, IRISA, TURBOCONCEPT, LIP6, SILICOMP, UNIVERSITE BRETAGNE SUD, LISIF.

C6) Interopérabilité

Il s'agit de s'assurer que les modèles sont intégrables dans une autre plate-forme quel(s) que soi(en)t le(s) niveau(x) de représentation utilisé(s) dans cette plateforme, le plus facilement possible, c'est-à-dire avec une charge de travail minimum (ou modification minimale). Il faut donc que le modèle « timé » soit écrit d'une façon facilitant son utilisation avec d'autres modèles qui n'ont pas forcément le même niveau d'abstraction

Justification : ce point est important car avec l'exploration d'architecture, les concepteurs cherchent à déterminer quelle est l'architecture la mieux adaptée à leurs besoins. Ce processus de recherche utilise un prototype virtuel de la plate-forme constitué de composants communiquant entre eux via des modèles d'interconnexion, composants

pouvant avoir été modélisés dans d'autres équipes ou selon des niveaux de précision différents. Ces composants de niveau TLM-T doivent donc respecter des normes de communications standards pour être facilement interopérables. Ils doivent être facilement intégrables avec peu de modification dans un flot SystemC (SocLib ou non SocLib). A ce titre, le degré de maturité de la modélisation est aussi important.

Ce critère a été mentionné par Thales, LIP6, SILICOMP, CEA-LIST, STMicroelectronics, LISIF.

C7) Facilité de réutilisation et d'intégration

Il s'agit de pouvoir :

- *réutiliser tels quels (sans modification) et le plus rapidement possible les modèles dans une autre plate-forme, le plus souvent avec un niveau d'abstraction égal ou éventuellement incluant des niveaux d'abstraction différents si cette plate-forme offre déjà les convertisseurs adéquats*
- *intégrer rapidement ces modèles en développant si nécessaire par ailleurs (sans toucher aux modèles importés) les interfaces ou les convertisseurs nécessaires à leur intégration dans la plate-forme.*

Justification : un composant TLM-T doit être spécifié/conçu/décrit par son concepteur pour pouvoir être rapidement réutilisé et intégré par d'autres. Par exemple :

- ne pas avoir à se plonger dans le code détaillé,
- comprendre rapidement l'interface de communication
- maintenance peu coûteuse en temps
- qualification rapide des modèles TLM-T

Ce critère a été mentionné par Thales, TIMA, ENST, Turboconcept, MAGILLEM DESIGN SERVICES, CEA-LIST, IRISA, LIP6, UNIVERSITE BRETAGNE SUD, CEA-LETI.

C8) Disponibilité du modèle au plus tôt

Les règles de modélisation du modèle TLM-T doivent permettre au modèle d'être disponible au plus tôt dans le cycle de développement.

Justification : les modèles de composants TLM-T doivent permettre aux designers d'effectuer des tests tôt dans le cycle de développement. Le processus réel est de commencer par le PV pour faire éventuellement après du PVT.

Ce critère a été mentionné par SILICOMP, Thales, TIMA, ENST, Turboconcept, MAGILLEM DESIGN SERVICES, CEA-LETI.

C9) Parallélisation

Le modèle TLM-T doit être écrit et pouvoir être exécuté de manière à ce que la charge de travail nécessaire pour l'utiliser dans un contexte de simulation parallèle sur stations de travail multi-cores (machines SMP) soit minimum.

Ce critère a été mentionné par LIP6 et CEA-LIST

D) Tableau de synthèse et priorités

N°	Critère	Priorité (Nombre de partenaires citant le critère)
1	Vitesse d'exécution	12

3	Charge de développement	11
5	Facilité de réutilisation	10
2	Précision de l'évaluation et fidélité au modèle final	10
6	Disponibilité du Modèle	7
4	Interopérabilité	6
7	Parallélisation	2