

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

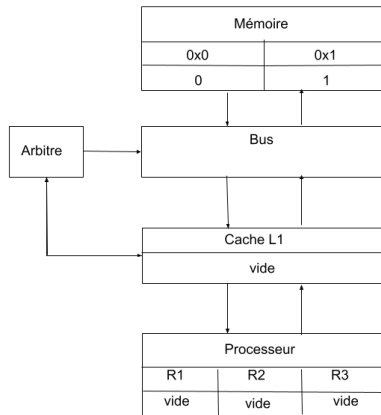
Associés à chaque processeur et comprennent deux parties distinctes :

- Caches d'instructions
- Caches de données

Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

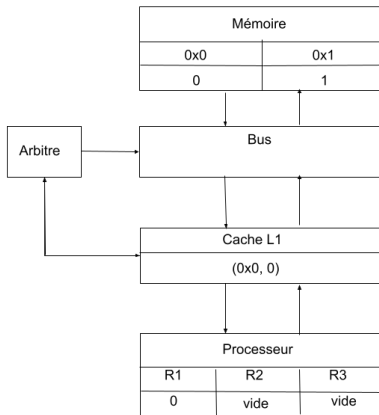


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

- `ld R1, [0]`

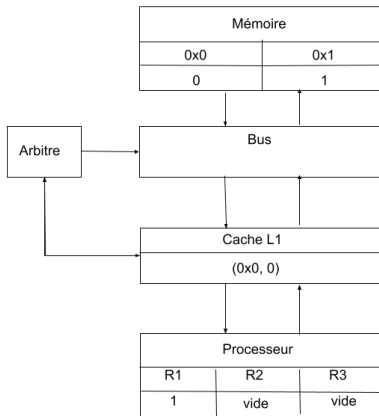


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

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

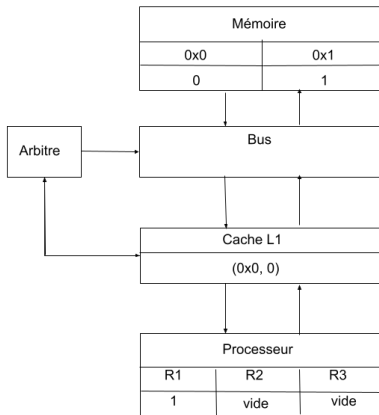


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

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

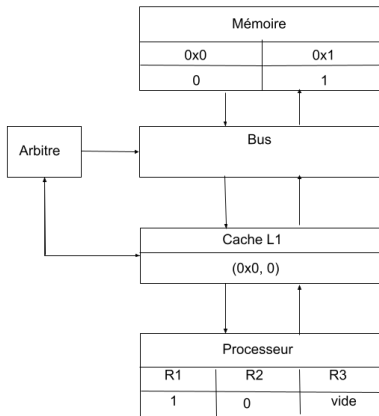


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

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

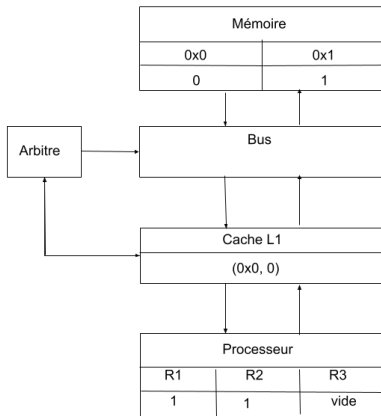


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

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

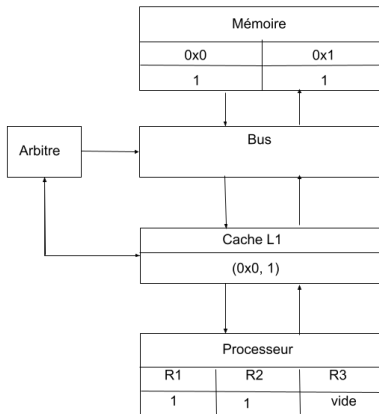


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : effet cache

P1a :

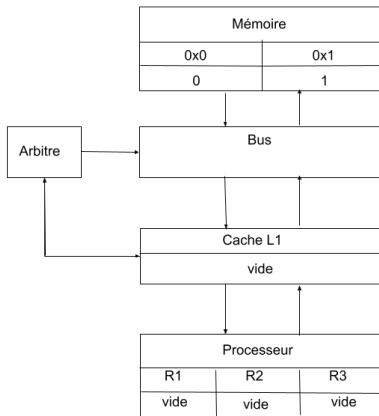
- `ld R1, [0]`
- `add R1, R1, 1`
- `st R1, [1]`
- `ld R2, [0]`
- `add R2, R2, 1`
- `st R2, [0]`



Etude du protocole et des accès aux données partagées

Cas monoprocesseur : éviction du cache / écriture write-through

P1b :

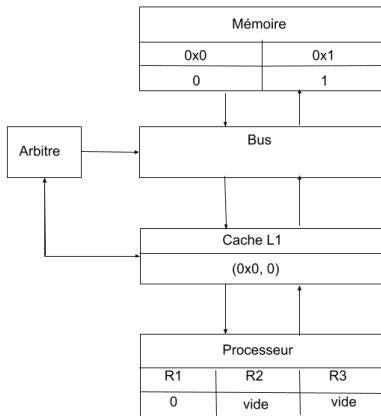


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : éviction du cache / écriture write-through

P1b :

- `ld R1, [0]`

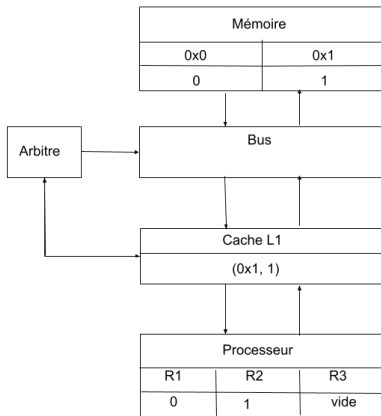


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : éviction du cache / écriture write-through

P1b :

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

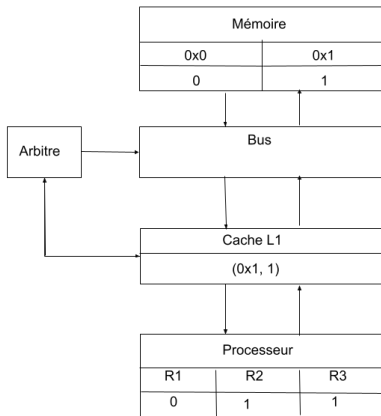


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : éviction du cache / écriture write-through

P1b :

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

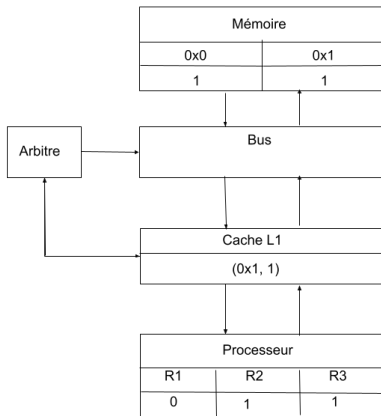


Etude du protocole et des accès aux données partagées

Cas monoprocesseur : éviction du cache / écriture write-through

P1b :

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



Etude du protocole et des accès aux données partagées

Cas multiprocesseur : partage en lecture / une écriture

P1a :

- `ld R1, [0]`

P2a :

- `ld R1, [0]`

P3a :

- `inactif`

