

Modélisation et vérification de systèmes concurrents

Architecture multiprocesseur à mémoire partagée

Kimmeng Ly Max Eliet

Sorbonne Université Sciences

Encadrante : E. Encrenaz

Introduction

On considère un système multiprocesseur à mémoire partagée. Le système est muni d'une hiérarchie mémoire à deux niveaux :

On considère un système multiprocesseur à mémoire partagée. Le système est muni d'une hiérarchie mémoire à deux niveaux :

Mémoire centrale

Stocke les instructions, données, pile des programmes en cours d'exécution et du système d'exploitation.

On considère un système multiprocesseur à mémoire partagée. Le système est muni d'une hiérarchie mémoire à deux niveaux :

Mémoire centrale

Stocke les instructions, données, pile des programmes en cours d'exécution et du système d'exploitation.

Caches privés

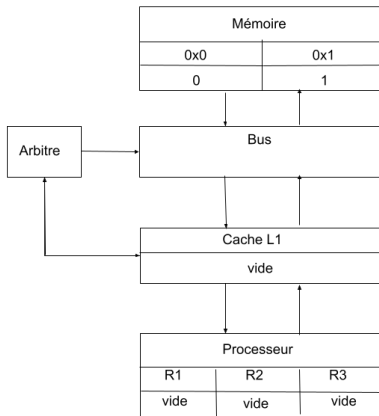
Associés à chaque processeur et comprennent deux parties distinctes :

- Caches d'instructions
- Caches de données

Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

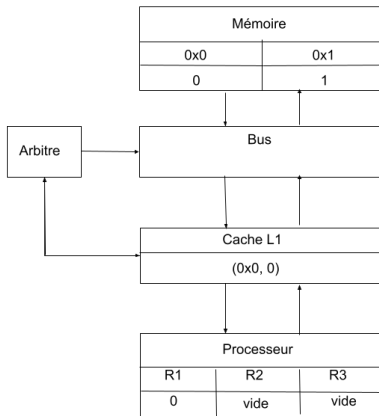


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

- `ld R1, [0]`

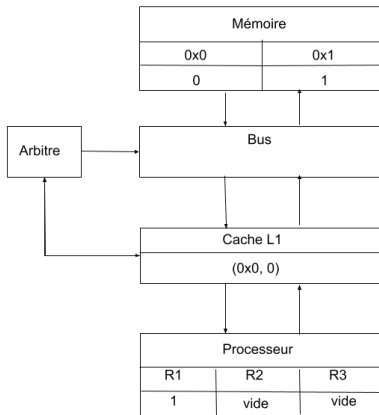


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

- `ld R1, [0]`
- `add R1, R1, 1`

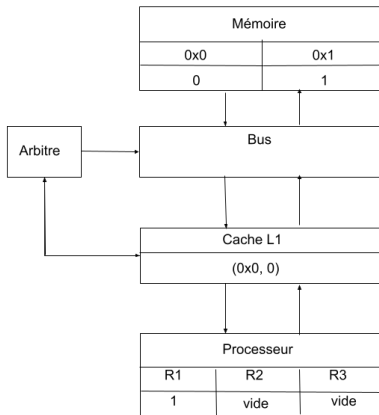


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

- `ld R1, [0]`
- `add R1, R1, 1`
- `st R1, [1]`

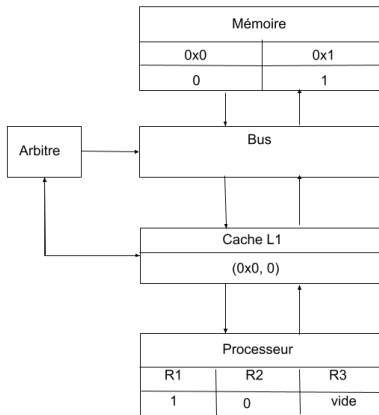


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

- `ld R1, [0]`
- `add R1, R1, 1`
- `st R1, [1]`
- `ld R2, [0]`

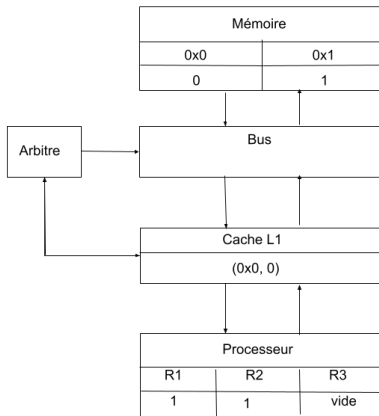


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

- `ld R1, [0]`
- `add R1, R1, 1`
- `st R1, [1]`
- `ld R2, [0]`
- `add R2, R2, 1`

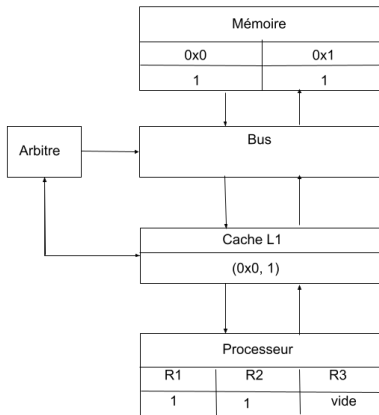


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

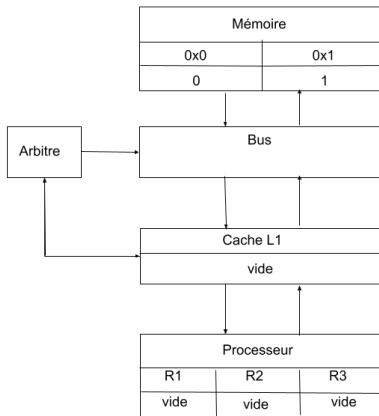
- `ld R1, [0]`
- `add R1, R1, 1`
- `st R1, [1]`
- `ld R2, [0]`
- `add R2, R2, 1`
- `st R2, [0]`



Etude du protocole et des accès aux données partagées

Cas monoprocesseur : éviction du cache / écriture write-through

P1b :

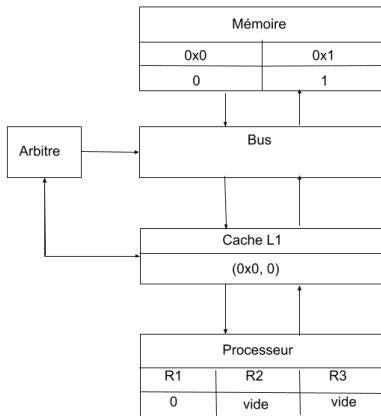


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : éviction du cache / écriture write-through

P1b :

- `ld R1, [0]`

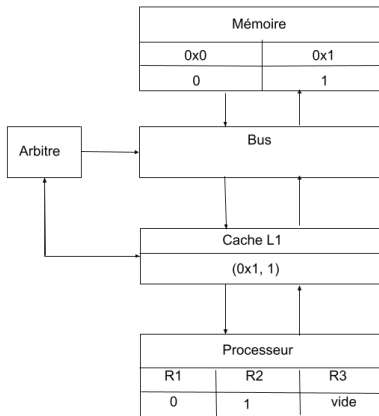


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : éviction du cache / écriture write-through

P1b :

- ld R1, [0]
- ld R2, [1]

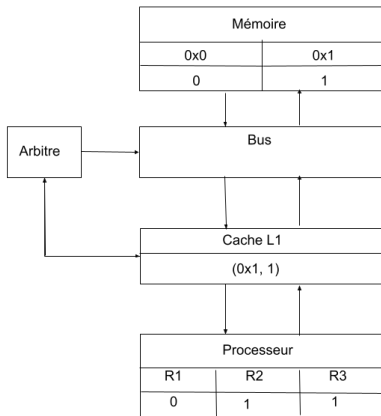


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : éviction du cache / écriture write-through

P1b :

- `ld R1, [0]`
- `ld R2, [1]`
- `add R3, R1, R2`

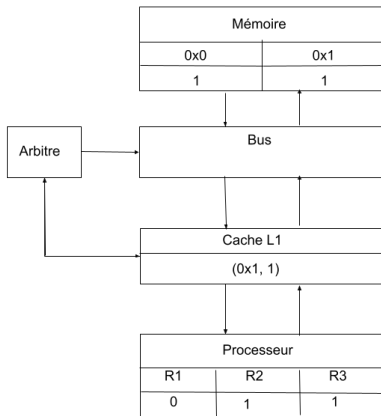


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : éviction du cache / écriture write-through

P1b :

- `ld R1, [0]`
- `ld R2, [1]`
- `add R3, R1, R2`
- `st R3, [0]`



Cas multiprocesseur : partage en lecture / une écriture

P1a :

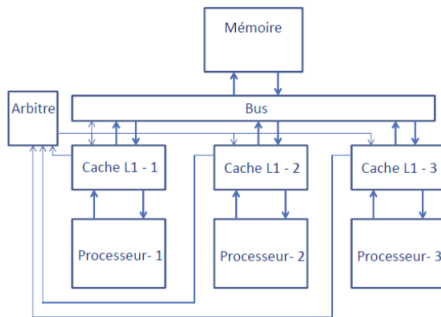
- Id R1, [0]

P2a :

- Id R1, [0]

P3a :

- inactif



Cas multiprocesseur : partage en lecture / une écriture

P1a :

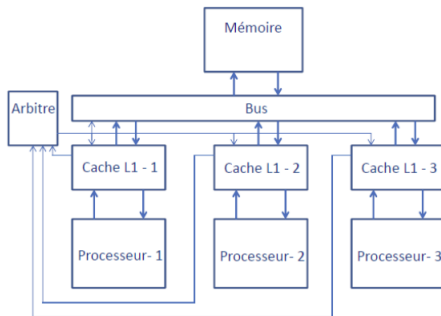
- add R1, R1, 1

P2a :

- ...

P3a :

- inactif



Cas multiprocesseur : partage en lecture / une écriture

P1a :

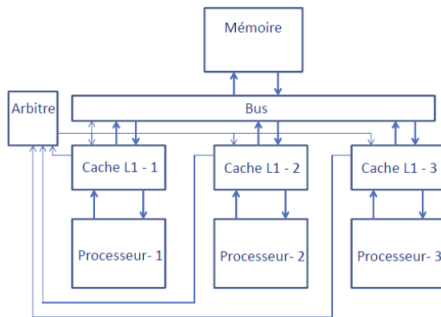
- st R1, [0]

P2a :

- ...

P3a :

- inactif



Cas multiprocesseur : partage en lecture / une écriture

P1a :

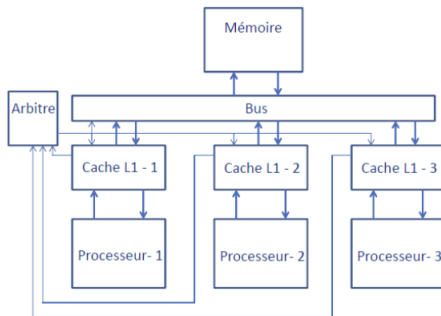
- ...

P2a :

- add R2, R1, 1

P3a :

- inactif



Cas multiprocesseur : partage en lecture / une écriture

P1a :

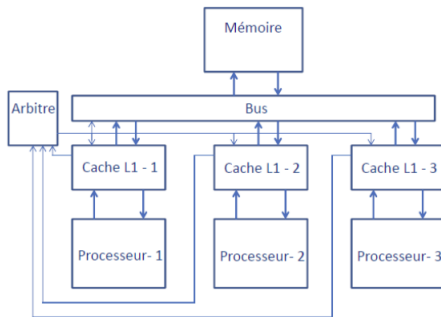
- ...

P2a :

- st R2, [1]

P3a :

- inactif



Cas multiprocesseur : boucle d'attente active / partage en lecture - écriture

P1b :

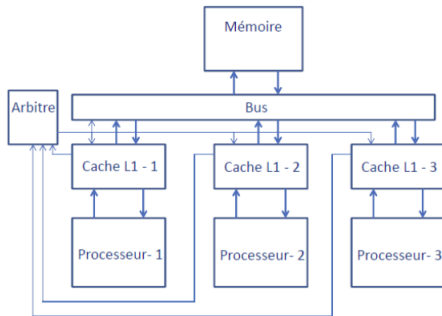
- dbt : Id R1, [0]

P2b :

- dbt : Id R1, [0]

P3b :

- ...



Cas multiprocesseur : boucle d'attente active / partage en lecture - écriture

P1b :

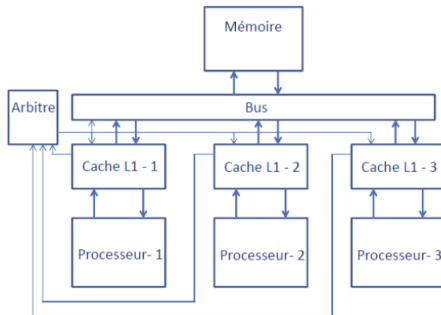
- `cmp R1, 0`

P2b :

- `cmp R1, 0`

P3b :

- ...



Cas multiprocesseur : boucle d'attente active / partage en lecture - écriture

P1b :

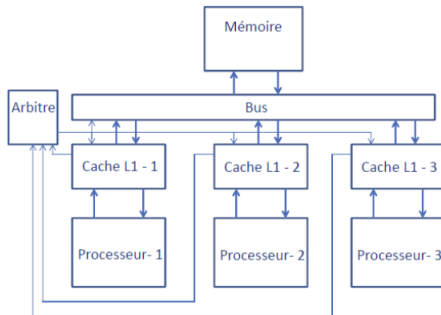
- beq dbt

P2b :

- beq dbt

P3b :

- ...



Cas multiprocesseur : boucle d'attente active / partage en lecture - écriture

P1b :

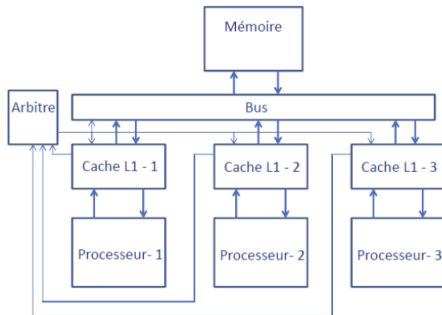
- dec R1

P2a :

- dec R1

P3a :

- ld R1, [1]



Etude du protocole et des accès aux données partagées

Cas multiprocesseur : boucle d'attente active / partage en lecture - écriture

P1b :

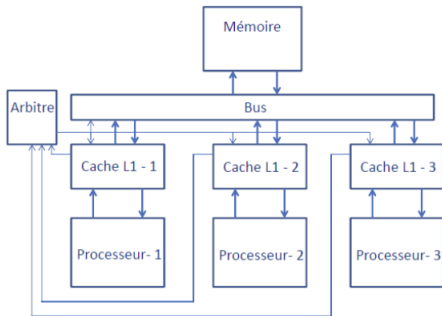
- st R1, [0]
- SC

P2b :

- st R1, [0]
- SC

P3b :

- ...



Cas multiprocesseur : boucle d'attente active / partage en lecture - écriture

Mécanisme de SNOOP

Lorsqu'une requête d'écriture circule sur le bus, chaque cache connecté doit déterminer s'il dispose du mot concerné. Dans l'affirmative, il met à jour sa copie locale avec la donnée à écrire, véhiculée sur le bus.

Cas multiprocesseur : boucle d'attente active / partage en lecture - écriture

Mécanisme de SNOOP

Lorsqu'une requête d'écriture circule sur le bus, chaque cache connecté doit déterminer s'il dispose du mot concerné. Dans l'affirmative, il met à jour sa copie locale avec la donnée à écrire, véhiculée sur le bus.

Garantir un accès exclusif aux données partagées

Un verrou pour l'accès à une donnée.