

QUESTIONS sur le TP5 / Partage du bus dans les architectures multicœurs

Q1) Le protocole PIBUS est-il représentatif des protocoles des bus utilisés dans les machines commerciales, et suffit-il à une bonne compréhension du fonctionnement de ces bus?

Le protocole PIBUS est suffisamment simple pour qu'on puisse construire facilement les machines à états qui réalisent ce protocole. Mais il possède toutes les caractéristiques des bus utilisés dans les machines multi-cœurs à espace d'adressage partagés commerciales, telles que le comportement synchrone, la négociation entre les maîtres pour l'accès au bus, les transactions en deux temps (commande/réponse), la possibilité de faire des rafales, et le mécanisme de SNOOP pour la cohérence des caches. Et par dessus tout, il respecte le principe de base du bus qui est de ne supporter qu'une seule transaction à la fois, ce qui crée un problème de bande passante bornée.

Q2) Ne peut-on résoudre le problème de bande passante du bus en augmentant la largeur du bus (64 bits ou même 128 bits au lieu de 32 bits pour le champs DATA) ?

Cette technique est évidemment utilisée dans les bus modernes, mais elle permet seulement de gagner un facteur 2 ou 4, et de déplacer le seuil de saturation vers 8 ou 16 coeurs, mais elle ne résout pas le problème fondamental de ne permettre qu'une seule transaction à la fois.

Q3) Ne peut-on résoudre le problème de bande passante du bus en augmentant la fréquence de fonctionnement du bus ?

Non. La fréquence maximale de fonctionnement d'un bus est inversement proportionnelle à la capacité (électronique) des fils reliant les différents composants attachés au bus. Par conséquent cette fréquence maximale diminue lorsque le nombre de coeurs augmente, car la longueur des fils augmente, et la capacité des fils aussi. C'est la raison pour laquelle, dans les machines multi-cœurs actuelles, la fréquence du bus est généralement inférieure à la fréquence interne de chaque coeur. Cet comportement physique des bus va plutôt dans le sens de réduire la bande passante du bus, quand on voudrait au contraire l'augmenter ...

Q4) Comment le remplacement du bus par un micro-réseau peut-il résoudre le problème de bande passante, alors que la traversée des routeurs présents dans le micro-réseau va augmenter la durée d'une transaction commande/réponse ?

Il ne faut pas confondre *bande passante* et *latence*. La bande passante est l'inverse d'une durée (c'est un nombre d'octets par cycle). La latence est une durée (nombre de cycles pour effectuer une transaction commande/réponse). Le principe du micro-réseau est de permettre un grand nombre de transactions simultanées pour fournir une bande passante qui s'adapte au nombre de coeurs. Cela peut être obtenu en augmentant le nombre de routeurs du réseau proportionnellement au nombre de coeurs. Il est vrai que prix à payer pour cette bande passante illimitée est une augmentation de la latence puisque la commande doit traverser plusieurs routeurs à l'aller, et la réponse doit traverser plusieurs routeurs au retour. Cette augmentation de la latence est d'autant plus élevée que le nombre de coeurs et donc le nombre de routeurs à traverser est grand.

Q5) Les processeurs many-core industriels Intel ou AMD à 64 ou 128 coeurs utilisent-ils des bus ou des micro-réseaux ?

Ce qui différencie une architecture *multi-core* d'une architecture *many-core*, c'est précisément le type d'interconnexion. Une architecture devient *many-core* lorsque le nombre de coeurs est trop grand pour que tous les coeurs partagent le même bus.

Les architectures contenant 64 ou 128 coeurs sont généralement partitionnés en sous-systèmes (appelés clusters), contenant chacun quelques coeurs (2, 4 ou 8) partageant un même bus local et un même cache L2. Mais les différents clusters sont interconnectés entre eux par un micro-réseau, ce qui permet à n'importe quel coeur d'accéder à n'importe quel cache L2. L'espace d'adressage physique reste donc partagé par tous les coeurs de l'architecture, comme dans une machine multi-core telles que celles vues en TP.