

SYSTÈMES INTÉGRÉS : UN MICRO-RÉSEAU D'INTERCONNEXION À COMMUTATION DE PAQUETS RESPECTANT LA NORME VCI

Hervé CHARLERY

Alain GREINER

Département ASIM, laboratoire LIP6, Université Pierre et Marie CURIE
4, place Jussieu, 75252 Paris Cedex 05
Herve.Charlery@lip6.fr
Alain.Greiner@lip6.fr

RÉSUMÉ : Le micro-réseau SPIN (pour Scalable Programmable Integrated Network) est un réseau d'interconnexion à commutation de paquets pour systèmes intégrés sur puce développé au LIP6. Cette technologie fournit un mécanisme de communication très général entre les différents composants virtuels connectés dans le système, puisque la bande passante croît linéairement avec le nombre de processeurs intégrés. On décrit plus particulièrement les fonctionnalités et l'architecture interne des *wrappers* VCI/SPIN qui permettent de fournir aux composants virtuels une interface de communication respectant le standard VCI (Virtual Component Interface).

I – INTRODUCTION

Le concept SPIN [1] [2] d'architecture de communication à haut débit pour systèmes intégrés sur puce dérive de l'expérience acquise dans le domaine des calculateurs parallèles. Ces machines ont de gros besoins en bande passante. Elles utilisent souvent des réseaux d'interconnexion multi-étages qui mettent en oeuvre des liaisons point à point et des routeurs, comme substitut au traditionnel "bus système". Dans les systèmes intégrés, le bus de communication est souvent le goulot d'étranglement, et cette tendance ne peut que s'accroître avec l'augmentation des capacités d'intégration. La technologie de micro-réseau à commutation de paquets SPIN développée au LIP6 est une réponse possible à ce problème. Cependant pour permettre une migration aisée d'une architecture à base de bus vers une architecture utilisant un réseau commuté, il faut permettre la réutilisation des composants existants (IP cores). Par conséquent, le réseau doit fournir aux concepteurs de systèmes la même interface et le même type de service que le bus traditionnel. Le standard VCI [3] (Virtual Component Interface), normalisé par le consortium VSIA, définit un protocole de communication de type "espace mémoire adressable partagé", qui peut être implémenté aussi bien par un bus système traditionnel de type AMBA ou PIBUS, que par un réseau d'interconnexion commuté. Ce papier présente comment la technologie de réseau commuté SPIN développée au LIP6 peut être utilisée pour fournir un mécanisme d'interconnexion générique respectant la norme VCI.

VCI présente de nombreux avantages. L'interface VCI introduit un découplage complet entre la conception ou le choix des composants du système (coeurs de microprocesseur, DSP ou coprocesseurs spécialisés), et le choix du dispositif matériel de communication (bus classique, bus hiérarchiques, réseau commuté, etc...). De plus, l'extrême simplicité du mécanisme de contrôle de flux défini par VCI facilite beaucoup la conception des interfaces d'accès au réseau. Enfin, VCI supporte les transactions éclatées.

Dans ce papier nous présentons successivement les caractéristiques essentielles du réseau SPIN, et les principes généraux de l'interface VCI. Nous décrivons ensuite l'architecture des "wrappers" qui réalisent la conversion entre les deux protocoles. Nous terminons par une description de la méthode générale de validation de cette architecture.

Dès lors, une connexion au réseau SPIN prend l'allure suivante :

- un composant doté d'une interface de type VCI,
- un wrapper VCI/SPIN de type initiateur ou cible afin de passer du monde VCI au monde SPIN ou inversement,

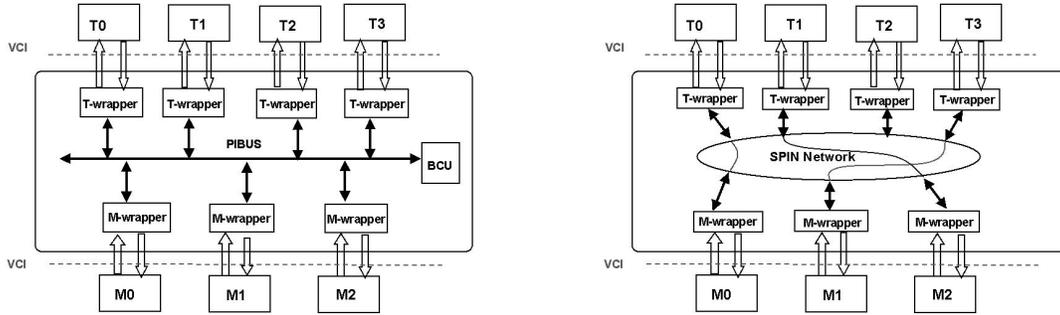


FIG. 1 – Exemples de systèmes interconnectant des composants VCI

II – LE RÉSEAU SPIN

II-1 – Structure d'un paquet SPIN

Les informations qui circulent sur le réseau SPIN sont des paquets. Un paquet SPIN est une séquence de mots de 36 bits, dont le premier mot possède un marqueur de début de paquet (BP) et le dernier mot un marqueur de fin de paquet (EP). La largeur d'un mot est de 36 bits qui se répartissent en 32 bits de données, 3 bits d'étiquette qui permettent de typer les mots, et un bit de parité. Les marqueurs EP et BP sont transportés dans le champ étiquette.

II-2 – Routage d'un paquet SPIN

Le premier mot d'un paquet SPIN constitue l'en-tête du paquet et contient nécessairement le numéro du port destinataire. Ce numéro, codé sur 10 bits est utilisé par le réseau pour acheminer le paquet vers le port destinataire. Chaque port est donc identifié par un numéro unique qui est imposé par la topologie du réseau, et ne peut donc être modifié.

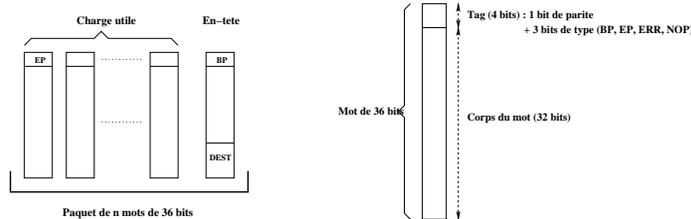


FIG. 2 – Structure d'un paquet et d'un mot de type SPIN

III – L'INTERFACE VCI

La norme VCI fait l'hypothèse que tous les composants du système partagent un même espace d'adressage. Les maîtres (ou initiateurs) envoient des requêtes de lecture ou d'écriture vers des cibles, qui sont identifiées par les bits de poids fort de l'adresse, comme sur un bus. Il existe donc deux types de paquets VCI : Les maîtres envoient des paquets *requête* constitués de une ou plusieurs adresses (en cas de rafale). Les cibles renvoient des paquets *réponse*. Les paquets sont de longueur quelconque, et sont terminés par un marqueur de fin de paquet. Le paquet *réponse* possède la même longueur que le paquet *requête* auquel il est associé et toute requête reçoit un acquittement. Chaque mot d'un paquet *requête* VCI comporte une centaine de signaux parmi lesquels on trouve des signaux de contrôle de flux, de donnée (en cas d'écriture), d'adresse et de définition de la requête. Chaque mot d'un paquet *réponse*, comporte une quarantaine de signaux qui sont les signaux de contrôle de flux, de donnée, et d'acquiescement pour la gestion des erreurs.

Le mécanisme d'accès au réseau d'interconnexion est extrêmement simple, puisque chaque composant a l'impression d'écrire dans une FIFO. Plusieurs initiateurs peuvent donc émettre simultanément des requêtes, et un même maître peut émettre une requête (n+1) sans attendre d'avoir reçu la réponse à la requête (n). Cette possibilité s'appuie sur un mécanisme d'étiquetage des paires (requête/réponse) et est utilisée par des processeurs *multi-thread* pour masquer la latence des accès mémoire.

IV – LES WRAPPERS VCI / SPIN

IV-1 – Principes généraux

Nous appelons " wrapper " un composant matériel qui réalise une conversion entre deux protocoles de communication.

Puisqu'il existe deux types de paquets VCI (paquets " requête " et paquets " réponse "), il faut définir un mécanisme de traduction spécifique pour chacun de ces deux types. De plus une requête de lecture ne contient pas les mêmes informations qu'une requête d'écriture (puisque'il n'y a pas de données dans une requête de lecture). Il en va de même pour les réponses. Il faut donc distinguer en pratique 4 types de paquets VCI :

- cas a) un paquet " requête de lecture " VCI de longueur N (rafale de N lectures) engendrera un paquet SPIN de longueur N+1 : un mot par adresse, plus un mot d'en-tête.
- cas b) un paquet " requête d'écriture " VCI de longueur N (rafale de N écritures) engendrera un paquet SPIN de longueur 2N + 1 : deux mots par adresse ,plus un mot d'en-tête.
- cas c) un paquet " réponse lecture " VCI de longueur N engendrera un paquet SPIN de longueur N+1 : un mot par adresse, plus un mot d'en-tête.
- cas d) un paquet " réponse écriture " VCI de longueur N engendrera un paquet SPIN de longueur N+1 : un mot par adresse, plus un mot d'en-tête.

Puisqu'il existe deux types de composants VCI (des initiateurs et des cibles), il y a donc deux types de wrappers. Le wrapper de l'initiateur est responsable des conversions VCI -> SPIN dans les cas a) et b), ainsi que des conversions SPIN -> VCI dans les cas c) et d). Le wrapper de la cible est responsable des conversions VCI -> SPIN dans les cas c) et d), ainsi que des conversions SPIN -> VCI dans les cas a) et b).

Les requêtes et les réponses étant fondamentalement asynchrones, dans chacun des deux wrappers (initiateur et cible), il y aura deux automates indépendants pour traiter les paquets "requête" dans un sens, et les paquets "réponse" dans l'autre.

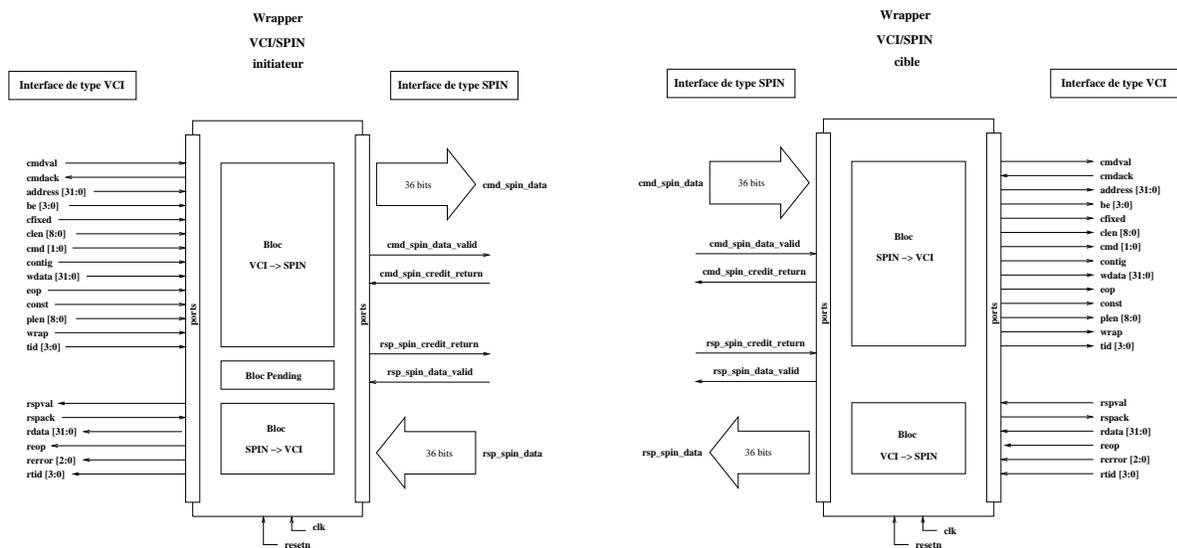


FIG. 3 – Les wrappers VCI/SPIN initiateur et cible

IV-2 – Le wrapper initiateur

Le wrapper initiateur en plus des conversions de format décrites à la section précédente, a deux autres fonctionnalités importantes :

C'est lui qui est chargé de décoder les bits de poids fort de l'adresse et de déterminer le numéro de port destinataire à placer dans l'en-tête du paquet SPIN. Il effectue ce " transcodage " en fonction des " tables de routage " qui définissent la structure de la carte mémoire (la carte mémoire attribuée à chaque composant cible

un segment particulier de l'espace adressable. Ces tables de routage sont évidemment identiques pour tous les wrappers, puisqu'elles constituent une caractéristique globale du système, mais elles doivent être " câblées " dans chaque wrapper initiateur. Cette fonction de sélection de la cible, traditionnellement assurée par le contrôleur de bus, est ici répliquée dans tous les wrappers. C'est un des prix à payer pour avoir une architecture de communication " extensible ". Accessoirement, le wrapper initiateur doit indiquer dans l'entête du paquet SPIN son propre numéro de port, pour permettre au wrapper cible de connaître le numéro du port auquel il doit envoyer le paquet réponse.

Par ailleurs, le standard VCI " advanced " permet à un même maître d'envoyer une seconde requête sans attendre d'avoir reçu la réponse à la première. Il utilise pour cela le champs TID du mot VCI, qui permet de définir un " numéro de requête. Evidemment, le wrapper ne doit pas transmettre deux requêtes avec le même numéro de transaction. Il gère donc une table des requêtes pendantes, identifiées par leur TID.

Le wrapper détecte les erreurs de routage, qui surviennent dans le cas où les bits de poids fort de l'adresse désignent une cible inexistante. Cette erreur déclenche l'activation d'une interruption à destination de l'initiateur.

IV-3 – Le wrapper cible

Le wrapper cible est plus simple, puisqu'il séquentialise les requêtes : Il attend d'avoir reçu le dernier mot de la réponse à la requête i avant de commencer à transmettre le premier mot de la requête $i+1$. On peut noter cependant qu'il faut absolument deux automates indépendant pour traiter les requêtes et les réponses si on veut éviter les situations de " dead-lock ".

V – VALIDATION

Pour la simulation, on utilise un outil développé au laboratoire LIP6 : le simulateur CASS [4]. Ce simulateur prend en entrée des modèles écrits en C, de niveau RTL, permettant une simulation *cycle-true / bit-true*. L'architecture des wrappers a été validée par la simulation d'une application parallèle multi-threads réalisant un calcul de Transformation de Fourier Rapide (FFT), sous le contrôle d'un petit système d'exploitation multiprocesseurs et multi-threads [5] développé au LIP6. Cette application a été simulée sur une architecture matérielle composée de huit routeurs RSPIN, de 16 wrappers initiateurs et cibles (respectivement WI et WT), de 4 processeurs de type MIPS R3000 doté de caches à interfaces VCI, de RAMs et d'une fenêtre d'affichage. La durée de la simulation pour ce système comportant 44 composants, est d'environ 80 secondes sur une machine Solaris à 450MHz pour 200 000 cycles simulés.

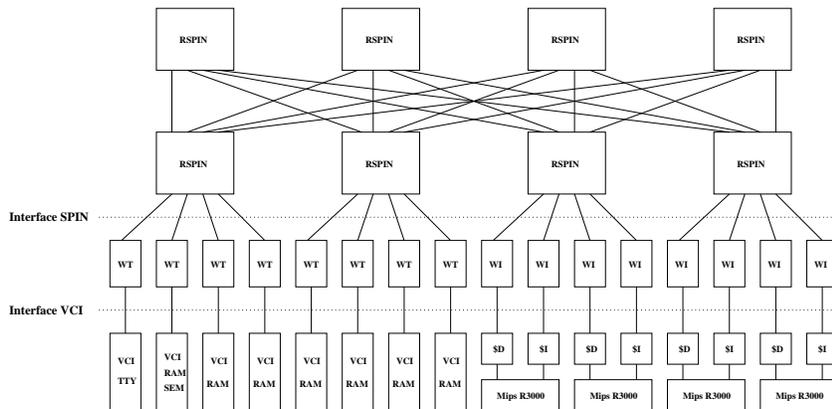


FIG. 4 – Système de test multiprocesseurs

RÉFÉRENCES

- [1] *Un réseau d'interconnexion pour systèmes intégrés*, P. Guerrier, Thèse, 10 mai 2000.
- [2] *S.P.I.N.*, A. Adriahtenaina, A. Greiner, Rapport technique, Université Pierre et Marie Curie, 2001.
- [3] *System VC Interface Strawman Version 0.3.2*, Strawman.
- [4] *Efficient Combinational Loops Handling for Cycle Precise Simulation of System on a Chip*, Denis Hommais and Frédéric Pétrot, "24th Euromicro", pages 51–54, IEEE, Vasteras, Sweden, 1998.
- [5] *MUTEK*, Frédéric Pétrot, Rapport technique, Université Pierre et Marie Curie, 2001.
- [6] *Les wrappers VCI/SPIN*, H. Charlery, A. Greiner, Rapport technique, Université Pierre et Marie Curie, 2001.