Arbitrage d'un bus La gestion du partage d'un bus se fait par le contrôleur du bus qui possède une logique d'arbitrage permettant d'allouer et désallouer le bus. Cet arbitrage se fait directement au niveau hardware, selon plusieurs stratégies courantes :

1. Priorité fixe ou *daisy chaining* : les différentes unités rattachées au bus reçoivent une priorité fixe qui règle leur accès au bus. Si deux unités demandent simultanément le bus, celle de priorité maximum l'emporte. Les unités sont d'ailleurs réparties physiquement le long du bus selon l'ordre de ces priorités (daisy chaining). Cette technique est simple à mettre en oeuvre mais pas symétrique. Une unité peut en principe attendre un temps infini avant d'accéder au bus.
2. Découpage en tranches : chaque unité reçoit un intervalle de temps d'accès au bus, selon un ordre fixe. Si une unité n'utilise pas le bus pendant cette période d'allocation, le bus reste inactif. Cela conduit à une faible utilisation des ressources mais garantit un temps d'attente fini et une symétrie entre les unités. Cette stratégie est plus adaptée à des systèmes synchrones qui permettent un meilleur usage des tranches de temps.
3. Priorités dynamiques : dans ce cas, on attribue des priorités comme dans le premier cas, mais celles-ci peuvent changer au cours du temps. Par exemple, le premier arrivé est le premier servi pour acquérir une priorité plus grande. Cette stratégie est techniquement plus difficile à réaliser mais utilise efficacement les ressources de façon symétrique et minimise les temps d'attente.

3.3.2 Le crossbar switch

Le réseau crossbar, ou matrice de points de croisement est l'opposé du bus : il a une haute complexité et permet un trafic important de message. Un crossbar switch est une grille dont chaque point d'intersection est un commutateur qui permet de relier entre eux deux éléments quelconques du réseau (par exemple un processeur à une mémoire, ou deux processeurs ensemble). C'est une solution couramment utilisée dans les ordinateurs haut de gamme ayant plusieurs processeurs et une mémoire partagée composée de plusieurs bancs. Un crossbar permet aussi de réaliser une topologie en «étoile» dans une architecture à mémoire distribuée, ainsi que l'illustre la figure 3.24.

Un crossbar peut être considéré comme un commutateur à N entrées et N sorties et permettant de réaliser n'importe quelle permutation $i \rightarrow f(i)$ de l'ensemble des entrées dans l'ensemble des sorties. Il y a $N!$ états différents du crossbar produisant des communications *one-to-one* différentes. La figure 3.25 montre un crossbar switch avec N processeurs en entrée et N modules mémoire en sortie.

La connectivité d'un crossbar est donc totale. Mais, si tous les chemins sont possibles, seul l'accès à des noeuds non occupés est possible. Si 2 processeurs

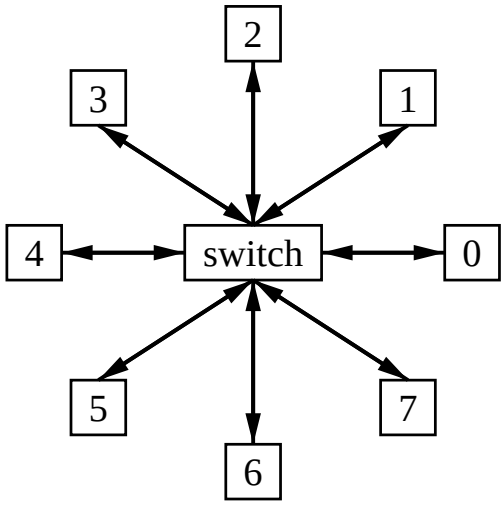

FIGURE 3.24 – Plusieurs processeurs sont courament interconnecté par un switch central qui a la fonctionnalité d'un crossbar.

veulent accéder à un même noeud, l'un d'eux devra attendre. Par contre, s'il n'y a pas de tels conflits, N chemins indépendants et sans intersections pourront être simultanément établis pour un élément de commutation matriciel de taille $N \times N$. Le temps de latence est dépendant du temps de réaction des commutateurs situés aux intersections de la grille et est indépendant de N.

Le coût d'un tel réseau croît comme N^2 en ce qui concerne les commutateurs et les fils de connexions, ce qui limite l'usage de ce type d'interconnexion à des systèmes comprenant relativement peu d'entrées.

La figure 3.26 donne une réalisation simple d'un crossbar à l'aide de commutateurs élémentaires 2×2 placés en chaque point (i, j) de croisement. Ces commutateurs sont caractérisés par deux états internes c_{ij} indiquant une commutation directe ($c_{ij} = 0$) ou croisée ($c_{ij} = 1$), comme l'indique la figure 3.26. Pour relier l'entrée E_i à la sortie S_j il suffit que

$$c_{ij} = 1, \qquad c_{kj} = 0, k > i \qquad \text{et } c_{i\ell} = 0, \ell < j$$

Dans l'exemple de la figure 3.26, on établit les connexions $E_1 \to S_2$, $E_2 \to S_1$ et $E_3 \to S_3$. Cependant, ce réseau ne peut réaliser des broadcasts *one-to-many* où une même entrée est connectée simultanément à plusieurs sorties. De même, si plusieurs entrées veulent envoyer simultanément un message sur la même sortie, le réseau de la figure 3.26 va systématiquement privilégier l'entrée dont l'indice de ligne est le plus grand.

Pour gérer correctement ces conflits il faut introduire de la logique supplémentaire. Ainsi, dans un crossbar, chaque point de croisement a la complexité d'un bus, ce qui explique le coût important d'un tel réseau.

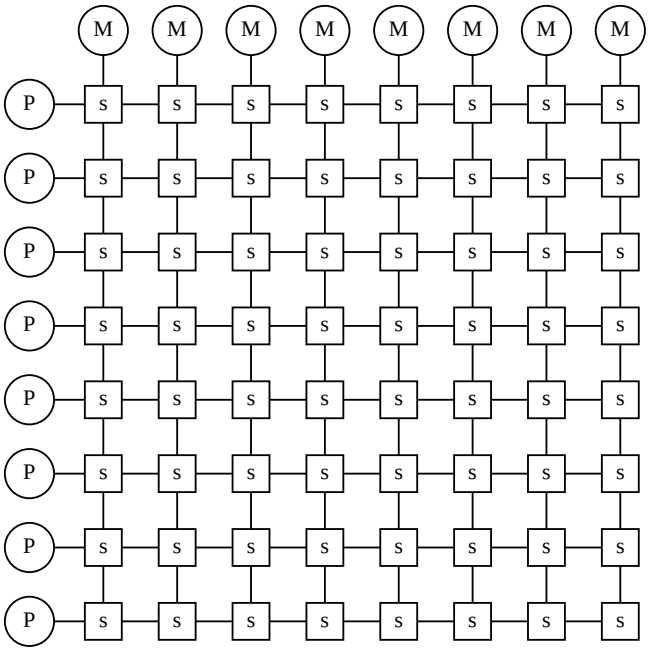

FIGURE 3.25 – *Schéma d'un crossbar* 8×8. *Les éléments notés s apparaissant aux points de croisement sont les switches. Cet exemple montre une situation où 8 processeurs sont reliés à 8 mémoires.*

3.3.3 Les réseaux multiétages

Ce type de réseau se situe entre le bus et l'interconnexion matricielle. Il est plus complexe et plus performant que le premier mais moins onéreux que le deuxième. L'idée de l'interconnexion multiétages (*multistage crossbar switch* en anglais) est de remplacer un commutateur matriciel $N \times N$ par un ensemble de couches comprenant des commutateurs 2×2 (ou éventuellement un peu plus grand). Par exemple, pour un système de 4 entrées et 4 sorties, on peut imaginer le dispositif illustré sur la figure 3.27.

Dans cette figure, chacune des boîtes est un commutateur 2×2 dont les fonctions internes de commutation sont schématisées sur la figure 3.28. Elles permettent une commutation directe ou croisée.

Ce dispositif permet de relier une quelconque des quatre entrées avec une quelconque des quatre sorties. Par exemple, pour mettre en contact l'entrée 1 avec la sortie 4 il faut avoir une commutation croisée dans les commutateurs en haut à gauche et en bas à droite. Cependant, ce choix contraint les autres connexions : l'entrée 2 ne peut alors être connectée qu'à la sortie 1 ou 2, selon l'état du commutateur en haut à droite. Ce système ne peut donc pas réaliser toutes les permutations des quatre entrées sur les quatre sorties, à la différence d'un crossbar 4×4. Il permet néanmoins d'assurer quatre communications simultanées.

Il faut remarquer qu'il est courant dans les machines vectorielles ou parallèles