

# Modélisation du protocole MSI avec Cubicle

Le but de ce TP est de modéliser et de prouver des programmes concurrents en utilisant Cubicle: <http://cubicle.lri.fr>.

## Le protocole MSI

Afin d’assurer la cohérence des données entre les caches d’un micro-processeur, on utilise des protocoles de cohérence de cache.

Dans ce TP, nous allons modéliser le protocole MSI. Dans ce protocole, un cache peut être dans trois états: **Modified**, **Shared** ou **Invalid**.

- **Modified**. Les données qui sont actuellement dans le cache sont « sales » : elles ont été modifiées par rapport aux données qui sont en mémoire centrale.
- **Shared**. Indique que les données qui sont dans le cache sont aussi potentiellement présentes dans d’autres caches.
- **Invalid**. Indique que les données qui sont dans le cache ne sont pas valides.

Nous décrivons maintenant les changements d’état des caches en cas de requêtes de lectures et d’écritures.

- Si un processeur effectue une requête de lecture alors que son cache est dans l’état **Invalid**, il y a un *read miss*. L’état du cache passe dans l’état **Shared** et simultanément, les caches qui étaient dans l’état **Modified** passent dans l’état **Shared**.
- Si un processeur effectue une lecture alors que son cache est dans un autre état, il y a un *read hit* et il reste dans son état.
- Si un processeur effectue une requête d’écriture alors que son cache est dans l’état **Invalid**, il y a un *write miss*. L’état du cache passe dans l’état **Modified** et tous les autres passent dans l’état **Invalid**.
- Enfin, si un processeur effectue une requête d’écriture alors que son cache n’est pas invalide, il y a deux cas possibles :
  - si son cache est dans l’état **Shared**, il passe en **Modified** et tous les autres passent dans l’état invalide ;
  - si son cache est dans l’état **Modified**, il reste dans cet état.

Dans le tableau ci-dessous, les cases marquées d’une croix marquent les états qui ne sont pas permis pour tout couple de caches.

	M	S	I
M	X	X	
S	X		
I			

Vous trouverez à l'adresse suivante une modélisation OCaml d'une version simple de MSI : [https://www.lri.fr/~conchon/FIIL/msi\\_basic.ml](https://www.lri.fr/~conchon/FIIL/msi_basic.ml)

**Question 1.** Modéliser en Cubicle cette version simplifiée de MSI et montrer qu'elle est sûre.

Vous trouverez à l'adresse suivante une modélisation OCaml d'une version plus complexe de MSI : <https://www.lri.fr/~conchon/FIIL/msi.ml>

**Question 2.** Modéliser en Cubicle cette version de MSI et montrer qu'elle est sûre.

**Question 3.** Comment rendre cette version *moins* atomique ?