

Démarche d'exploration d'architectures pour le dimensionnement d'un terminal mobile LTE

ANTHONY BARRETEAU, SEBASTIEN LE NOURS, OLIVIER PASQUIER

Univ Nantes, IREENA, EA1770
Polytech-Nantes, rue C. Pauc, Nantes, F-44000 France

{anthony.barreateau, sebastien.le-nours, olivier.pasquier}@univ-nantes.fr

Résumé – L'utilisation d'architectures multiprocesseurs pour la conception des systèmes embarqués complexifie le travail de dimensionnement, sous contraintes de temps, de consommation et de coût des ressources matérielles et logicielles requises. Nous présentons dans cet article une démarche visant à faciliter le processus de dimensionnement des architectures matérielles et logicielles des systèmes embarqués. Ces travaux reposent sur l'utilisation de modèles exécutables de niveau transactionnel intégrant les propriétés non fonctionnelles d'un système. Les modèles ainsi créés peuvent être simulés afin d'analyser les performances obtenues et de comparer rapidement les architectures étudiées. Nous illustrons ce travail dans le contexte des systèmes de radiocommunication de future génération.

Abstract – In embedded systems design, multi-core architectures are required to fully meet related power consumption, cost, and time constraints. Efficient methods are necessary to allow early performance evaluation and fast exploration of the design space. Our work focused on defining an approach to guide systems architect for the use of executable transaction-level models. Functional and non-functional properties required for performance analysis of different architectural solutions are described in proposed models. Results obtained by simulation are used to evaluate and compare expected performances. The benefits of the proposed approach are highlighted through the analysis of two architectures of a communication receiver supporting the physical layer of the LTE protocol.

1 Introduction

Compte tenu des contraintes de temps, de coût, de consommation et de surface, l'accroissement de la complexité des traitements à mettre en œuvre pour la conception des terminaux mobiles nécessite l'utilisation d'architectures multiprocesseurs [1]. Dans ce contexte, le processus d'exploration architecturale consiste à dimensionner les ressources de calcul, de mémorisation et de communication requises de manière à satisfaire les contraintes fonctionnelles et non fonctionnelles associées au système à concevoir. Compte tenu de l'influence des décisions prises lors de cette étape sur les coûts de développement, il devient nécessaire de réaliser efficacement ce travail le plus tôt possible dans le processus de développement [2].

L'exploration de l'espace de conception nécessite l'utilisation de modèles d'architectures afin de guider le concepteur dans ses choix [3]. Afin de comparer rapidement différentes solutions possibles, les modèles d'architectures utilisés doivent être définis à un niveau de représentation approprié de manière à offrir un bon compromis entre la précision des résultats et le temps de simulation. Il s'agit pour cela d'accroître le niveau d'abstraction des modèles utilisés pour mener cette exploration. Le paradigme TLM (*Transaction-Level Modeling*) a ainsi récemment émergé afin de faciliter le travail d'exploration et de mise au point des systèmes MPSoC (*Multiprocessor System on Chip*) [4]. Élevant le niveau de description au dessus du niveau RTL, les modèles de niveau transactionnel offrent ainsi un bon

compromis entre niveau de précision et vitesse de simulation [5]. Les approches proposées pour la création de modèles de niveau transactionnel reposent généralement sur le paradigme en Y formalisé par D. Gajski [6] et sur l'approche orientée plate-forme [3]. En suivant cette approche, une description du système est obtenue en allouant les différents éléments de la vue applicative sur les différents éléments de la plate-forme. La description obtenue peut alors être analysée analytiquement ou par simulation. La capture et la simulation des modèles s'effectuent en utilisant des langages tels que SystemC et SystemVerilog. Les résultats de simulation obtenus permettent au concepteur d'évaluer et de comparer différentes alternatives d'architectures.

Dans ce contexte, nos travaux portent sur la définition d'une approche pour accompagner le concepteur dans le processus de création de modèles de niveau transactionnel en vue de l'exploration d'architectures. Ce papier évoque la démarche d'exploration considérée. Parmi les travaux existants, l'approche proposée dans [7] vise à permettre l'évaluation de propriétés non fonctionnelles telles que la consommation et la température. La description de l'application est définie à partir d'un modèle appelé « *communication dependency graph* ». A cette description est associé un modèle prenant en compte la consommation induite par les ressources. Une évaluation de la consommation est alors obtenue par simulation. Les deux approches présentées dans [8] et [9] associent une description UML2 de

l'application à un modèle de plate-forme décrit en SystemC pour l'évaluation de performances. Dans l'approche présentée dans [8], un méta-modèle est défini afin de guider le concepteur lors du processus de création des modèles de l'application et de la plate-forme. Les diagrammes d'activités UML2 sont ici utilisés pour capturer ces modèles. Une fois les fonctions de l'application allouées sur les ressources de la plate-forme, une description en langage SystemC est alors générée automatiquement. La simulation du modèle obtenu permet alors au concepteur d'évaluer les performances pouvant être obtenues avec l'architecture étudiée. Dans [9], des modèles de charge de calcul associés aux différentes fonctions de l'application sont définis. Trois techniques d'extraction d'information sont proposées pour obtenir ces modèles: analytiquement, par mesure ou à partir du code source associé. Ces modèles sont ensuite alloués sur le modèle de la plateforme et le modèle résultant est simulé au niveau transactionnel pour évaluer les performances obtenues.

L'approche proposée dans ce papier est différente de celles présentées précédemment dans la façon de modéliser l'architecture et les propriétés non fonctionnelles analysées. Dans la section 2, nous décrivons la démarche d'exploration proposée pour la création d'un modèle unique permettant d'évaluer et de comparer les performances obtenues selon les architectures étudiées. Cette approche est basée sur un paramétrage des propriétés temporelles du modèle utilisé compte tenu de l'architecture analysée et des contraintes à respecter. La section 3 illustre l'intérêt de notre approche à travers un cas d'étude. La conclusion et les perspectives associées à ces travaux sont présentées dans la section 4.

2 Démarche d'exploration d'architectures considérée

La démarche d'exploration considérée consiste à paramétrer un modèle de performances exprimant l'utilisation qui est faite des ressources de calcul, de communication et de mémorisation pour réaliser les fonctions de l'application. Le paramétrage permet d'adresser différentes architectures possibles. On entend par architecture une allocation des fonctions sur les ressources de la plate-forme. Le paramétrage du modèle s'effectue de manière à exprimer le degré de parallélisme offert par l'architecture. Ce paramétrage porte sur les caractéristiques temporelles liées à l'architecture compte tenu des contraintes imposées. Les architectures étudiées peuvent alors être comparées en analysant les résultats obtenus par simulation. Le principe de cette démarche est illustré sur la figure 1.

Modèle utilisé pour l'exploration d'architectures

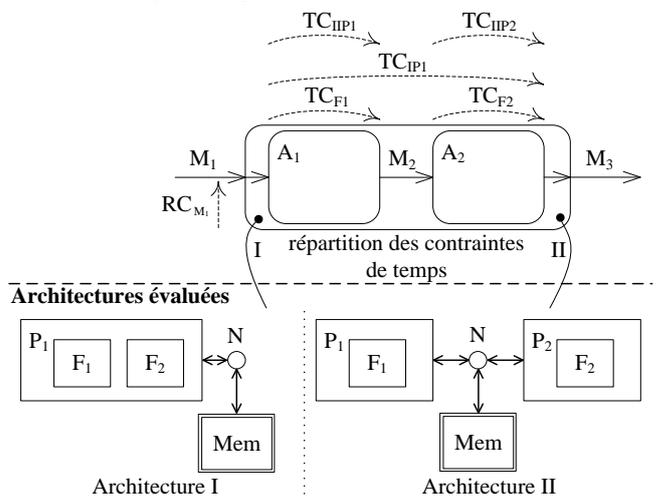


Figure 1 : Démarche d'exploration d'architectures proposée

La partie basse de la figure 1 présente deux architectures possibles selon l'allocation des fonctions F_1 et F_2 sur des ressources de traitement dénommées P_1 et P_2 . P_1 peut être considéré comme un processeur logiciel et P_2 comme une ressource matérielle spécifique. La partie haute de la figure 1 présente le modèle de performances utilisé pour analyser ces deux architectures. Ce modèle est décrit sous la forme d'un diagramme d'activité exprimant la structure de l'application et les relations entre les fonctions. Sur la partie haute de la figure 1, RC_{M1} représente le débit d'information que le système doit supporter en entrée. TC_{IP1} , TC_{IIP1} et TC_{IIP2} représentent les temps maximums d'exécution des traitements à respecter avec les architectures I et II. Selon la notation adoptée, TC_{IP1} indique le temps maximum pour l'exécution séquentielle des traitements associés aux fonctions F_1 et F_2 sur un processeur logiciel P_1 . TC_{IIP1} et TC_{IIP2} correspondent respectivement aux temps maximums pour l'exécution des traitements associés à F_1 sur P_1 et à F_2 sur P_2 .

La figure 2 présente la notation graphique adoptée pour décrire le comportement associé à chaque activité.

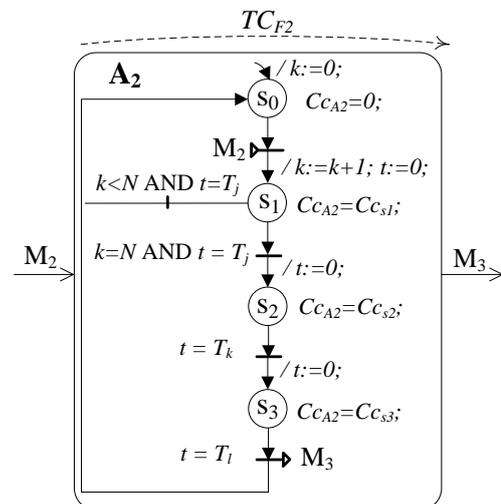


Figure 2 : Modèle de comportement adopté

Le comportement exprimé permet de décrire les conditions d'attente sur les relations d'entrée et de production sur les relations de sortie. Il permet aussi d'exprimer l'utilisation qui est faite des ressources pour exécuter chaque fonction pendant l'intervalle de temps qui sépare la réception et la production de transactions. Les transitions entre les états dénotés S_x peuvent être associées à l'attente ou à la production d'une transaction, à une condition temporelle ou à une condition logique liée à une variable interne. Sur la figure 2, l'état S_0 représente un état d'attente sur la relation d'entrée M_2 . La condition logique associée à la variable interne k permet d'assurer la réception de N transactions M_2 . L'utilisation de la ressource de traitement P_2 induite par l'exécution de F_2 est modélisée par l'évolution du paramètre dénoté Cc_{A2} . Durant les états S_1 , S_2 et S_3 , Cc_{A2} prend respectivement les valeurs associées à Cc_{s1} , Cc_{s2} et Cc_{s3} . Cc_{s1} , Cc_{s2} et Cc_{s3} représentent le nombre d'opérations de traitement réalisées. Le nombre d'opérations peut être exprimé à partir d'une expression analytique. Il peut ainsi varier en fonction des données reçues à traiter. T_j , T_k et T_l représentent respectivement les durées d'exécution des traitements associés aux états S_1 , S_2 et S_3 . Ces durées de traitement sont paramétrables et doivent être fixées de manière à satisfaire les contraintes de temps imposées. Le résultat est produit via la relation M_3 .

Dans [10], nous présentons un modèle d'exécution générique qui a été proposé afin de faciliter la capture des modèles utilisés. Ce modèle sert de base pour la constitution de modèles d'architectures. La simulation permet d'observer l'évolution de la puissance de calcul requise au cours du temps en fonction de l'architecture analysée et des données à traiter. En effectuant différents paramétrages temporelles des activités composant le modèle d'architecture, le concepteur peut ainsi comparer les performances pouvant être obtenues en analysant les différents résultats de simulation. Il dispose ainsi d'informations qui vont lui permettre de le guider dans ses choix.

3 Modélisation et dimensionnement d'un récepteur LTE

3.1 Modèle d'évaluation des performances obtenu

Le cas d'étude présenté se rapporte au dimensionnement des ressources de calcul requises pour mettre en œuvre les différents traitements associés à la couche physique du standard de communication LTE [11]. Nous considérons ici la partie réception associée à ce standard au niveau d'un terminal mobile avec une configuration à une seule antenne. La figure 3 présente le modèle utilisé pour comparer les performances obtenues avec les deux architectures considérées.

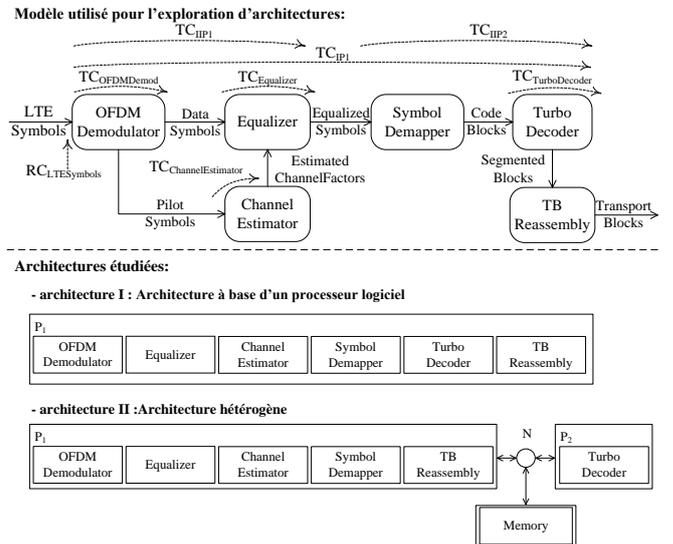


Figure 3 : Modèle d'exploration d'architectures du récepteur LTE

La partie basse de la figure 3 décrit les deux architectures étudiées. L'architecture I repose sur l'utilisation d'un processeur logiciel P_1 pour la réalisation de l'ensemble des traitements. L'architecture II repose aussi sur l'utilisation d'un processeur logiciel mais auquel il a été associé une ressource matérielle dédiée à la réalisation d'un traitement. La partie haute de la figure 3 présente le modèle défini pour analyser et comparer les performances obtenues avec ces deux architectures. Dans le mode de fonctionnement considéré, la chaîne de réception de la couche physique du standard LTE nécessite la mise en œuvre de six fonctions de traitement. Une activité est associée à chacune de ces fonctions de traitement. Les relations entre activités expriment les dépendances de données. La chaîne de réception reçoit toutes les millisecondes une trame radio composée de 14 symboles OFDM et produit en sortie un bloc de données qui est ensuite traité par les couches supérieures du protocole. La relation $LTESymbols$ représente la réception d'un symbole OFDM. La contrainte de temps en entrée dénommée $RC_{LTESymbols}$ est de 71428 ns. La relation $TransportBlocks$ se rapporte au bloc de données produit à la fin des traitements réalisés sur une trame radio. $TC_{OFDMDemod}$, $TC_{ChannelEstimator}$, $TC_{Equalizer}$ et $TC_{TurboDecoder}$ représentent les contraintes de temps associées à chaque fonction de traitement analysée. TC_{IP1} représente le temps d'exécution maximum à respecter si l'on considère l'architecture I avec l'ensemble des traitements réalisés sur un processeur logiciel P_1 . TC_{IP1} et TC_{IP2} représentent les temps de traitements maximums si l'on considère l'architecture II avec une partie des traitements s'exécutant cette fois-ci sur une ressource matérielle spécifique P_2 .

3.2 Paramétrage du modèle d'évaluation des performances

Le tableau 1 présente la répartition des contraintes de temps selon les deux architectures considérées.

Tab 1 : Répartition des contraintes de temps selon l'architecture considérée

	Architecture I	Architecture II
$TC_{OFDMDemod}$	$k_1 * TC_{IP1}$	$k_1' * TC_{IIP1}$
$TC_{ChannelEstimator}$	$k_2 * TC_{IP1}$	$k_2' * TC_{IIP1}$
$TC_{Equalizer}$	$k_3 * TC_{IP1}$	$k_3' * TC_{IIP1}$
$TC_{TurboDecoder}$	$k_4 * TC_{IP1}$	$k_4' * TC_{IIP2}$

Pour l'architecture I, l'ensemble des traitements peuvent être exécutés pendant l'intervalle de temps TC_{IP1} . Les coefficients k_1 , k_2 , k_3 et k_4 permettent de répartir cette contrainte de temps entre les différentes activités. Dans le cas de l'architecture II, le comportement est modifié afin de tenir compte du degré de parallélisme offert. Les traitements associés à l'activité TurboDecoder peuvent alors s'effectuer pendant les traitements associés aux activités ChannelEstimator et Equalizer.

3.3 Résultats obtenus

Le modèle proposé est ensuite utilisé pour générer un modèle SystemC de niveau transactionnel permettant d'évaluer par simulation les performances obtenues avec l'architecture analysée. La figure 4 donne un exemple de résultats de simulation obtenu avec l'architecture II en appliquant les paramètres suivants : $k_1'=0.01$, $k_2'=0.5$, $k_3'=0.49$ et $k_4'=0.24$.

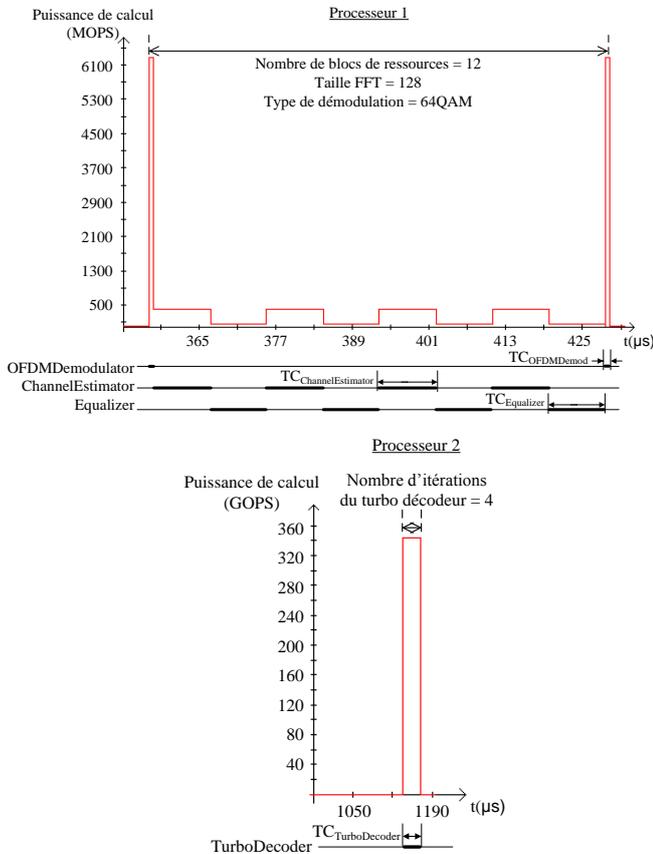


Figure 4 : Observation de la puissance de calcul requise avec l'architecture II pour un exemple de paramétrage

La figure 4 permet d'observer une partie de l'évolution au cours du temps de la puissance de calcul requise par P_1 et P_2 pour exécuter les traitements qui leur ont été alloués. En testant différentes répartitions de

contraintes de temps sur $TC_{OFDMDemod}$, $TC_{ChannelEstimator}$, $TC_{Equalizer}$ et $TC_{TurboDecoder}$, il est possible de comparer différents taux d'utilisation des ressources de calcul considérées pour une architecture donnée. De plus, il est possible de comparer ces résultats avec d'autres résultats obtenus en effectuant un paramétrage des propriétés temporelles du modèle pour des solutions architecturales alternatives.

4 Conclusion et perspectives

Dans ce papier, nous avons présenté une démarche d'exploration visant à réduire l'effort de modélisation nécessaire à l'évaluation de différentes solutions architecturales. Par simple paramétrage des propriétés temporelles du modèle, le concepteur peut analyser l'évolution au cours du temps de la puissance de calcul nécessaire selon l'architecture étudiée. En comparant les résultats obtenus avec différentes architectures, le concepteur dispose d'informations lui permettant de dimensionner correctement les ressources de calcul nécessaires à la mise en œuvre du système à concevoir. Les modèles proposés dans cette approche permettent aussi le dimensionnement des ressources de mémorisation. Des travaux sont actuellement menés afin d'effectuer des mesures sur une plate-forme afin d'évaluer la précision des résultats obtenus en s'appuyant sur la démarche d'exploration proposée.

5 Références

- [1] K. van Berkel, "Multi-Core for Mobile Phones," DATE'09, Proceedings of the Conference on Design, Automation and Test in Europe, pp. 1260-1265, 2009.
- [2] A. Viehl, B. Sander, O. Bringmann, and W. Rosenstiel, "Integrated requirement evaluation of non-functional system-on-chip properties", in *Proceedings of the Forum of specification and Design Languages (FDL'08)*, Stuttgart, Germany, September 2008.
- [3] D. Densmore, R. Passerone, and A. Sangiovanni-Vincentelli, "A platform-based taxonomy for esl design," *IEEE Design and Test of Computers*, vol. 23, no. 5, p. 359-374, 2006.
- [4] F. Ghenassia, *Transaction-Level Modeling with SystemC TLM Concepts and Applications for Embedded Systems*. Springer, 2005.
- [5] L. Cai et D. Gajski. *Transaction Level Modeling: An Overview*. ISSS-CODES'03
- [6] D. Gajski and R. Kuhn, "Guest Editors' Introduction New VLSI Tools," in *IEEE Computer*, 1983, pp. 11-14.
- [7] A. Viehl, B. Sander, O. Bringmann, and W. Rosenstiel, "Integrated requirement evaluation of non-functional system-on-chip properties", in *Proceedings of the Forum of specification and Design Languages (FDL'08)*, Stuttgart, Germany, September 2008.
- [8] T. Arpinen, E. Salminen, T. Hämäläinen, and M. Hännikäinen, "Performance evaluation of UML2-modeled embedded streaming applications with system-level simulation", *EURASIP Journal on Embedded Systems*, vol. 2009, 2009.
- [9] J. Kreku, M. Hoppari, T. Kestilä, et al., "Combining UML2 application and SystemC platform modelling for performance evaluation of real-time embedded systems", *EURASIP Journal on Embedded Systems*, vol. 2008, 2008.
- [10] S. Le Nours, A. Barreteau, and O. Pasquier, "Modeling technique for simulation time speed-up of performance computation in transaction level models," *Forum on Specification & Design Languages*. FDL'10. Southampton, 2010.
- [11] 3GPP. (2009, Mar.) 3GPP TS 36.201 V8.3.0, Evolved Universal Terrestrial Radio Access (E-UTRA); LTE physical layer; General description (release 8). [Online]. "http://www.3gpp.org/ftp/specs/html-info/36201.htm"