

PORTAGE D'UNE COUCHE DE COMMUNICATION NOYAU EN MODE UTILISATEUR

ABDELLOU Djilali

31/08/20001

LE PLAN

- I Introduction**
 - 1 L'organisme d'accueil**
 - 2 Quelques définitions**

- II Le machine MPC**

- III La couche de communication PUT noyau**
 - 1 La structure du PUT**
 - 2 Le changement de contexte**
 - 3 Les problèmes a résoudre**

- IV Le travail effectué sur cette couche**
 - 1 La nouvelle structure de PUT**
 - 2 l'accès au bus PCI**
 - 3 Le partage des ressources**
 - 4 La signalisation**
 - 5 L'initialisation**

- V Les mesures**

- VI Conclusion**

Introduction (1)

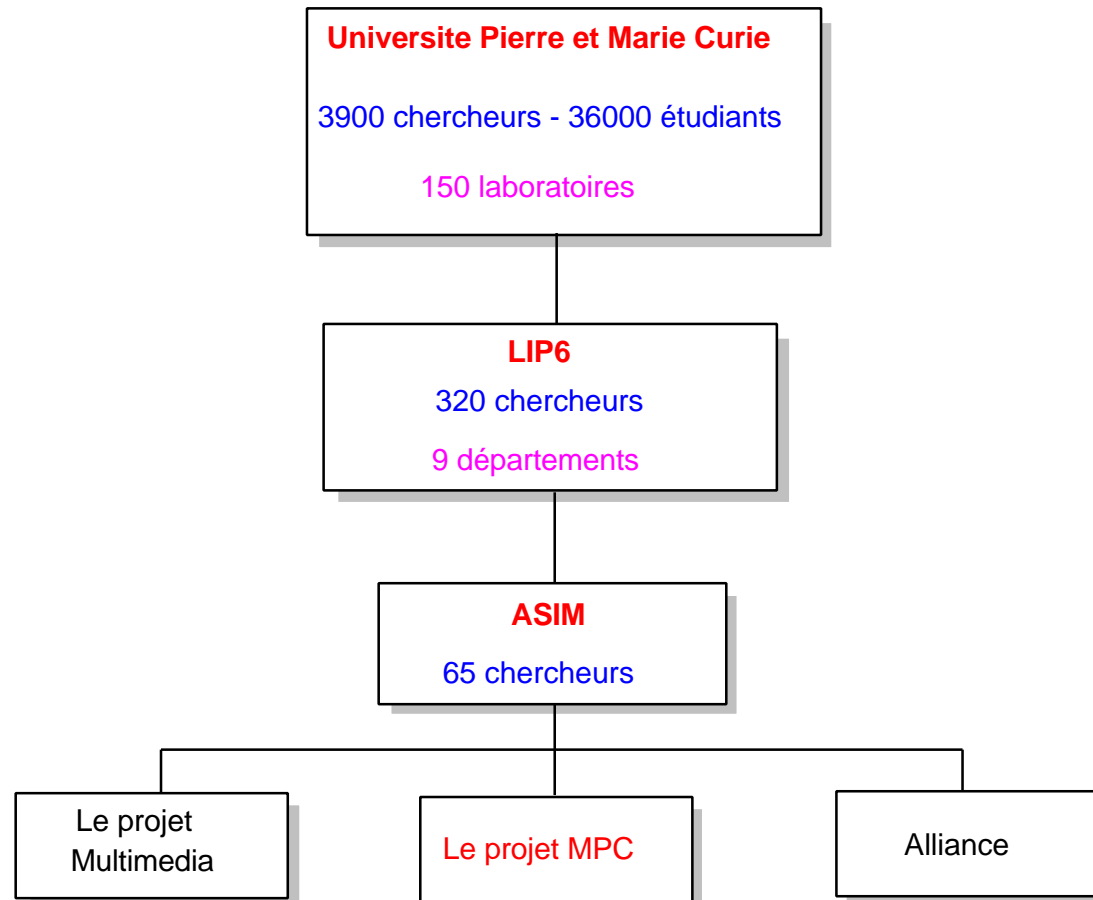


Figure 1 : de UPMCP6 au projet **MPC**

Le département ASIM

ACTIVITES DE RECHERCHE :

- le projet d'indexation multimédia
- le projet CAO de circuits et systèmes (Alliance)
- le projet **MPC**

ACTIVITES D'ENSEIGNEMENT :

- **DEA ASIM** (architecture des systèmes intégrés et micro-électronique)
- **DESS CISAN** (circuits intégrés et systèmes analogiques numériques)
- Maîtrise informatique
- Maîtrise EEA

Quelques définitions

- MODE UTILISATEUR** : mode d'exécution normal d'un processus, certaines instructions lui sont interdites
- MODE NOYAU** : mode privilégié, aucune restriction n'est imposée au noyau du système, il peut utiliser l'ensemble des instructions du processeur
- MODULE** : programme C qui s'exécute en mode noyau
- DRIVER** : programme C qui s'exécute en mode noyau et qui accède à une périphérie

Les machines parallèles

2 TYPES DE MACHINES PARALLELES :

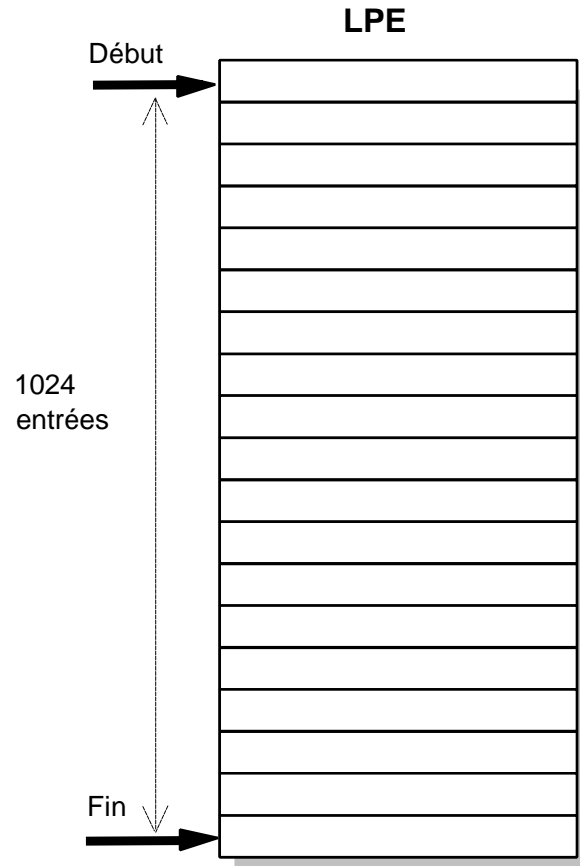
- **Les supercalculateurs** : machines spécifiques très performantes et chères (CRAY, PARAGON d'intel)
- **Les grappes de PCs** : raccordement de machines du commerce par un réseau rapide

Constitution de la carte FastHSL

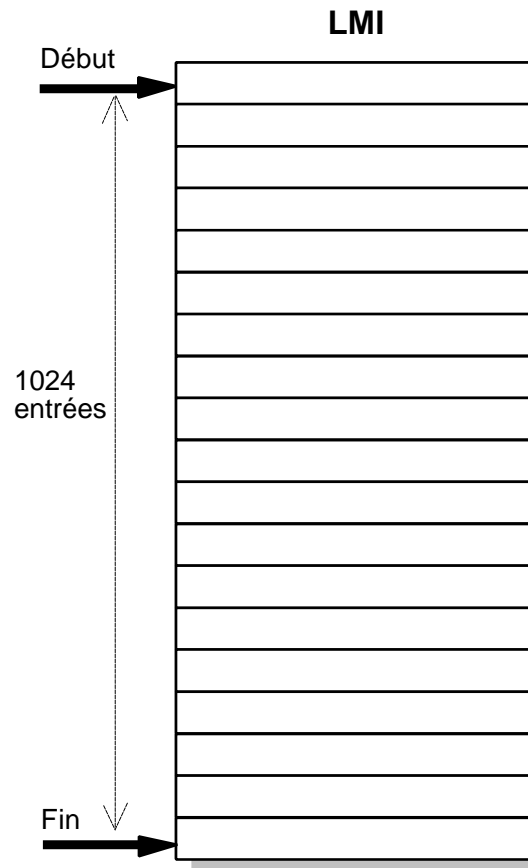
elle est constituée de deux éléments :

- le routeur Rcube à 1Gbits/S
- le contrôleur PCIDDC (implémente la primitive d'écriture distante en mémoire centrale)

Les listes

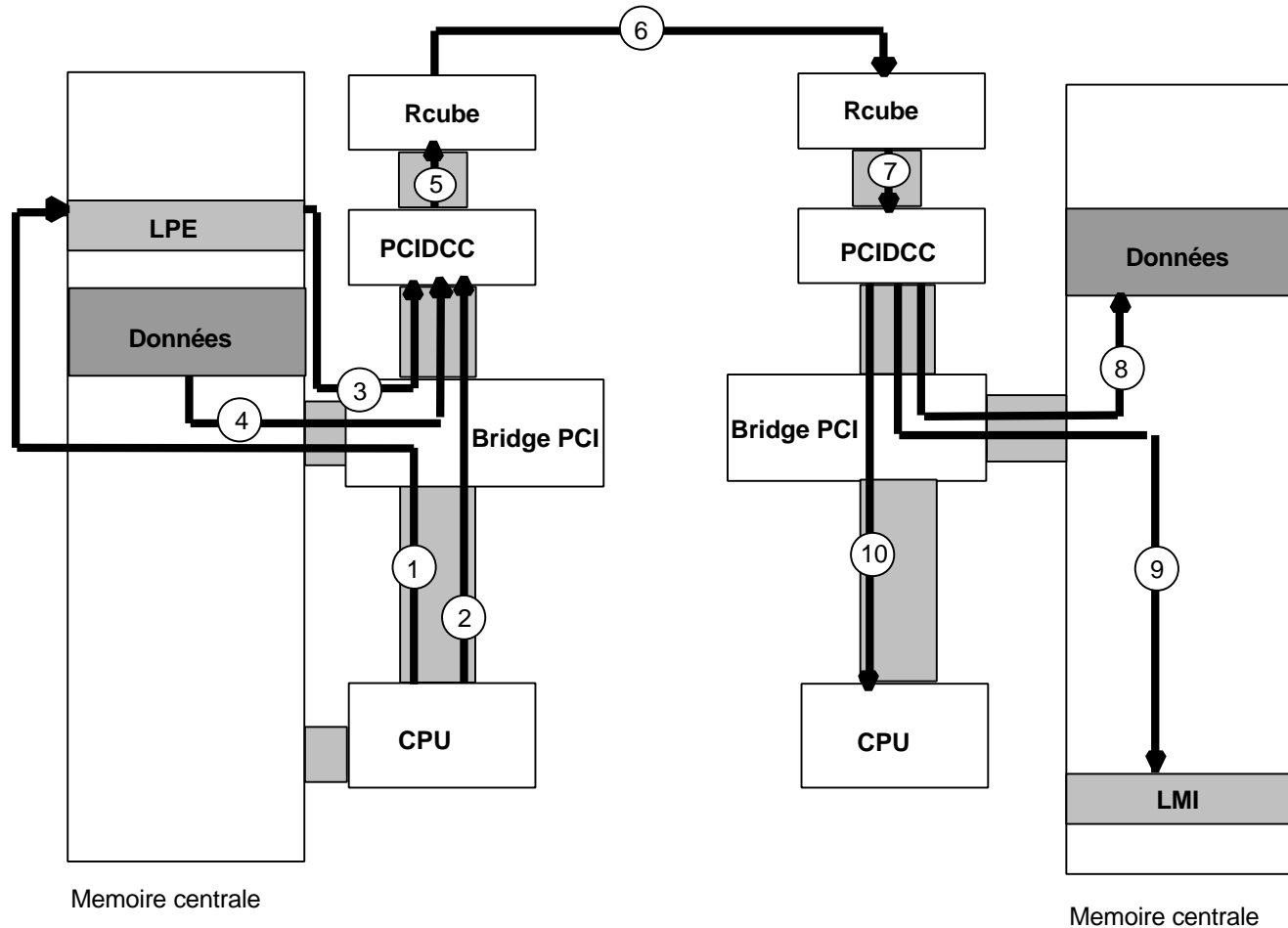


LPE : List Page Emit



LMI : List of Message Identifier

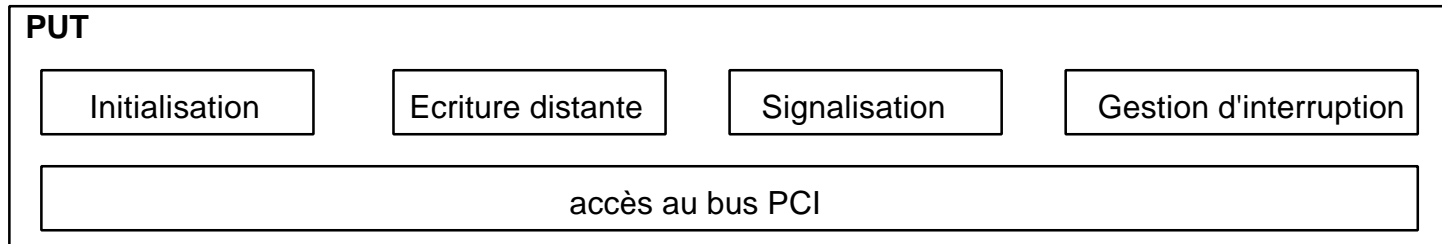
Fonctionnement de la carte FastHSL



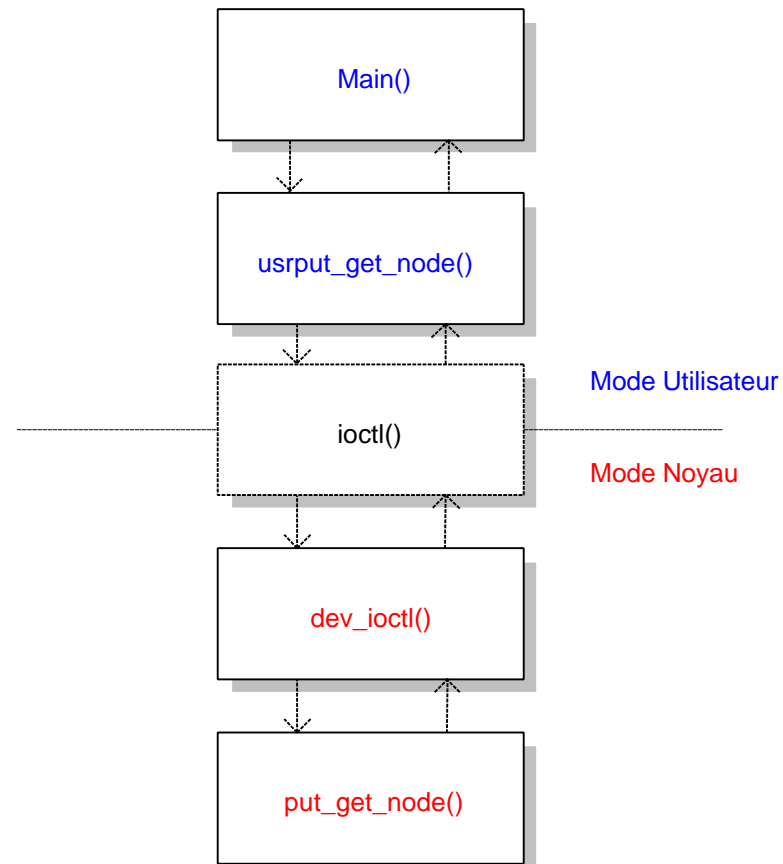
Structure du PUT noyau

MODE UTILISATEUR

MODE NOYAU



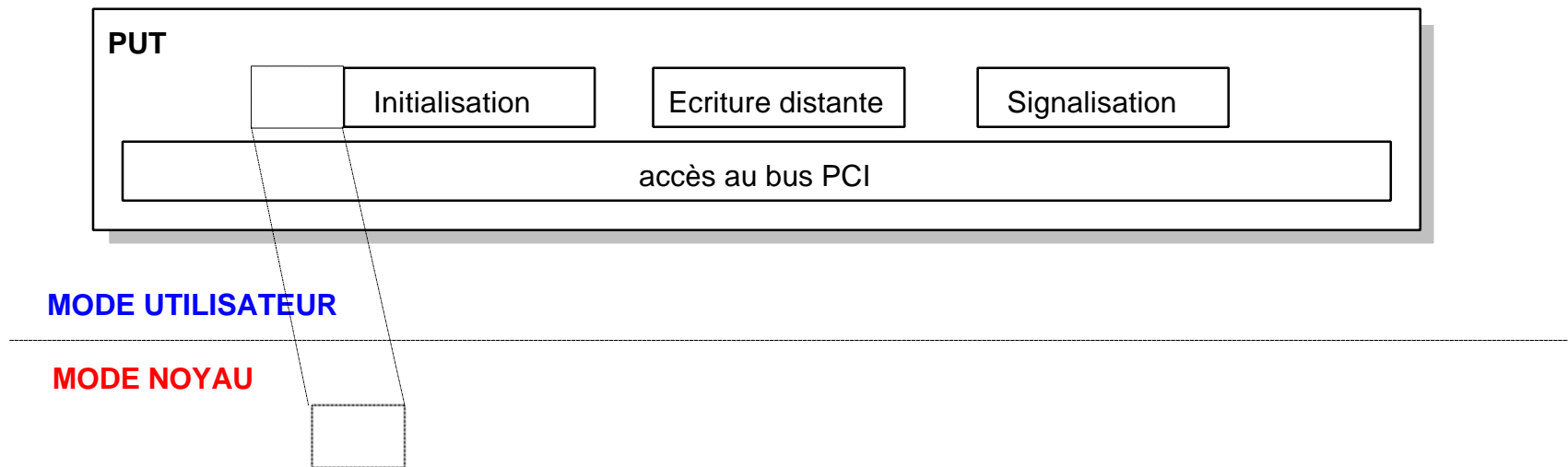
Changement de contexte



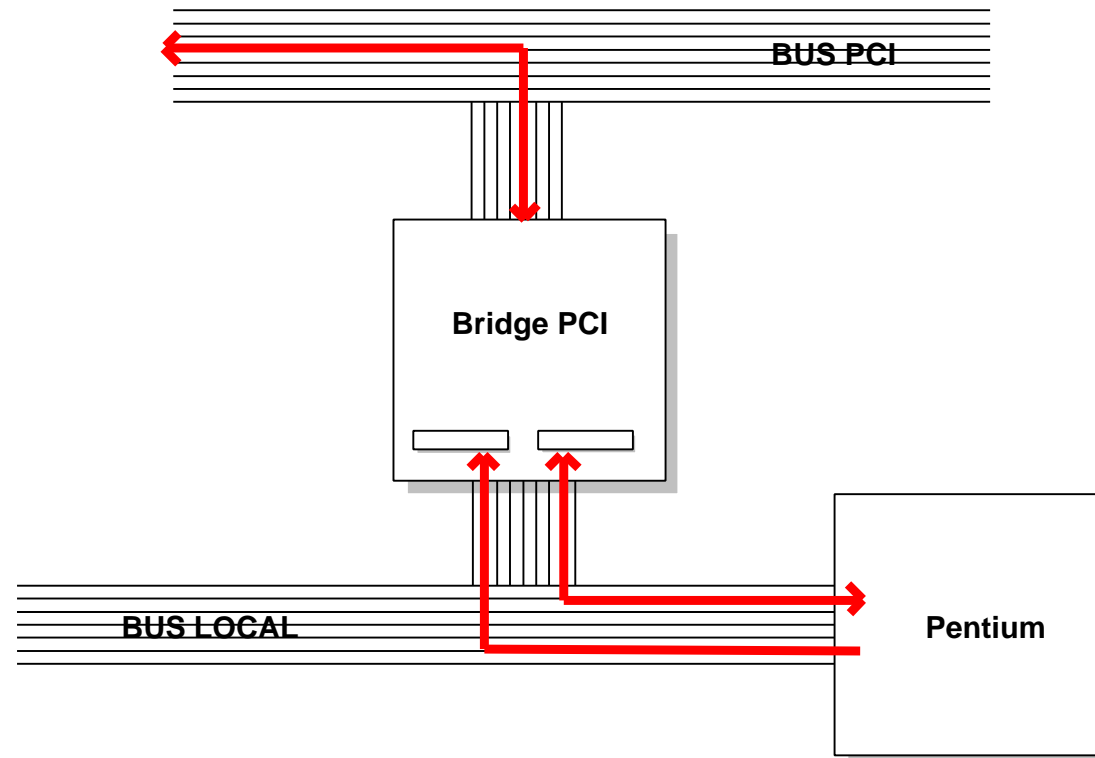
Les problèmes a résoudre

- **L'accès au bus PCI :** pas de fonctions C en mode utilisateur pour l'accès au bus PCI
- **Le partage des ressources :** verrouiller l'accès aux ressources communes en particulier de la carte , la LPE, la LMI
- **Les interruptions :** oblige au changement de contexte donc signalisation par polling sur les pointeurs de la LPE et de la LMI

Structure du PUT utilisateur

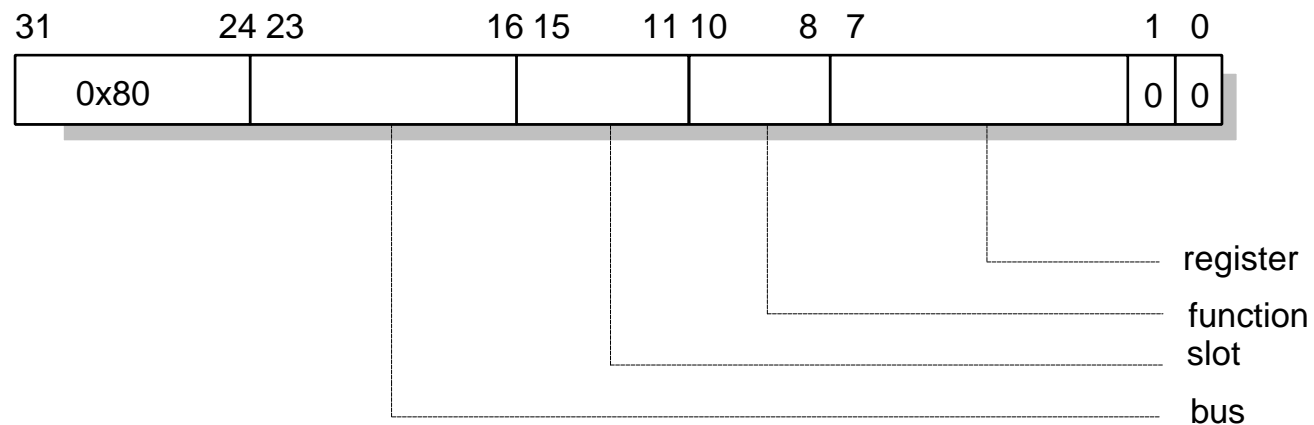


Les accès à la carte FastHSL



- on donne au processus le droit d'accéder aux bus d'entrées/sortie avec la fonction `iopl()`
- on fournit au bridge PCI l'adresse de la carte désirée
- on écrit ou on lit un registre du bridge qui exécute la même opération sur la carte

Structure de l'adresse



Registre : valeur du registre

Function : carte de type PCI donc function = 0

Bus : valeur comprise entre 0x00 et 0x3F

Slot : valeur comprise entre 0x00 et 0XFF

Les ressources à partager

- Les pointeurs images de ceux de la carte :

- lpe_new : incrémenté par PUT pour signaler un transfert
- lpe_current : actualisé par PCIDDC lorsque l'envoi est fait
- lmi_new : mis à jour par PCIDDC lorsque des données ont été réceptionnées
- lmi_current : actualisé par PUT

- Les listes :

- La LPE : liste pour l'émission des messages
- La LMI : liste pour la réception des messages

Le travail effectué sur PUT (5)

Les verrous

| Nom du verrou | Variables protégées | Fonction(s) utilisatrice(s) | opération lorsque le verrou est pris |
|----------------|------------------------|-----------------------------------|---|
| LPE_lock | la lpe | put_add_several_entries() | attente passive, la fonction boucle jusqu'a sa libération |
| LPE_flush_lock | pointeurs de la lpe | put_flush_lpe() | si le verrou est pris on ressort de la fonction sans code de retour |
| LMI_flush_lock | pointeur de la lmi | put_flush_lmi() | si le verrou est pris on ressort de la fonction sans code de retour |
| HSL_lock | l'accès a la carte hsl | hsl_conf_read() hsl_conf_write | si le verrou est pris on boucle jusqu'a sa libération |

La signalisation

- Un seul type de signalisation :

la scrutation, elle se fait par la lecture des registres de la carte

- deux fonctions :

`put_flush_lpe()` fait la notification de fin d'émission des données

`put_flush_lmi()` fait la notification de la réception des données

L'initialisation

- elle est de deux types :

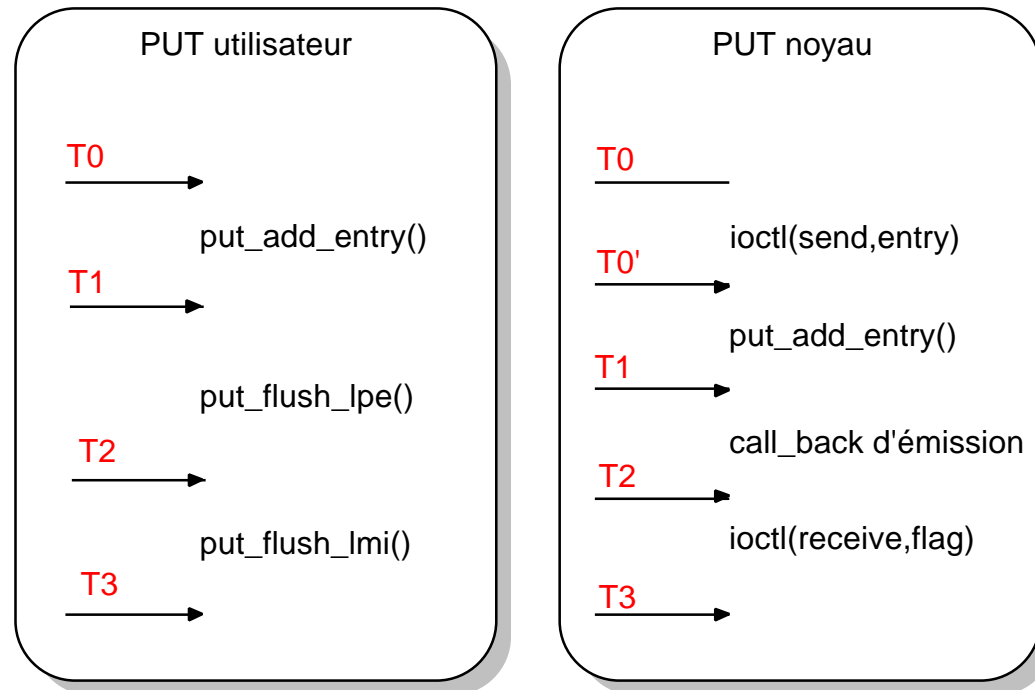
- initialisation matérielle

- initialisation logicielle

- applications utilisant PUT :

- une seule tranche de MI

La méthode utilisée pour les mesures



mesures faites sous 2 machines équipées de Pentium 166 MHz et 128 Moctets de RAM

Détails des mesures

| PUT utilisateur | | PUT noyau | | |
|-----------------|---------------------------|-----------|--------------------------------------|------|
| Fonction | temps microseconde | Fonction | temps microseconde | |
| T1 | put_add_several_entries() | 4,4 | put_add_entry() | 4,1 |
| T2 | put_flush_lpe() | 12 | call_back d'émission | 16,6 |
| T3 | put_flush_lmi() | 15,57 | apres l'ioctl() dans testputbench | 41,5 |

les 15,57 et 41,5 micros-secondes correspondent au temps d'un send suivi d'un receive le tout divisé par 2

Conclusion

Les apports personnels :

Les résultats sous MPI :

Les perspectives :

- portabilité**
- amélioration du polling**