

# Architecture manycore à mémoire partagée distribuée et routeurs programmables PNoC

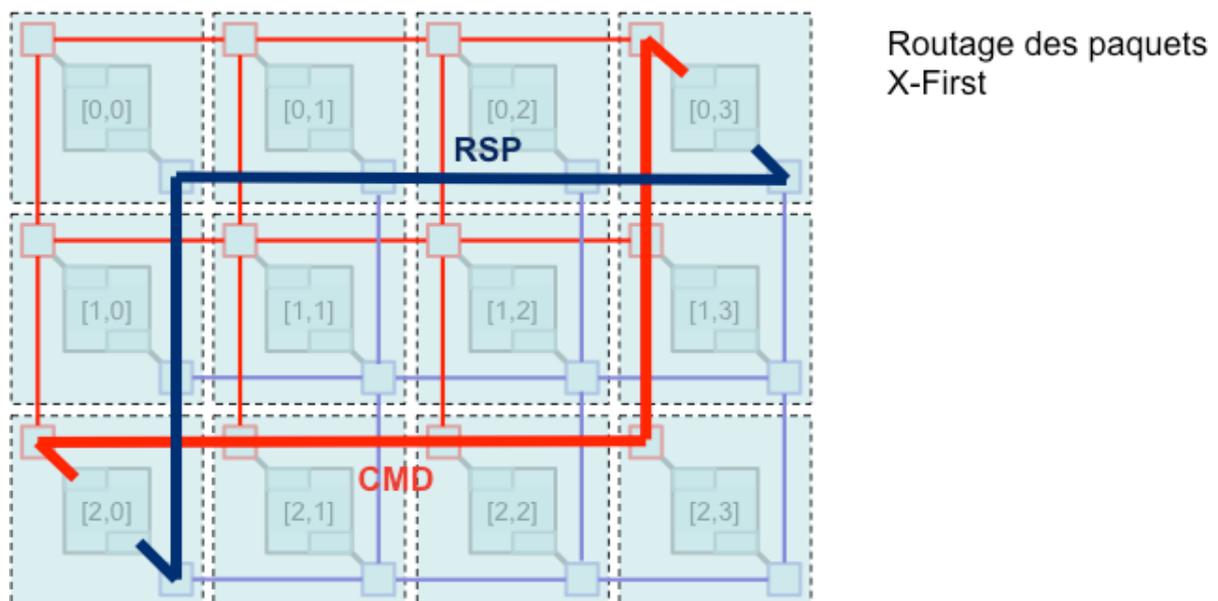
Dimitru Potop Butucaru, Manel Djemal INRIA  
Franck Wajsbürt, François Pêcheux, Zhen Zang LIP6

25/01/12

Ce document décrit l'architecture matérielle actuelle de la plateforme PNoC. Le document contient 3 sections : SoCLib, Architecture de la plateforme, Routeur programmable.

## 1) SoCLib

La plateforme PNoC proposée repose sur SoCLib, une bibliothèque de modèles SystemC interopérables pour la modélisation et la simulation de systèmes massivement parallèles manycore. Cette bibliothèque contient des modèles optimisés en SystemC à trois niveaux d'abstraction : RTL synthétisable (VHDL pour FPGA), Cycle Accurate Bit Accurate (CABA) et transactionnel (Transaction Level Modeling, TLM). Les composants majeurs de ces systèmes manycore sont présents dans la bibliothèque SoCLib (processeurs single-issue, caches, mémoires RAM, ROM, gestionnaire d'interruptions, périphériques, DMA, terminaux d'affichage) et sont interconnectés au moyen de modèles de réseaux d'interconnexion. Les réseaux d'interconnexion de SoCLib sont de deux types. Il y a d'un côté le VGMN (Virtual Generic Macro Network) qui permet de connecter virtuellement N initiateurs et M cibles sans se soucier de topologie (crossbar parfait), et l'autre DSPINPlus, qui est une implémentation physique réelle de réseau d'interconnexion, avec une topologie de mesh 2D et par défaut un algorithme de routage des paquets de type X-First. La figure suivante illustre le protocole de routage X-First par défaut.



L'architecture DSPINPlus décrit exactement le réseau global, mais est beaucoup plus long à simuler.

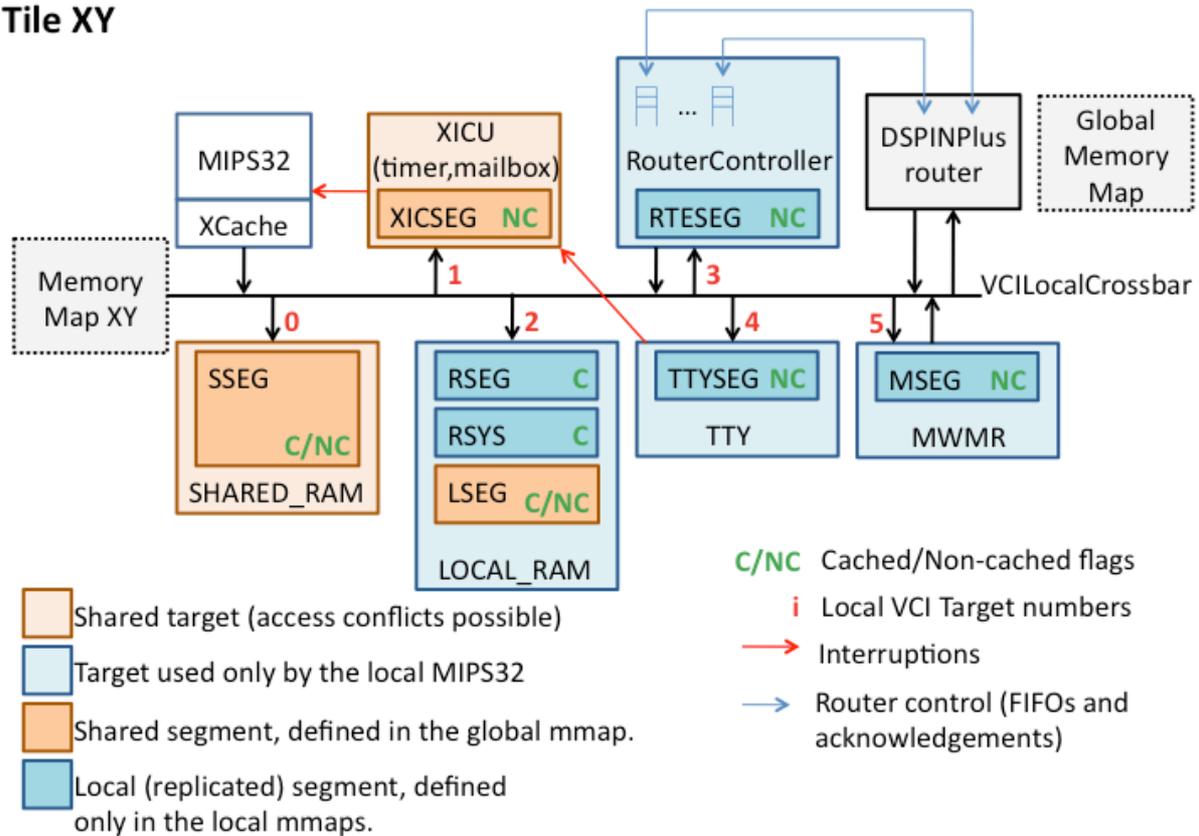
## 2) L'architecture manycore à mémoire partagée distribuée PNoC

En s'appuyant sur cette bibliothèque, l'INRIA et le LIP6 ont conjointement développé un générateur d'architectures manycore tuilées, le générateur PNoC. Le générateur prend en entrée les dimensions du mesh 2D, le type de tuiles (A, B ou C actuellement) et génère une topcell C++ réalisant l'instanciation des tuiles et des routeurs du réseau d'interconnexion global ainsi que toute la connectique. L'application logicielle qui s'exécute sur cette architecture est principalement écrite en C (une centaine de lignes en assembleur), et compilable avec les outils de compilation croisée GNU. Grâce aux « attribute » du C compris par GCC et permettant de définir des sections paramétrables, il est possible de placer de manière très précise le code et les données sur les bancs mémoire matériels de PNoC (et donc de définir quel code s'exécute sur quel processeur). Les modèles de mémoire SoCLib sont capables d'initialiser le contenu des segments mémoires à partir de fichiers compatibles ELF.

Comme le montre la figure suivante, une tuile PNoC contient pour l'instant les ressources suivantes :

- un processeur single issue (MIPS32, PPC405, ARM7TDMI, SPARC V8) et son cache d'instructions et de données associé (XCache)
- un composant concentrateur d'interruptions, Timer, XICU autorisant les IPI (Inter Processor Interrupts)
- une RAM partagée, accessible de n'importe quelle tuile
- une RAM locale, accessible localement
- un contrôleur MWMR permettant le cas échéant de piloter un coprocesseur matériel par l'intermédiaire de FIFOs d'entrée et de sortie, de registres de commande et d'état
- un terminal pour afficher des résultats locaux
- une interface DSPINPlus router avec le réseau d'interconnexion global
- un contrôleur de router permettant de rendre le DSPINPlus router programmable

## Tile XY



Les adresses des composants de cette architecture à mémoire partagée sont les suivantes (X et Y représentent les coordonnées de la tuile considérée) :

SSEG : 0xXY400000 - shared address space, public  
 XICSEG : 0xXY900000 - shared address space, public  
 LSEG : 0xXY800000 - shared address space, private  
 RSEG : 0xBFC00000 - local address space  
 RSYS : 0x80000000 - local address space  
 TTYSEG : 0x10100000 - local address space  
 RTESEG : 0xB0500000 - local address space  
 MSEG : 0xC0D00000 - local address space

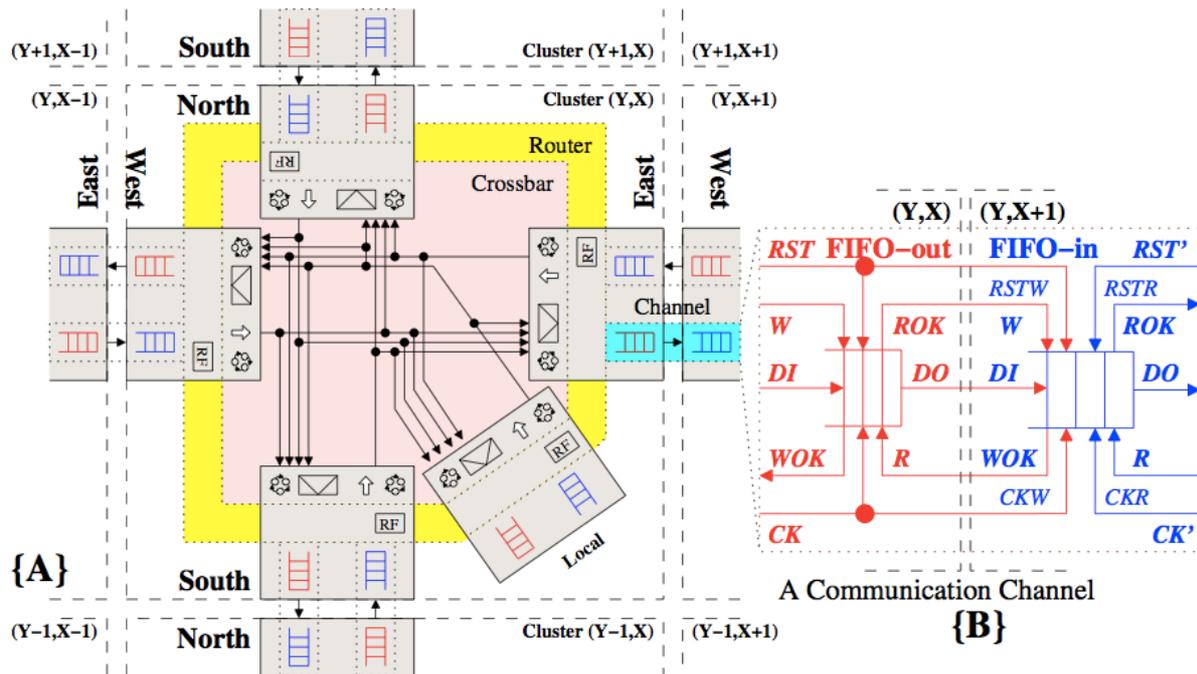
où

X = 4 bits donnant la position sur X

Y = 4 bits donnant la position sur Y

### 3) Routeur programmable DSPINPlus et son contrôleur RouterController

Ce sont les vraies nouveautés de l'architecture. Le routeur programmable DSPINPlus est un composant dérivé du routeur DSPINPlus original, représenté par la figure ci-dessous.



Le contrôleur de routeur contrôle/programme le routeur DSPINPlus modifié pour le mettre dans un des 2 modes suivants:

- BestEffort – Le mode classique DSPIN, avec routage RoundRobin
- Programmed – Tous les choix de routage sont pre-etablis.

Le contrôle du routeur se fait au niveau des 5 multiplexeurs associés aux 5 sorties (N/S/E/W/L), et au niveau de l'admission des paquets par le multiplexeur. Pour laisser passer un paquet, le multiplexeur doit recevoir une de 2 commandes:

- BestEffort – laisse passer le paquet arrivant en premier sur n'importe quelle entrée du multiplexeur (« en premier » = arbitrage)
- Static(i) – laisser passer le premier paquet arrivant sur l'entrée i du multiplexeur déterminé par programme propre. Les paquets éventuellement disponibles sur les autres entrées sont bloqués.

Une fois le paquet accepté (1er flit passé), le multiplexeur signale et attend une nouvelle commande. En l'absence de commande, le trafic est bloqué. Le RouterController est le composant responsable des commandes sur les 5 multiplexeurs, pour réaliser les politiques de routage statiques ou dynamiques définies par l'utilisateur.

