"Communications dans les systèmes intégrés sur puce : du bus système au micro-réseau à commutation de paquets"

Alain Greiner

Outline

- Introduction
- Virtual Component Interface (VCI)
- The SPIN micro-network
- VCI over SPIN
- SPIN physical implementation
- Some issues in micro-networks
- Conclusion

System on Chip

Future consumer applications demand gigabit/s:

- **High Definition Video processing**
- **Network processing : ATM / Gigabit Ethernet**

Bus-based systems-on-a-chip cannot achieve reconfigurable multitask computing :

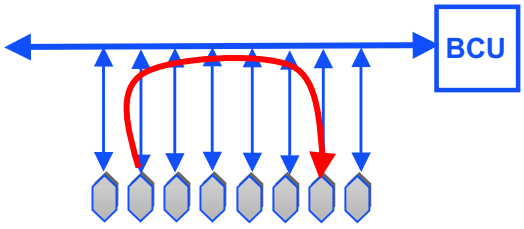
- **About 1 Gbit/ for each embedded core**
- **Over 50 cores in a typical soc**

  => Aggregated bandwidth over 50 Gbit/s

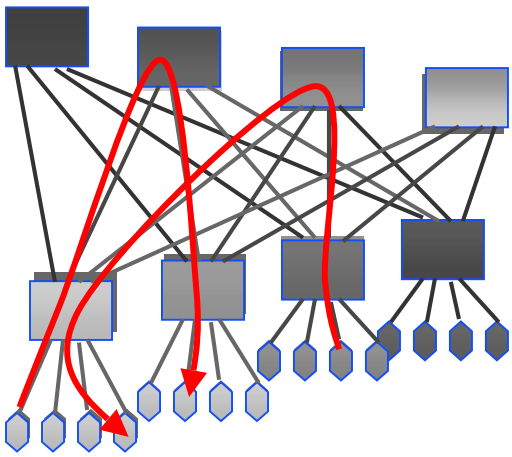
The bandwidth challenge

Shared bus

- Non scalable bandwidth
- Physical limitation on the frequency
- Central arbitration is a bottleneck



Multi-stage micro-network



- Several simultaneous connexions
- Point to point physical links
- Fully scalable

Outline

- Introduction
- Virtual Component Interface (VCI)
- The SPIN micro-network
- VCI over SPIN
- SPIN physical implementation
- Some issues in micro-networks
- Conclusion

VCI / Objectifs

- Séparer clairement - au niveau du matériel - la fonction de calcul de la fonction de communication.

  => réutilisation des composants matériels

- Supporter les architectures multi-processeurs

  => plusieurs dizaines de maîtres

- Conserver le paradigme de communication « espace d 'adressage partagé » : Un maître désigne sa cible par les bits de poids fort de l 'adresse et une case mémoire particulière par les bits de poids faible.

  => réutilisation des composants logiciels

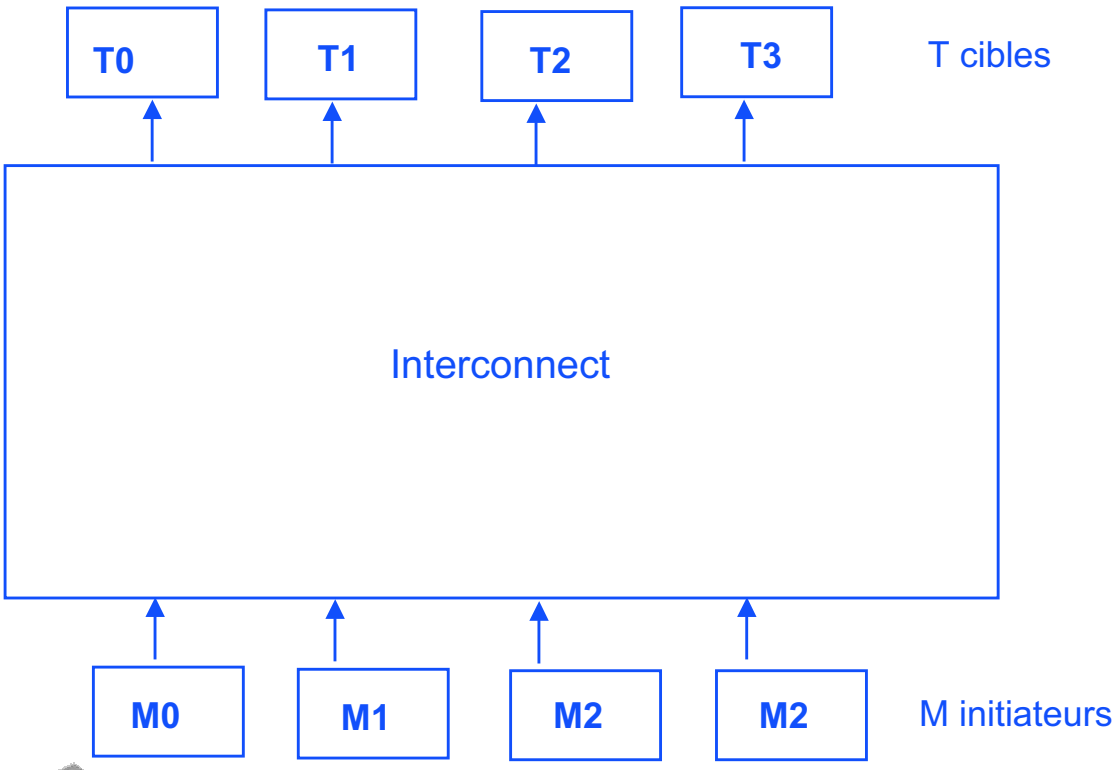
- Fournir à chaque maître l'illusion qu 'il dispose d 'un canal de communication point à point avec chaque cible.

  => simplification du protocole d 'accès au « bus »

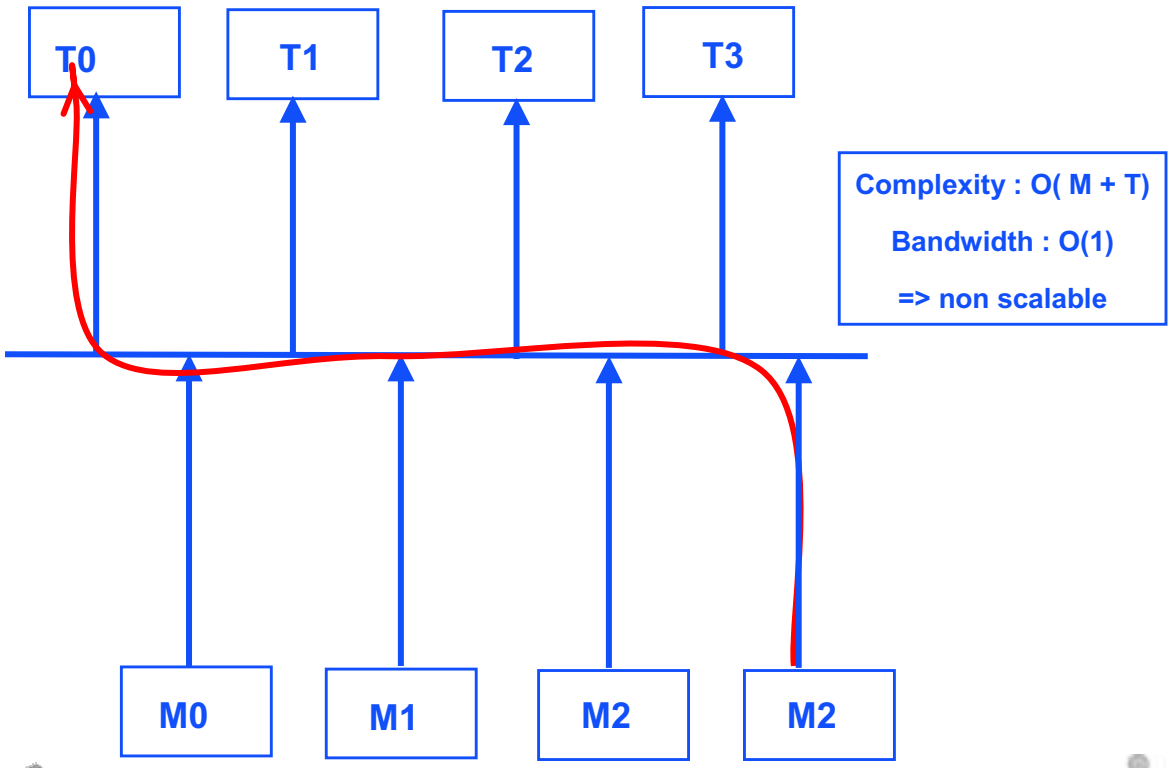
- Possibilité de mécanismes de communication autres que le bus.

  => bande passante « illimitée »

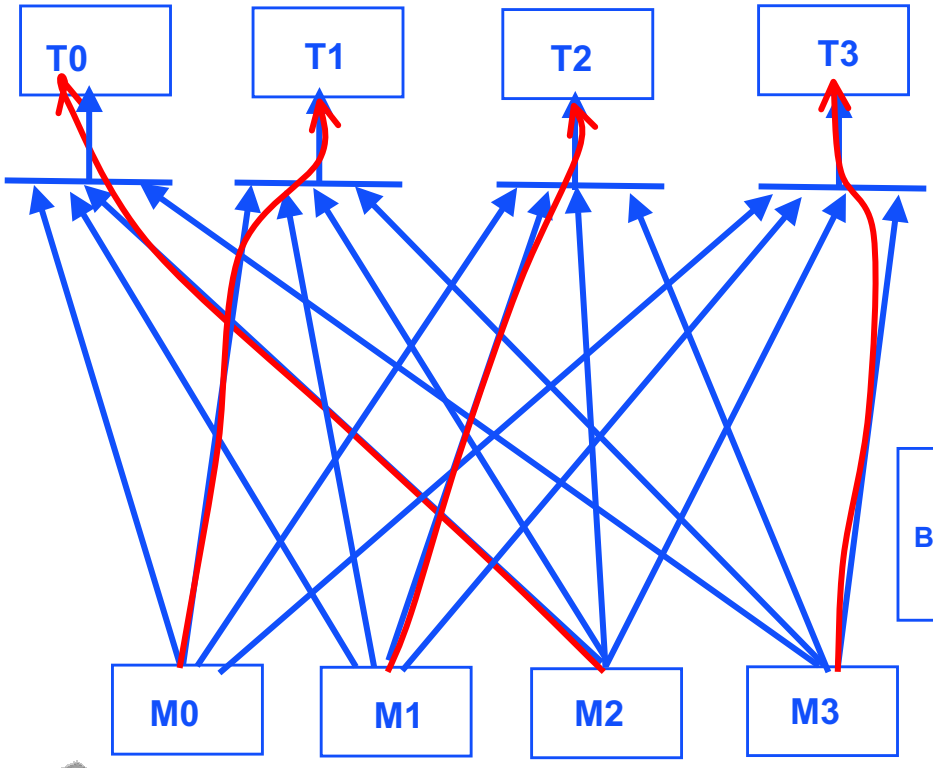
Communications ...



Communication : le bus



Communication : le cross-bar

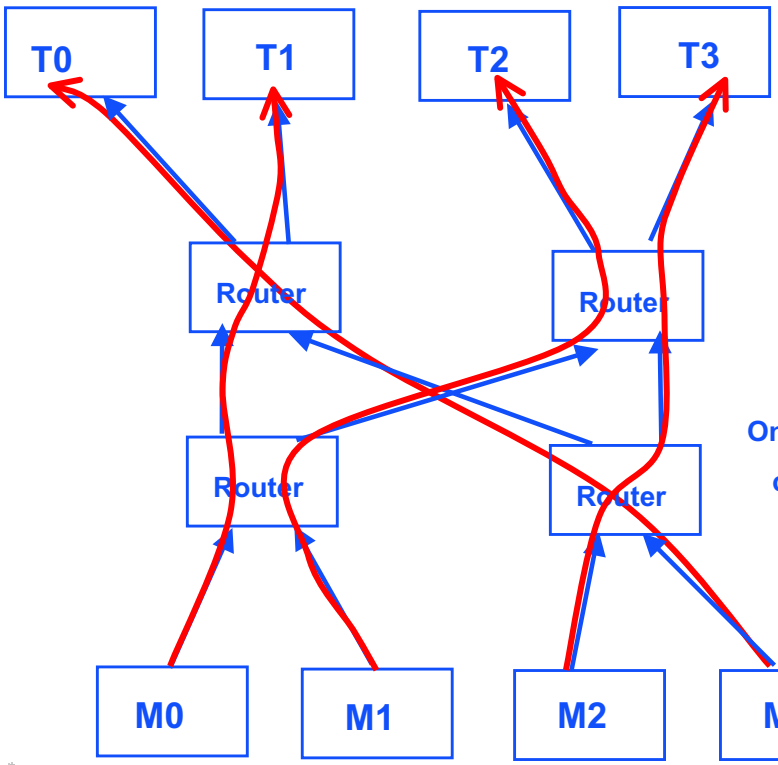


Complexity : O(M * T)

Bandwidth : O(max (M,T))

=> non scalable

Communication : le réseau multi-étages



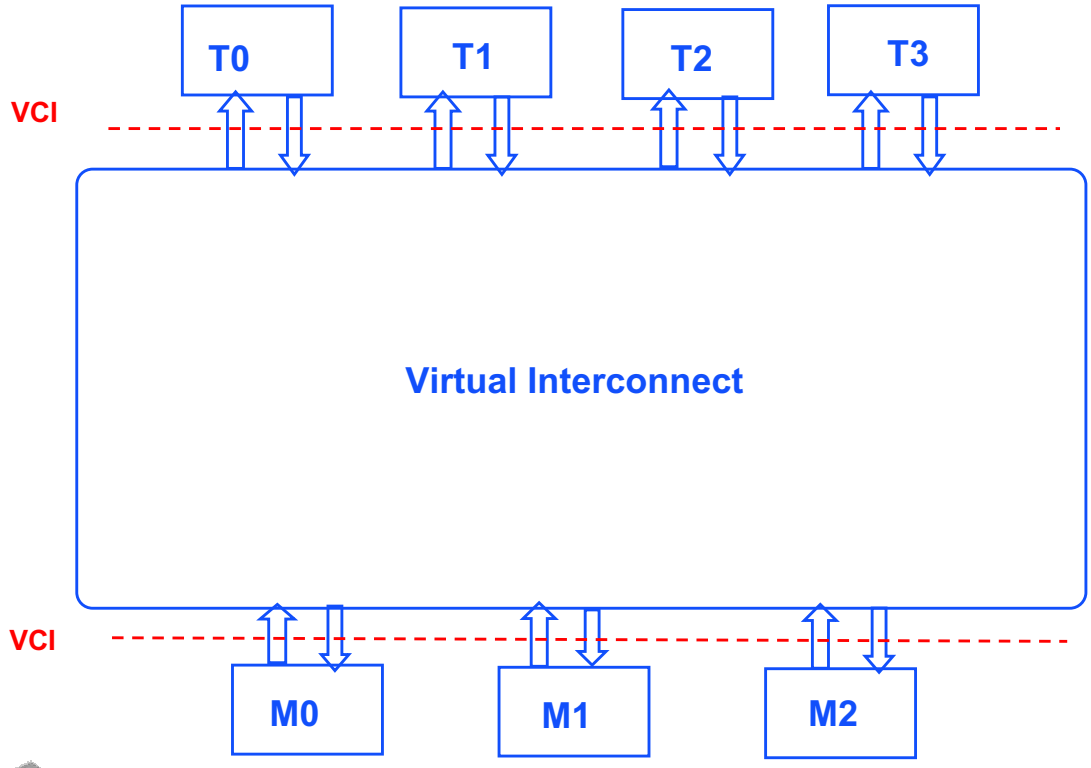
Complexity : O(log(M + T))

Bandwidth : O(max (M,T))

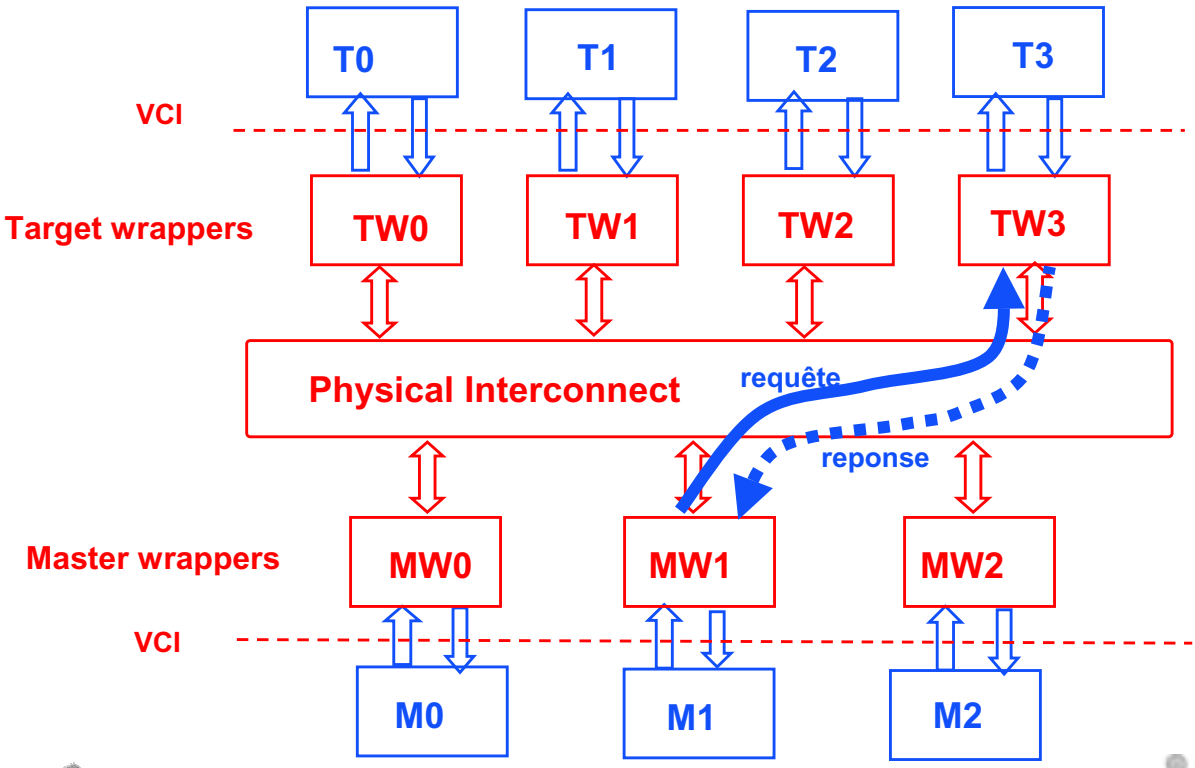
=> scalable

On cherche à émuler un cross-bar complet pour un coût moindre que celui du cross-bar

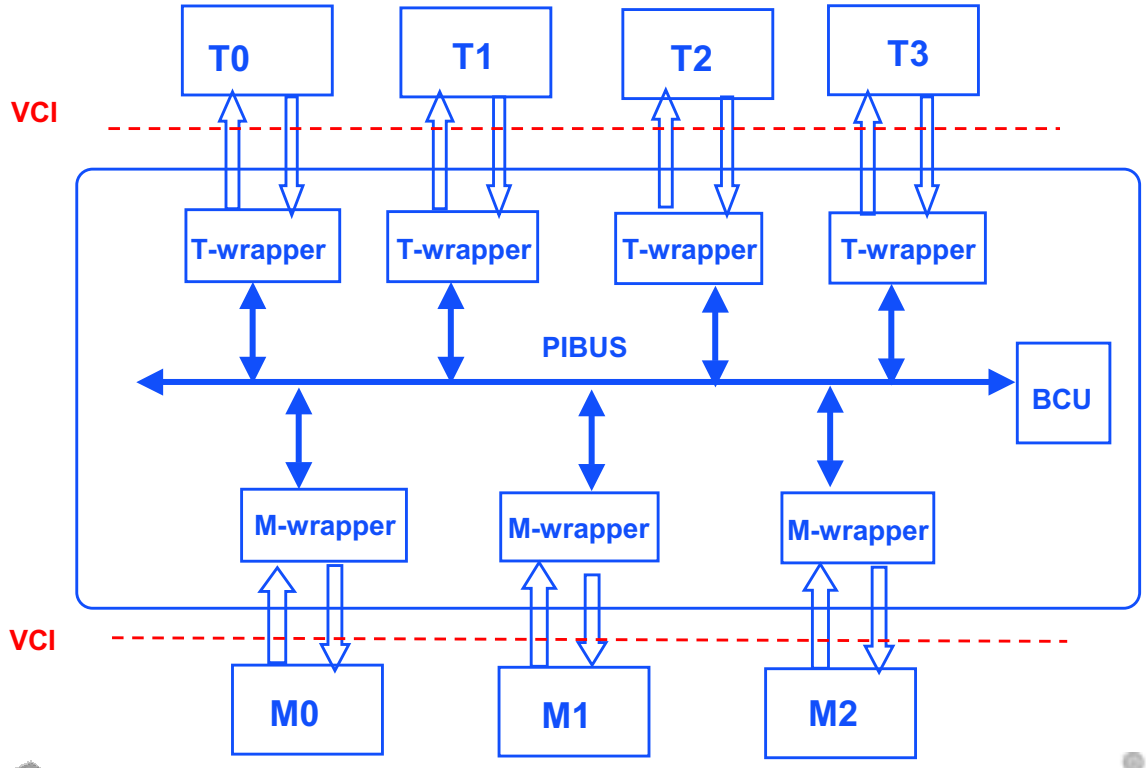
Virtual Component Interface



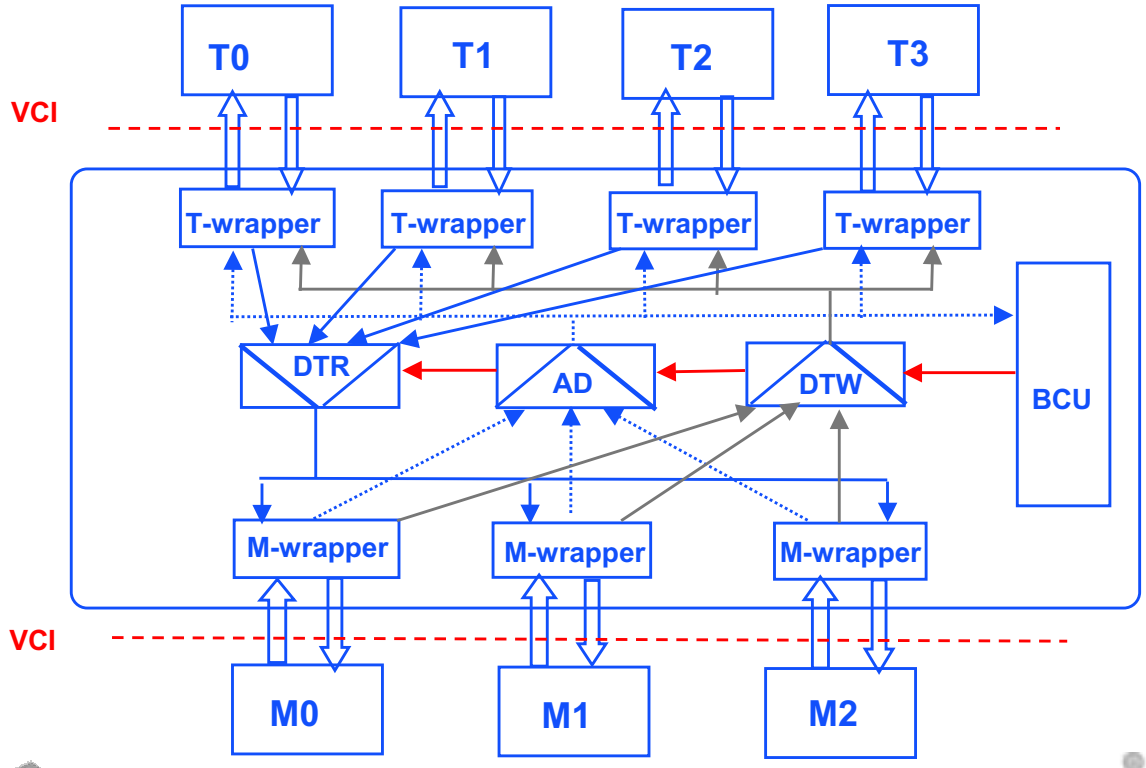
VCI wrappers



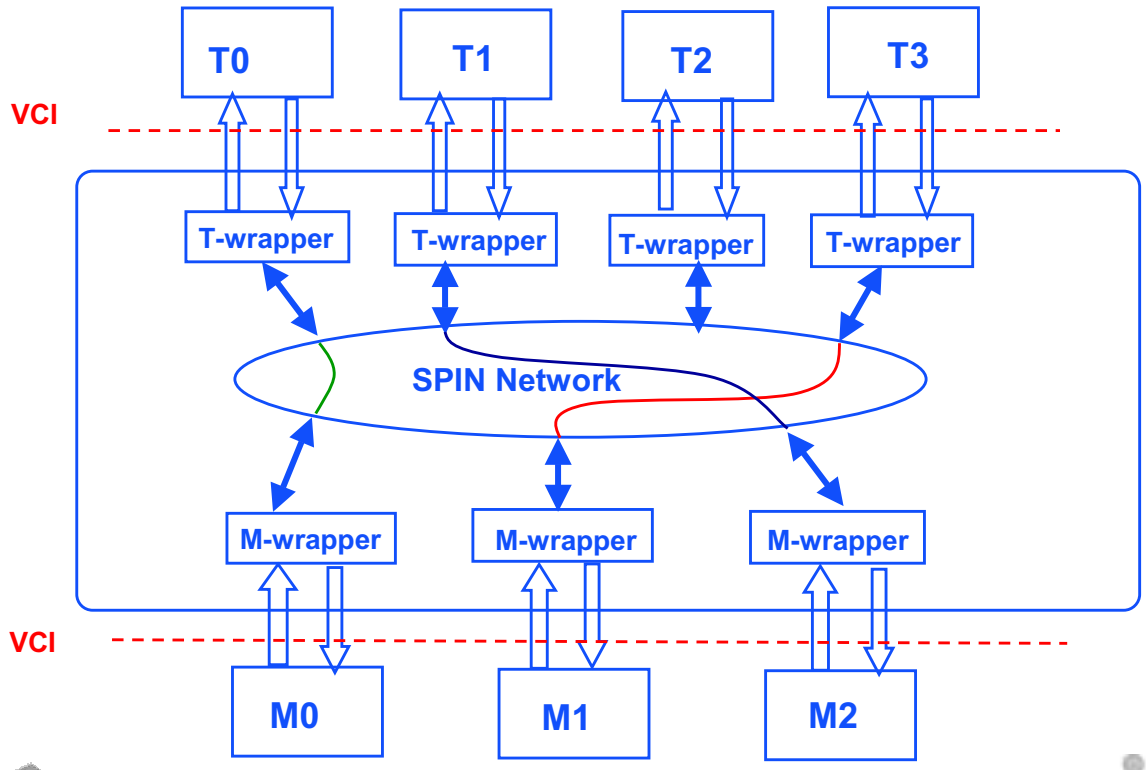
WRAPPERS VCI / PIBUS



WRAPPERS VCI / AMBA



WRAPPERS VCI / SPIN



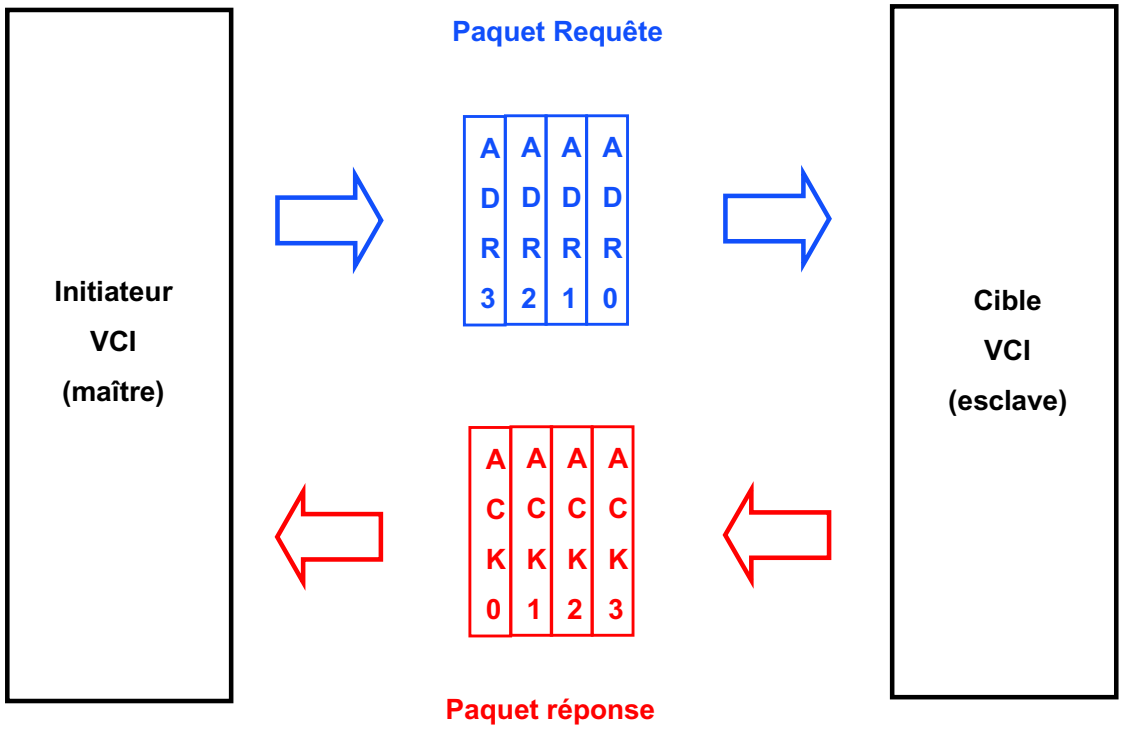
VCI / Principes

- VCI est construit autour d 'un espace adressable partagé.
- Les initiateurs émettent des requêtes de lecture ou d 'écriture. Une requête peut contenir une seule adresse (transaction simple) ou plusieurs adresses (transaction rafale).

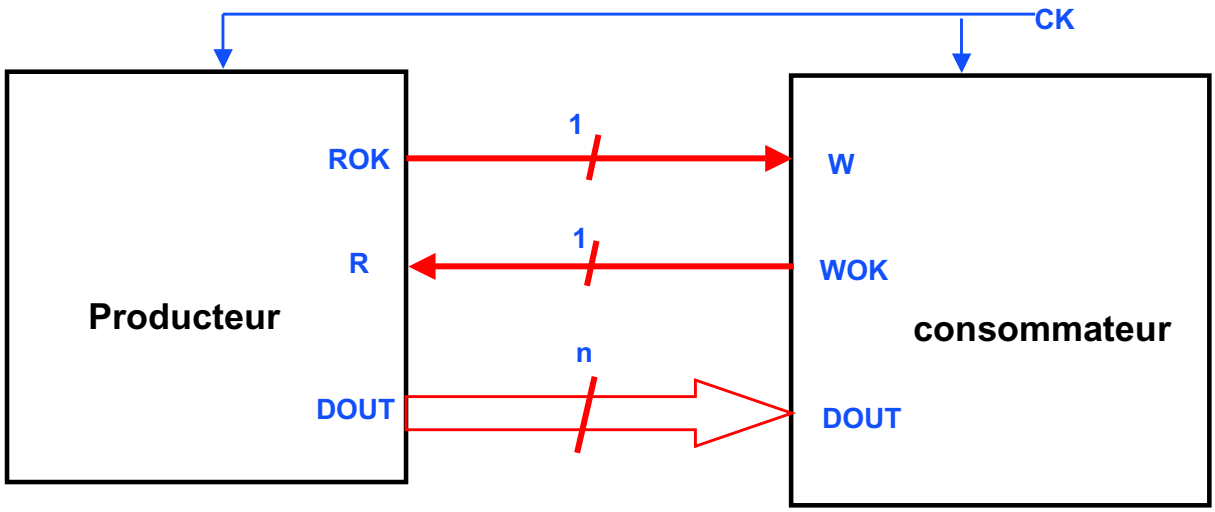
  => un paquet requête VCI contient autant de mots qu'il y a d 'adresses dans la rafale.
- Une cible est identifié par les poids forts de l 'adresse et renvoie un paquet réponse.

  => la longueur du paquet réponse est égale à celle du paquet requête.
- Une transaction est une paire paquet requête / paquet réponse.
- Tous les paquets sont transportés de façon atomique.
- Un même maître peut émettre une requête (n+1) sans attendre la réponse à la requête (n) …mais les réponses ne reviennent pas nécessairement dans l 'ordre où les requêtes ont été émises.

VCI / requêtes et réponses



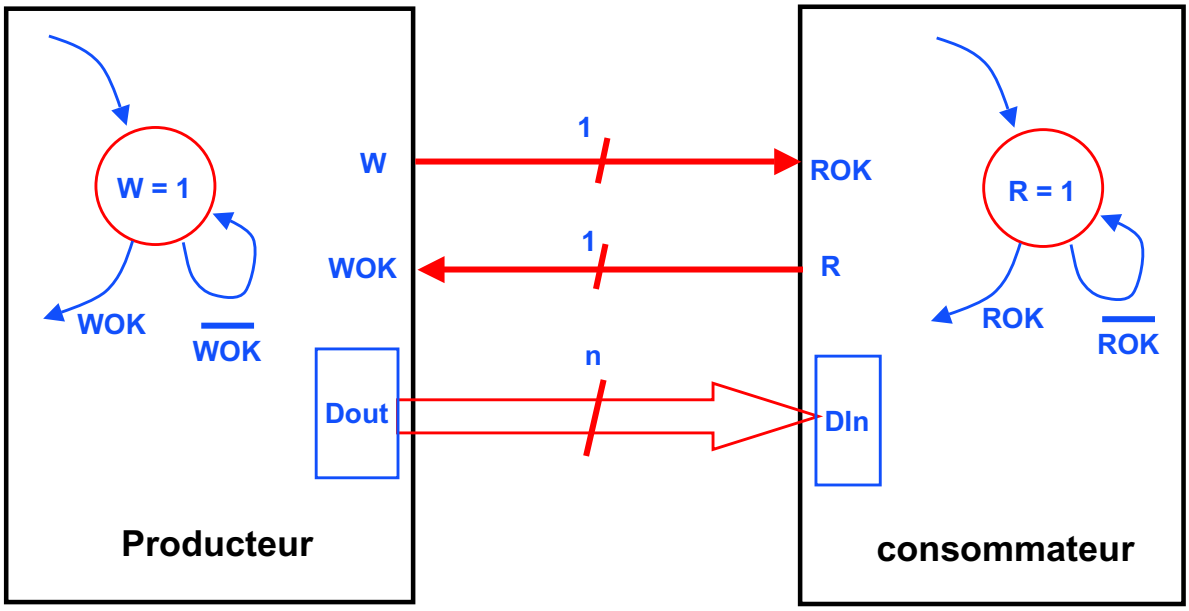
Contrôle de flux / accès au réseau



Protocole « FIFO » :

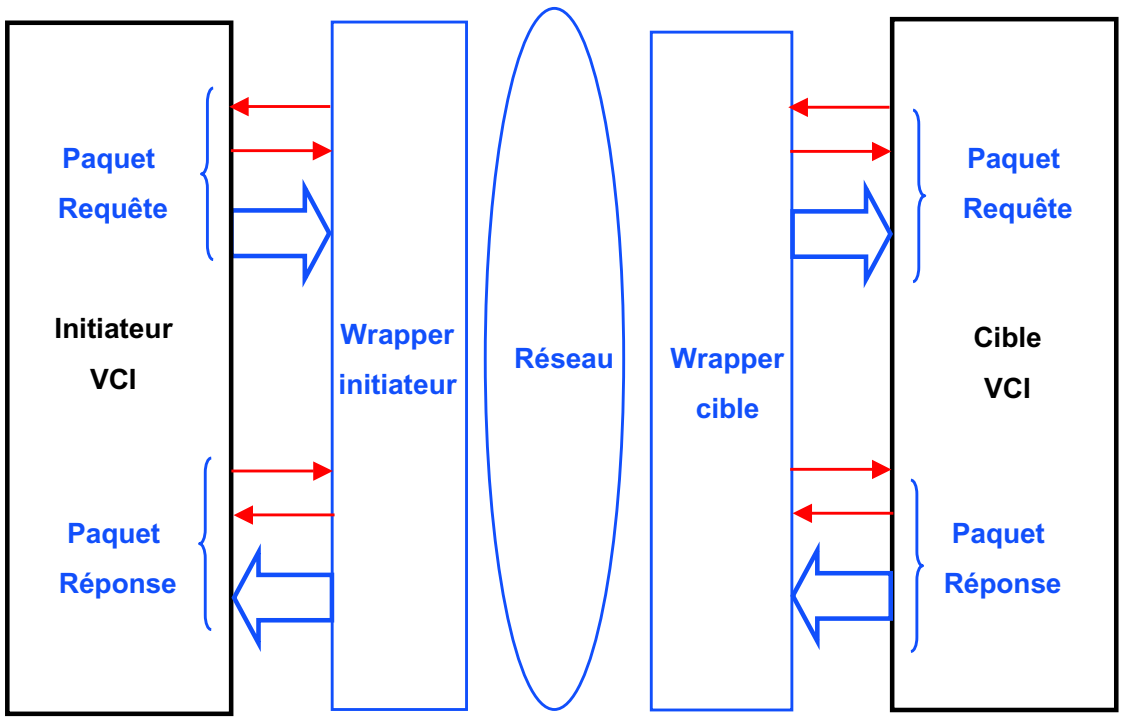
- communication asynchrone entre un producteur et un consommateur
- une donnée est transmise à chaque cycle où les deux signaux W/ROK et R/WOK sont tous les deux actifs.

Contrôle de flux / accès au réseau

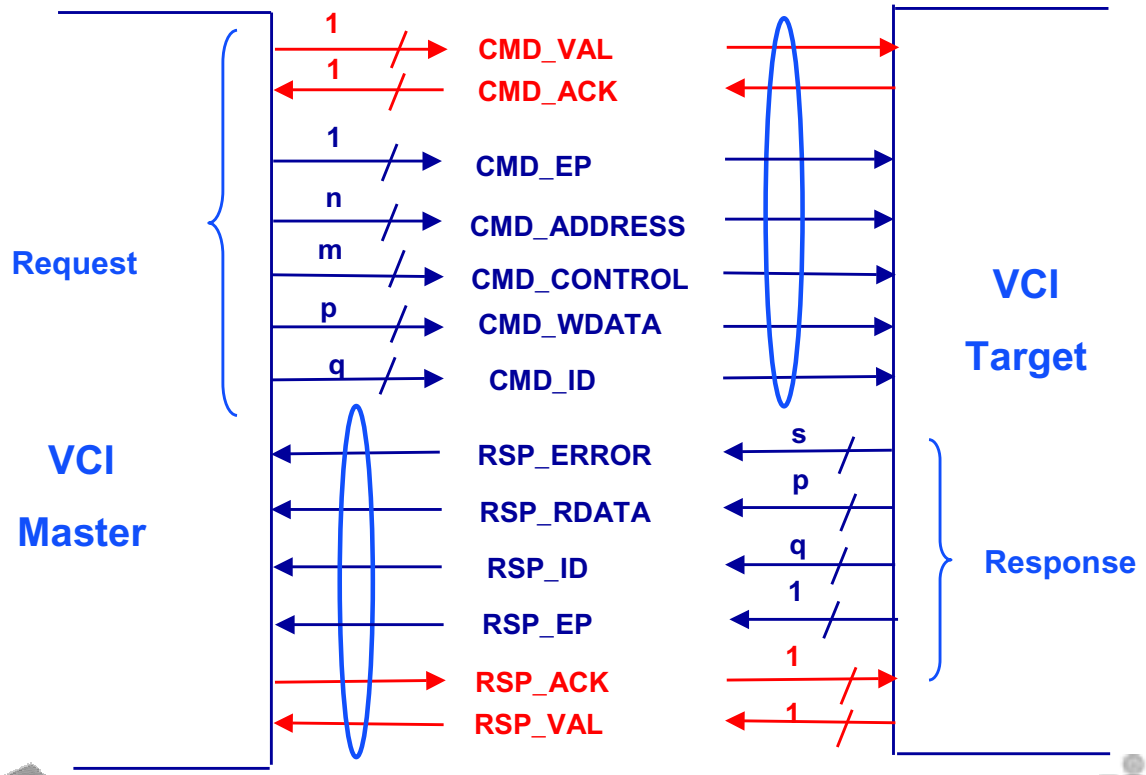


Le protocole « FIFO » peut être facilement implanté par des automates de MOORE, ce qui est essentiel pour des composants de communication !!!

VCI / requêtes et réponses



VCI / Physical Interface



Les commandes VCI

- Il y a quatre types de commandes
  - READ : lecture
  - WRITE : écriture
  - LINKED READ : lecture avec prise d 'exclusivité
  - NOP : no opération (?)
- Pour toutes les requêtes d 'une même rafale, les champs commande, ainsi que les bits de poids fort de l 'adresse doivent rester constants.
- Aucun signal ne permet d 'implanter un mécanisme d 'espionnage de bus (snoop) pour la cohérence des caches…
- Les signaux CMD_ID et RSP_ID permettant l 'émission de multiples requêtes par un même maître sont définis dans la norme « VCI advanced »

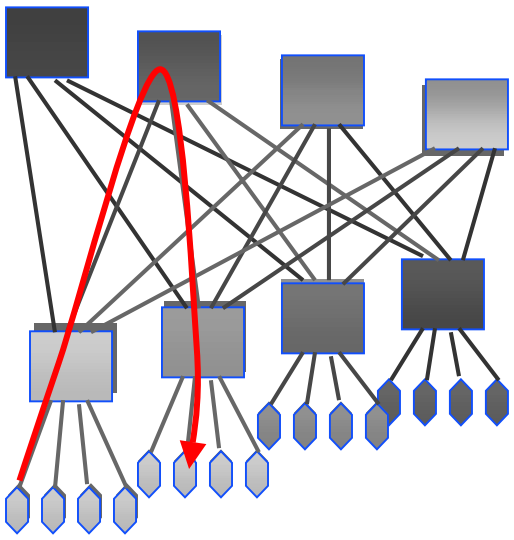
Outline

- Introduction
- Virtual Component Interconnect (VCI)
- The SPIN micro-network
- VCI over SPIN
- SPIN physical implementation
- Some issues in micro-networks

The SPIN Network

- packet switching network
- wormhole routing
- multi-level Fat-Tree topology
- point to point bidirectional links
- credit-based flow control
- adaptative routing

SPIN network



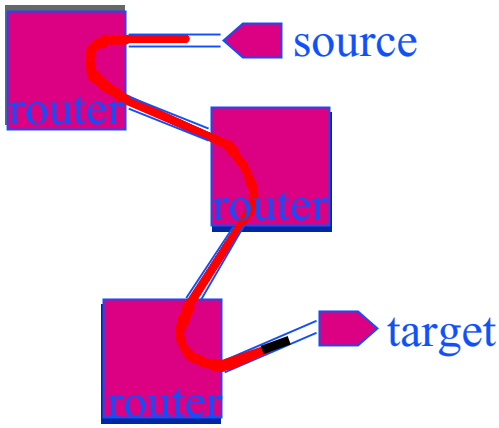
Multistage Packet Switched Networks

Packet Switching :

- **No circuit reservation**
- **Atomic transaction = *packet***
- **each packet contains the target ID**

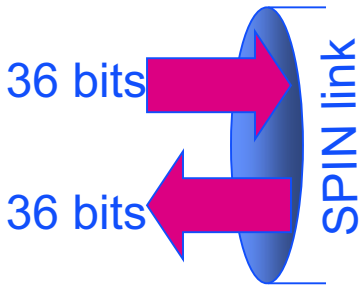
Wormhole routing:

- **routers forward packets ASAP**
- **packets span several routers**



SPIN : The bidirectional link

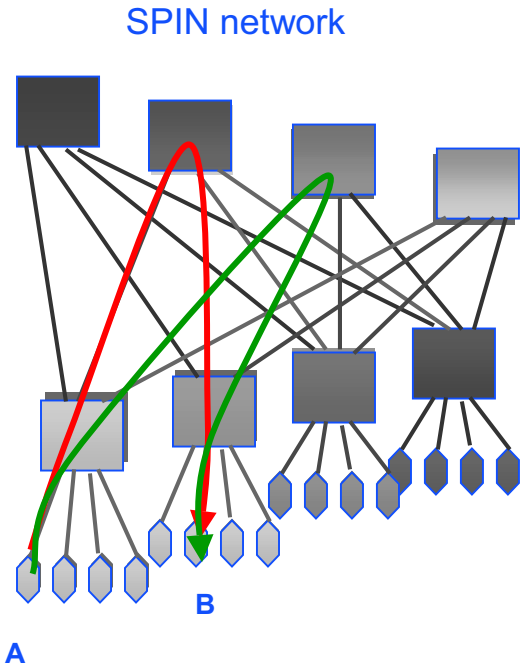
Tags **Data**



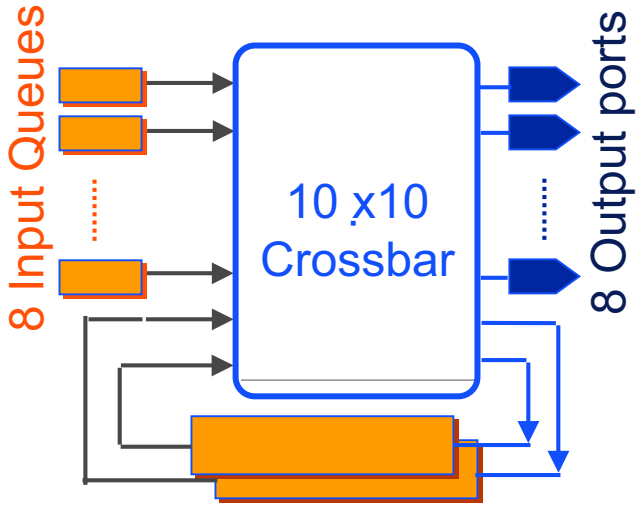
- 2 * 32-bits data links @ 200 MHz
  => peak link Bandwidth = 12.8 Gbit/s
- Asynchronous, credit based flow control
  => easy floorplan routing & timing in DSM process

Adaptive routing

- Upward routing is adaptive
- Downward routing is deterministic
- Round- Robin strategy

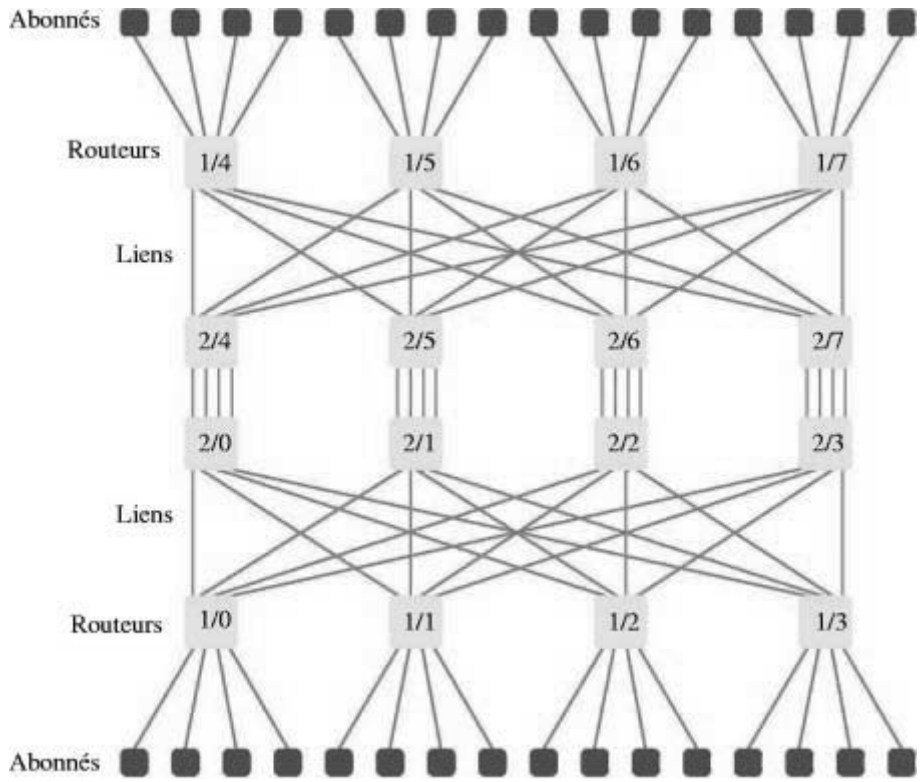


The RSPIN router



- © 10 * 10 internal crossbar
- © Elementary output queuing, tuned for 64-byte payloads
- © 2 kbits of memory
- © Peak bandwidth = 50 Gbit/s
- © Pipe-lined arbitration
- © Latency = 2.5 cycles

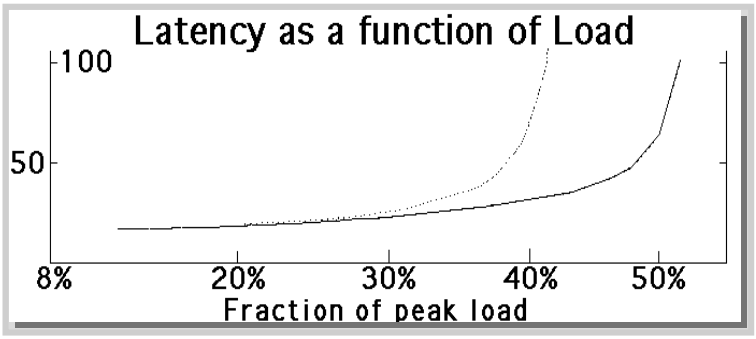
A 32 ports SPIN network



SPIN : performances evaluation

- Bit-true, cycle-true simulation for multi-million cycles for a 32 ports « raw » SPIN network:
- Worst-case workload (random, non-local)

- **50% of peak bandwidth**
- **3+3 Gbit/s per port**



Outline

- Introduction
- Virtual Component Interconnect (VCI)
- The SPIN micro-network
- VCI over SPIN
- SPIN physical implementation
- Some issues in micro-networks
- Conclusion

VCI / SPIN

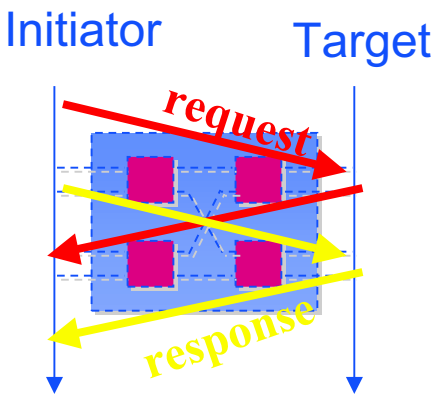
VCI / SPIN Wrappers

- VCI provides a single, shared address-space for existing, bus oriented, IP cores.

- VCI split transactions can be mapped into SPIN request and response packets.

-The round- trip latency (~ 30 cycles) can be hidden by overlapping transactions.

- Embedded RAM can be split in multiple banks for concurrent accesses



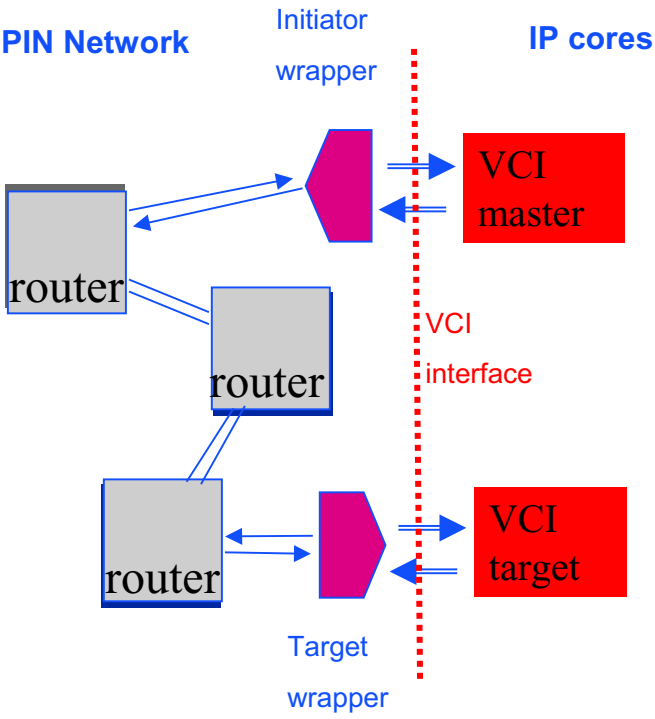
VCI / SPIN Wrappers

VCI master wrapper

- 16 concurrent requests
- 1- 3 cycles latency
- unconstrained packet length
- fully asynchronous mode

VCI target wrapper

- only one request
- 1- 3 cycles latency
- fully asynchronous mode

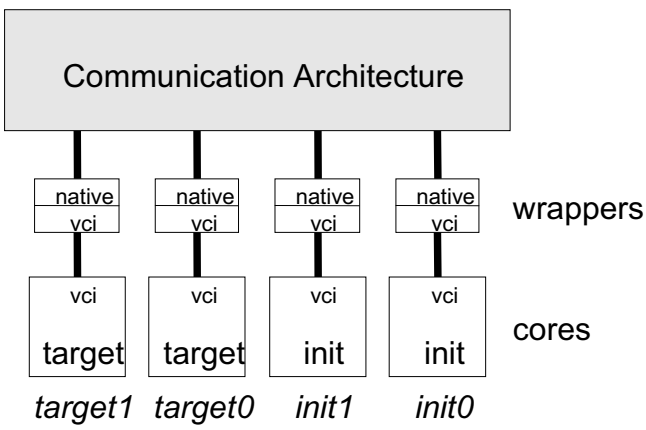


SPIN / PIbus

© Simulation of multi-master / multi-target architectures running a synthetic workload.

© All components are described as cycle-true bit-true models for SystemC.

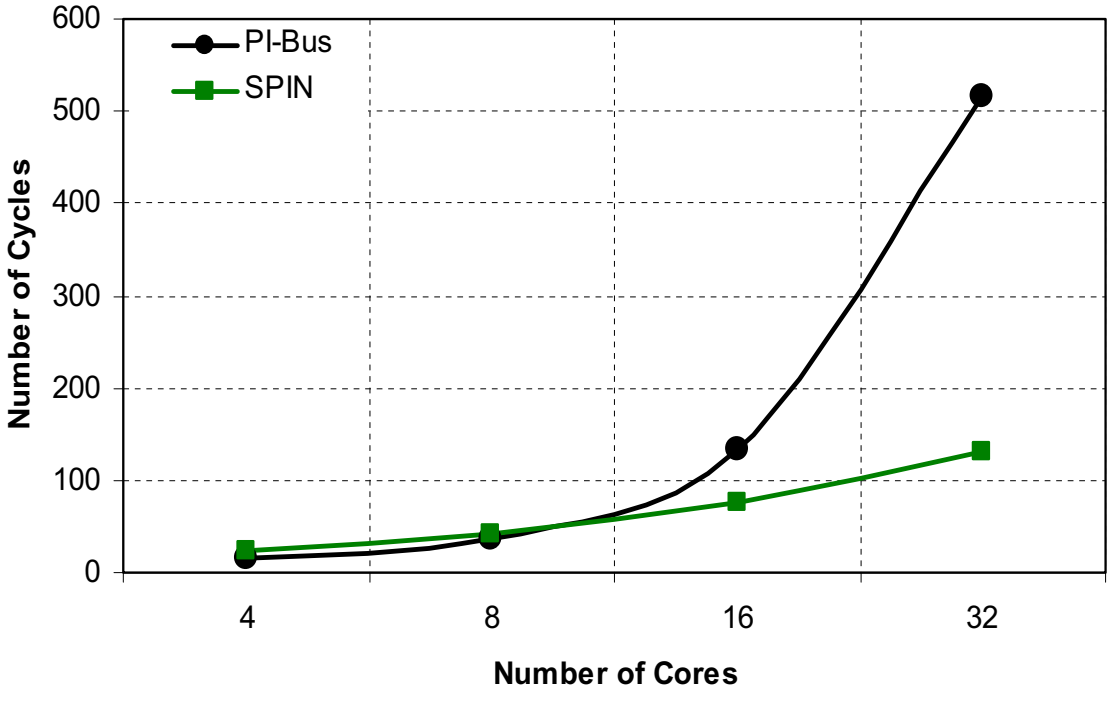
© The simulation environment is the LIP6 CASS simulator.



Total latency / Number of cores

- The variable parameter N is the number of cores.
- N cores : N/2 initiators and N/2 targets.
- Pooling : each initiator sends a write request to each target (for both architectures).
- Measure the total number of cycle to complete a pooling.

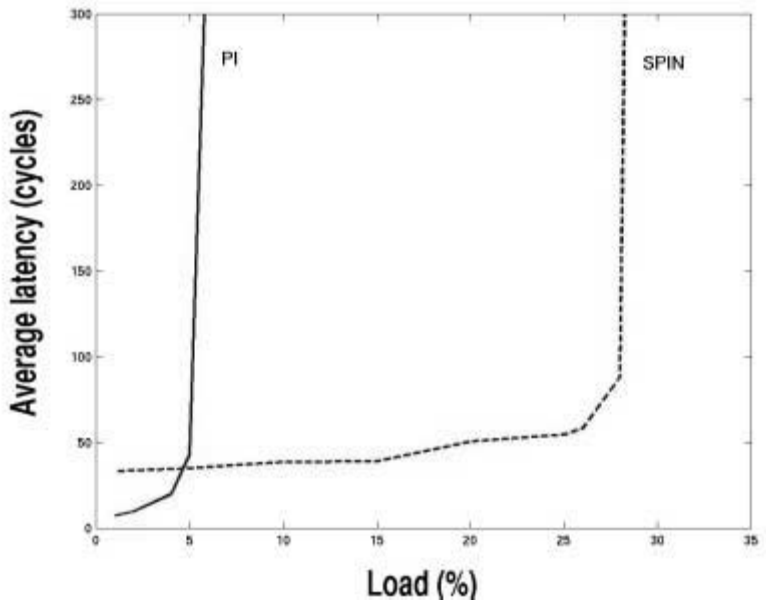
SPIN / PI-Bus Results 2



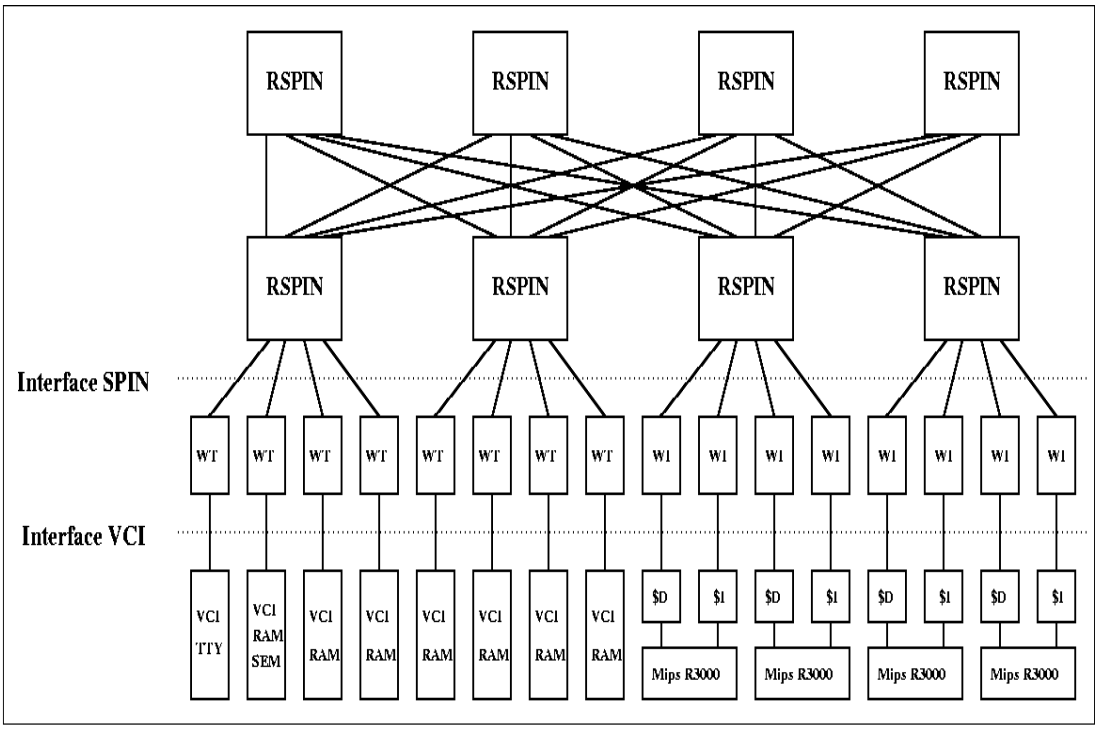
Latency / Load

- The variable parameter is the offered load
- 32 cores : 16 initiators and 16 targets.
- Random traffic : The 16 initiators send randomly read request (8 words = a cache line) to the 16 targets (for both architectures).
- Measure the average latency for a transaction.

SPIN / PI-Bus Results 2



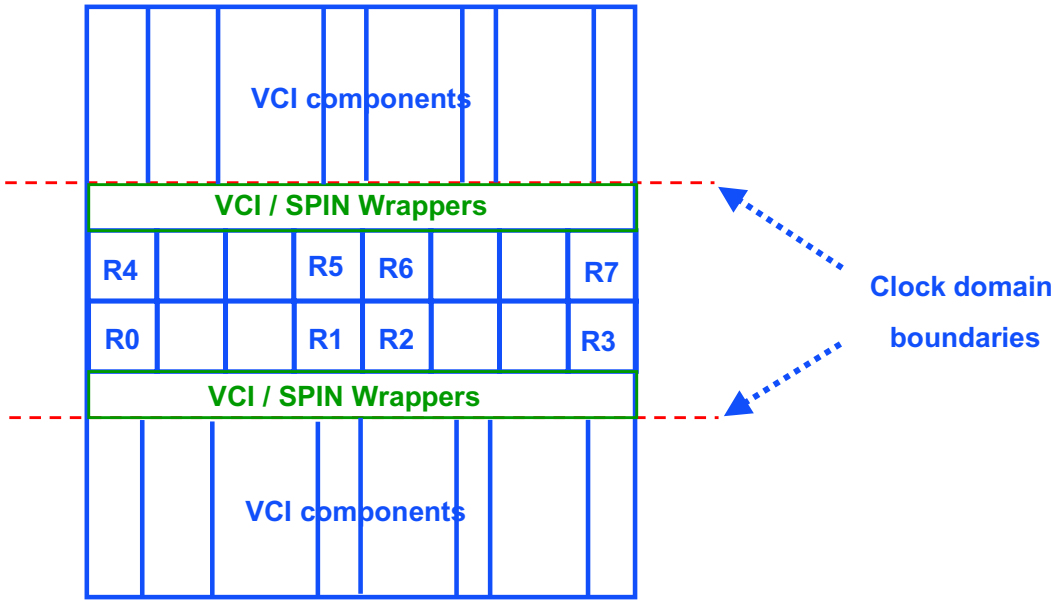
A SPIN based quadri-processor architecture



Outline

- Introduction
- Virtual Component Interconnect (VCI)
- The SPIN micro-network
- VCI over SPIN
- SPIN physical implementation
- Some issues in micro-networks
- Conclusion

A centralized micro-network



SPIN32

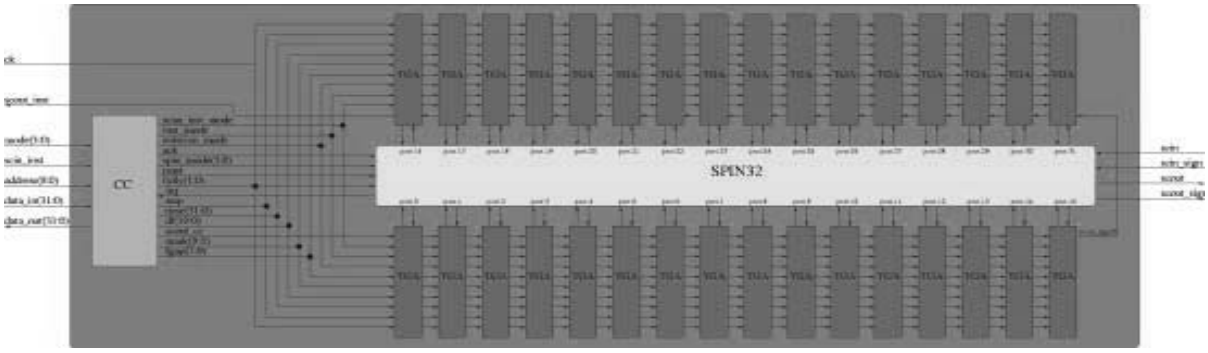
The SPIN32 evaluation chip has been developed in cooperation with STMicroelectronics, with the following goals :

- **Provide a silicon proof of the *SPIN* concept in 0.13µ CMOS**
- **Confirm the simulation results : latency & throughput**
- **Measure the maximum clock frequency**
- **Measure the reliability under several stress environments**
- **Measure the power consumption**

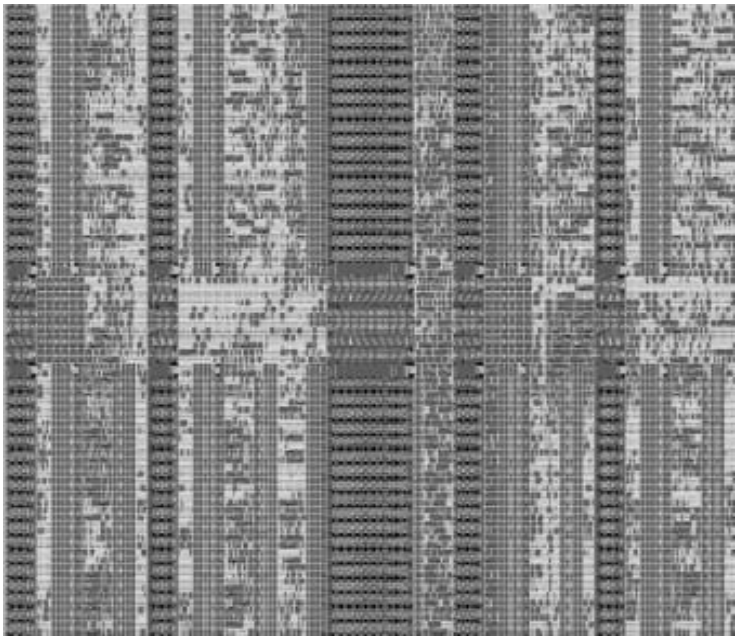
Evaluation chip architecture

The 32 ports SPIN micro-network (*spin32* macrocell) is surrounded by a dedicated instrumentation logic.

This evaluation chip will be fabricated by *ST Microelectronics* in *0.13 µ* CMOS process.



RSPIN router layout

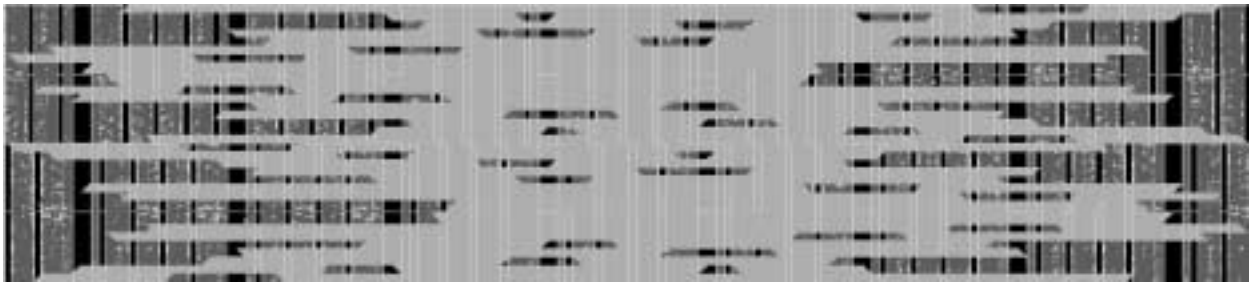


© Symbolic layout for process portability

© Area is 0.24 mm² in *CMOS 0.13µ*

SPIN32 macrocell layout

- © Symbolic layout for process portability
- © Inter-routers wires are routed over cells in metal 4, 5 and 6
- © 1 390 464 transistors
- © 4.6 mm^2 in *STMicroelectronics 0.13 μ* with 6 metal layers



Outline

- Introduction
- Virtual Component Interconnect (VCI)
- The SPIN micro-network
- VCI over SPIN
- SPIN physical implementation
- Some issues in micro-networks
- Conclusion

Micro-Networks : some issues

- increased latency
- dead-lock prevention
- cache consistency
- synchronisation primitives

The latency issue

The illimited bandwidth provided by on chip switched networks has a cost : Increased latency !

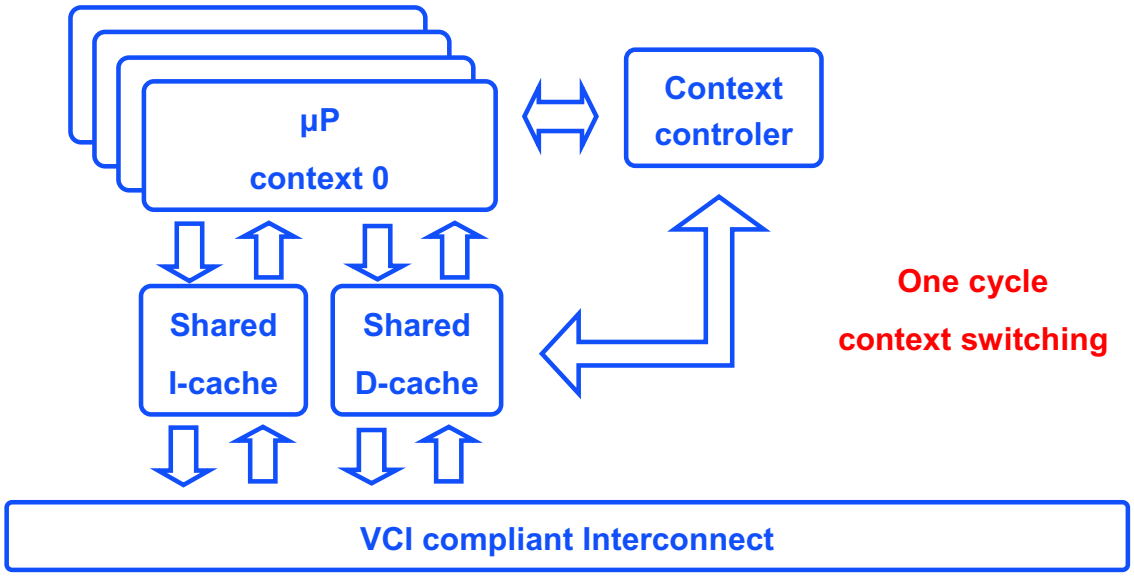
16 ports network	~ 30 cycles
32 ports network	~ 35 cycles
64 ports network	~ 40 cycles

cost of a cache MISS in the SPIN network

Possible solution : hardware support for fast context switching in multi-threaded applications

Fast context switching : the hardware view

- A single microprocessor contains 4 harware contexts
- Both Instruction and Data caches are shared by all contexts



One cycle
context switching

Fast context switching : the software view

- The embedded software application must be multi-threaded (using for example the POSIX threads).
- The operating system must support symmetric multiprocessor (SMP) architectures.

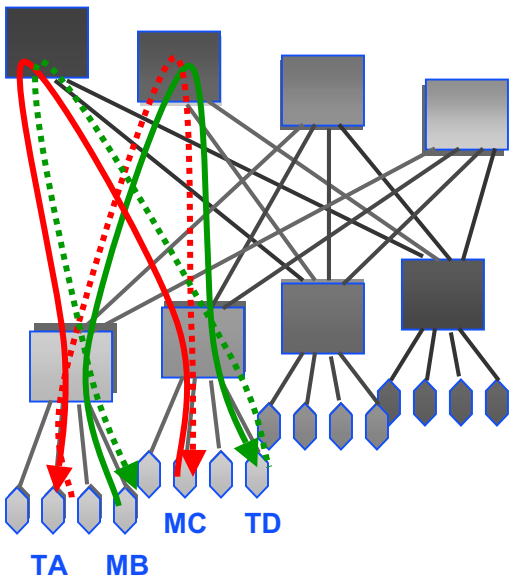
=> Each hardware context is seen as a
separate processor.

The dead- lock issue

- For one-sided packets,
the SPIN network is dead-lock-safe

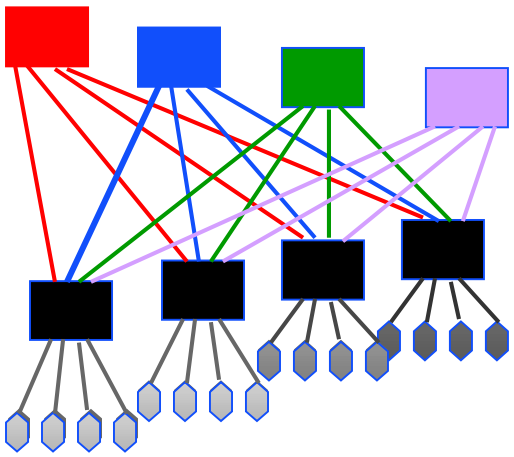
but…

- VCI supports multiple
simultaneous transactions.
- The request/response loops
can create deadlocks...



Dead-lock prevention

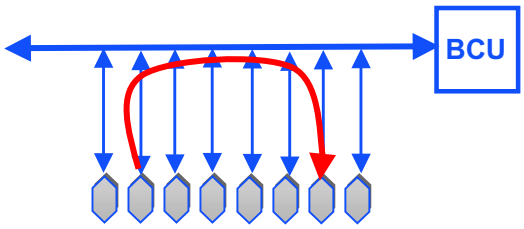
- A very general rule is
to have separated data-paths
for requests and responses
- It is possible to define
separated « sub-networks »
in the SPIN fat-tree.
- the routing algorithme must
take into account the packet
type (request / response)



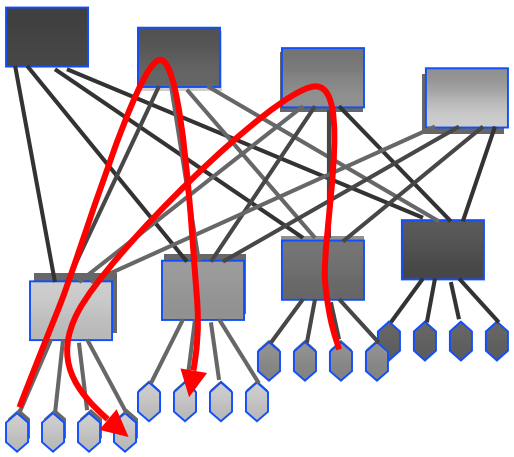
The cache consistency issue

Shared bus

All cache controllers can « snoop »
the system bus.
=> hardware cache consistency



Multi-stage network



Snoop is not posible...

Cache management with micro-networks

The cache management in micro-network based SOC
is an open issue…
Several answers are possible :

- **supress the problem : no caches !**
  => video applications, network processing
- **hardware consistency (directory based)**
  => very costly
- **software consistency**
  => this will impact both application & system software
  (uncached segment for shared data,
  controlled task migration, etc…)

Synchronisation primitives

The problem is to provide atomic « Read then Write » operations used for inter-process synchronization.

- Traditionnal bus locking method cannot be used !
- The atomic acces should be garanteed by the target, using the VCI « Linked Read » command.

=> Hardware impact : The architecture should provide a « semaphore engine » with queue manahgement.

=> Software impact : All physical locks should be stored in a specific memory segment.

Conlusion

- System on chip : old problems requiring new solutions
- On chip micro-networks will be a key component
- Hardware / software codesign is mandatory
- An open modelisation plat-form will be available soon

=> Action Spécifique SOCLIB du CNRS