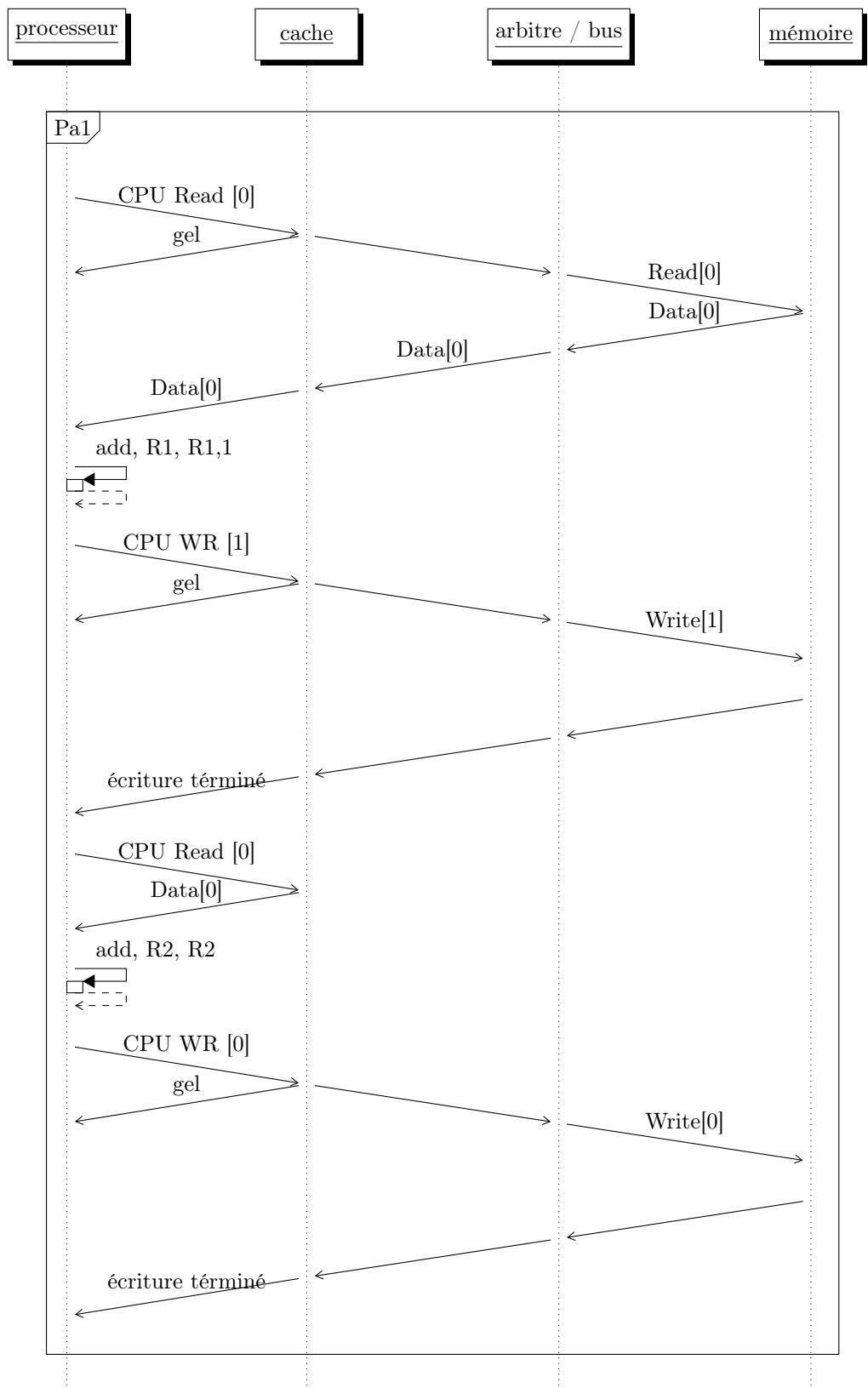


Rapport projet spécif

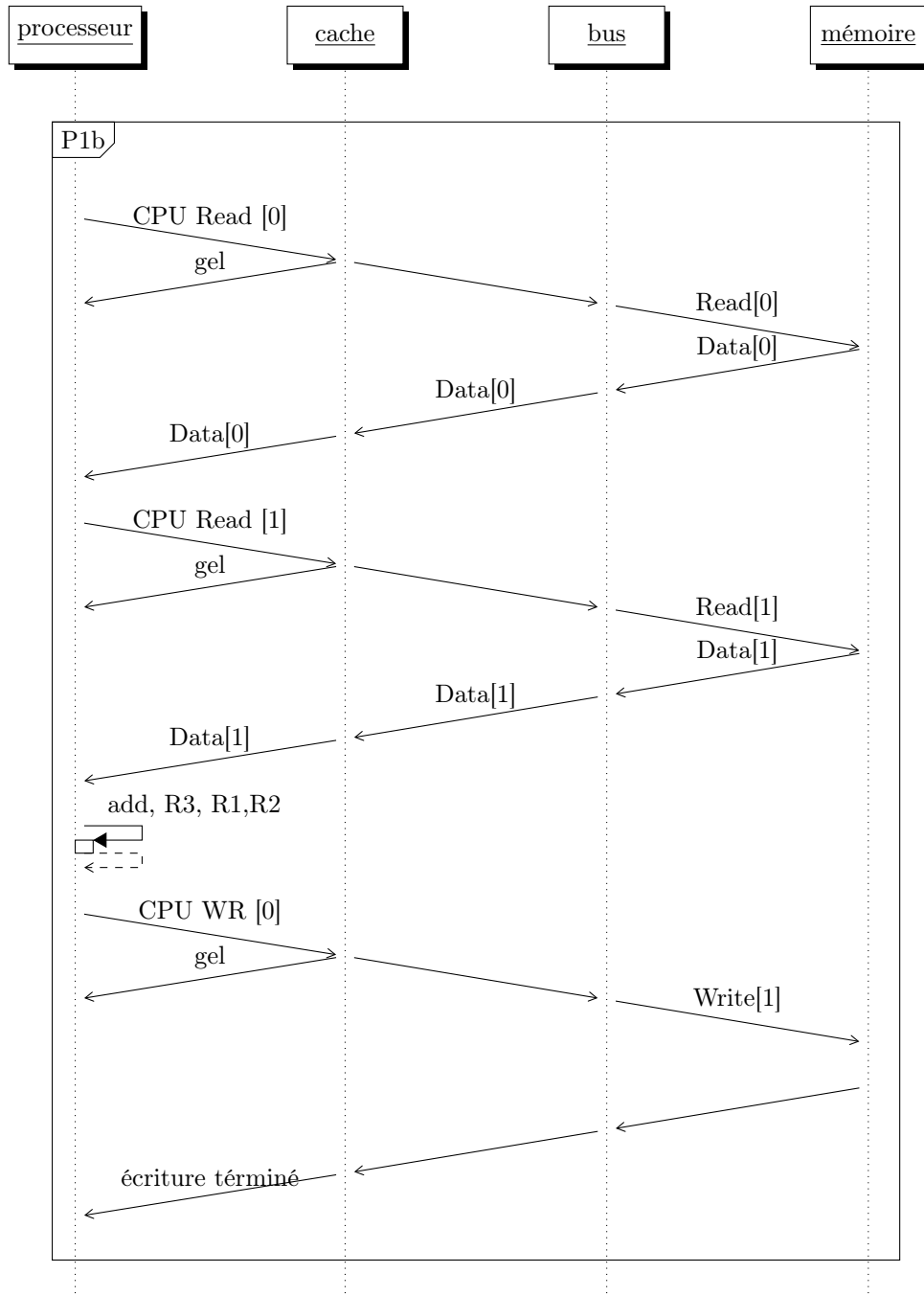
Ilyas TOUMLILT
Massine BITAM

18 avril 2015

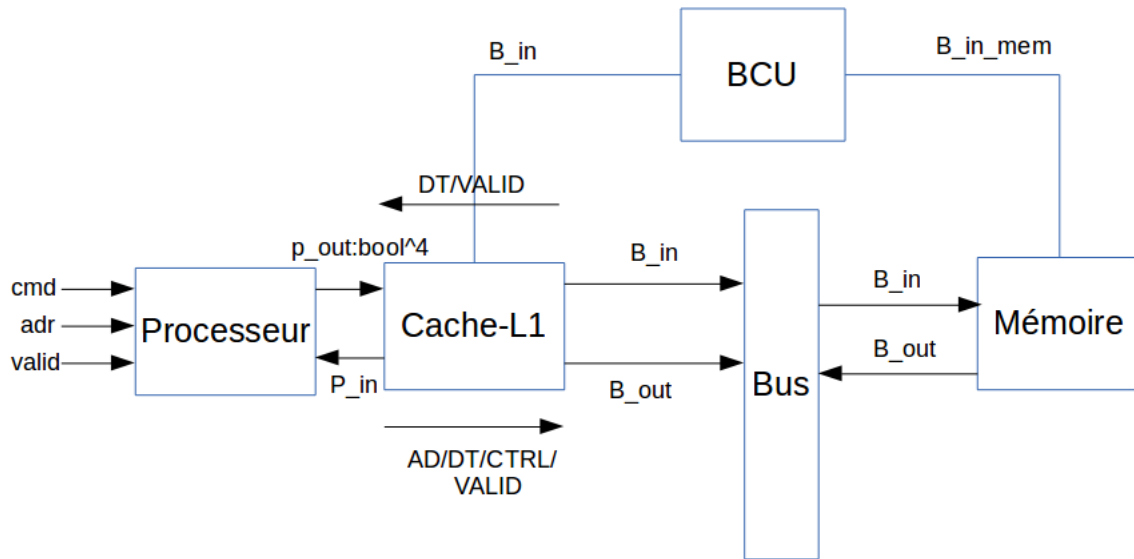
1 Cas 1 : Système mono processeur



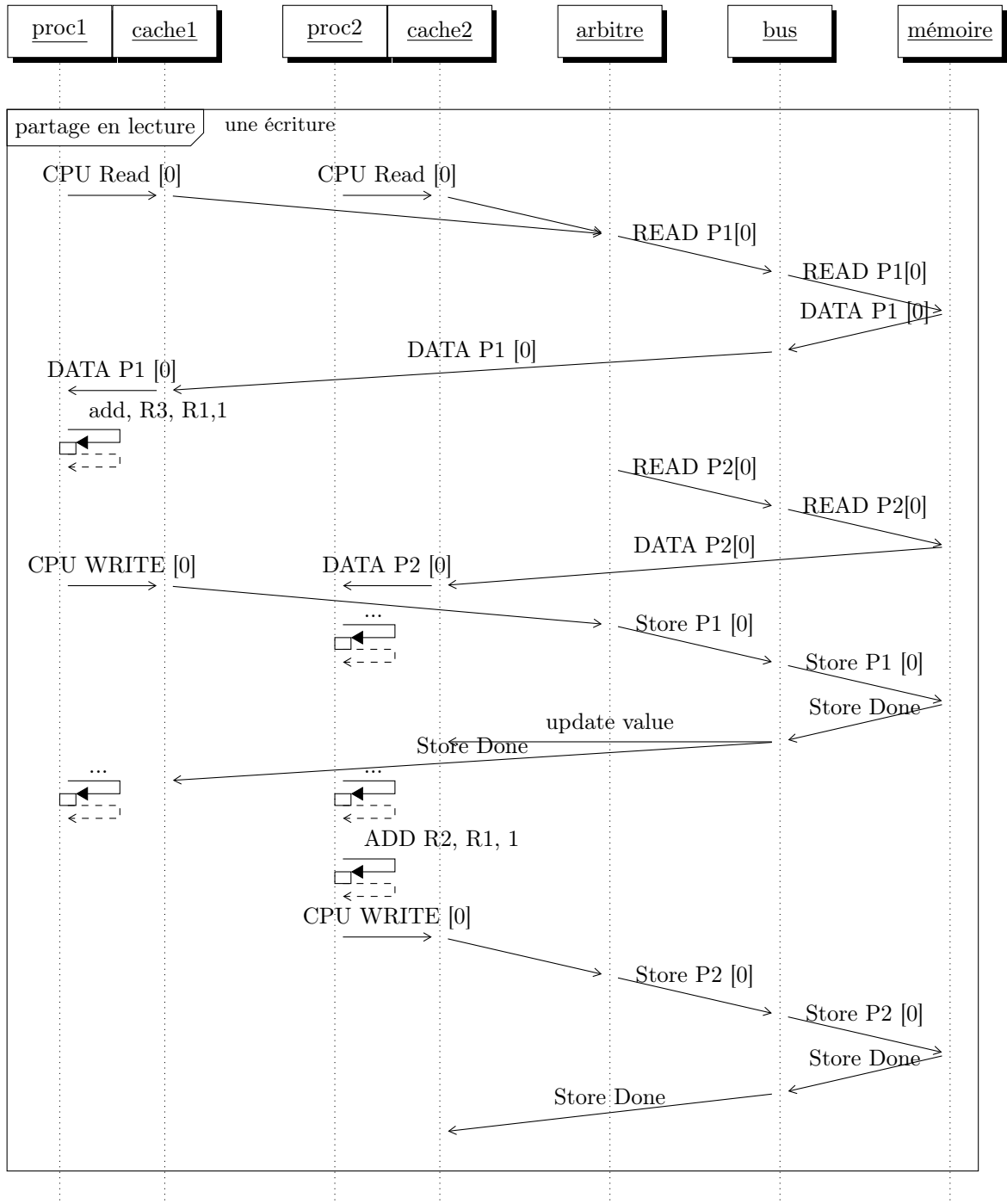
Après éviction du cache voici le diagramme de séquence du programme P1b.



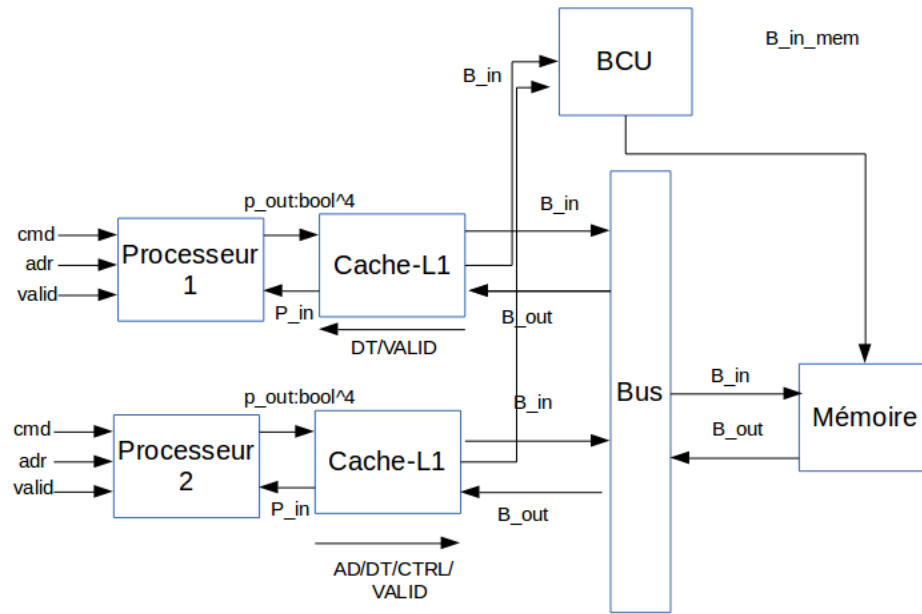
Voici l'architecture de notre Système mono-processeur ainsi que les connections des différent composant.



2 Cas 2 : Système multi processeur



Voici l'architecture de notre Système Multi-processeur ainsi que les connections des différent composant.



3 Realisation de la plateforme : description des composants

Les requetes :

De type bool4 dont les composants sont les suivants :

0 : AD, adresse du mot à transmettre.

1 : DT, valeur du mot transmis.

2 : CTRL, sens de transfert.

3 : VALID, si les données sont significatives.

NB : à partir de là, nous considérons que le terme requête fait référence à ce type de structure de données.

Le processeur :

Le processeur reçoit en entrée des requêtes du cache ou du cmd (main), le signal src gère la source de la requête. Il émet donc selon la source et le type des données une réponse vers un récepteur (cmd / cache) géré par le signal de sortie trg. Le proc peut être gelé en cas de MISS sur le cache (signal de sortie gel).

Comportements du proc :

—> à la réception d’une requête cmd : si c’est une requête significative (VALID), et que le proc n’est pas gelé, cette requête sera transmise au cache, qui pourra ou non geler le proc selon la disponibilité de la donnée en cache. sinon si ce n’est pas une requête significative, on redirige simplement l’entrée en sortie. si le proc est gelé il ne traite pas la commande. —> à la réception d’une requête du cache : cette requête peut ou être une requête de gel, qui ne fait que geler le proc en attendant la réponse. Ou une réponse à la requête, qui sera donc redirigée en sortie.

Le cache :

Le cache stocke le mot préalablement demandé en lecture par le processeur auquel il est associé, jusqu’à ce que ce mot soit mis à jour ou évincé. On a supposé que le cache ne peut contenir qu’un seul mot à la fois.

Le cache reçoit plusieurs signaux en entrée :

- arb_gnt : la décision de l’arbitre, et si oui ou non on a droit sur le bus.

- l1_in : requête du proc

- b_out : sortie du bus

Il retourne les signaux suivants en sortie :

- req : si le cache est demandeur de bus.

- gel : s’il le cache veut geler le proc.

- l1_out : request vers proc

- b_in : request vers bus

Le bus :

Le bus interconnecte la mémoire et les caches L1 des processeurs.

son architecture est multipléxée.

Le noeud reçoit la décision de l’arbitre et les sorties des caches et de la mémoire

Il retourne l’entrée du composant abr_gnt.

L’arbitre :

L’arbitre définit, lorsqu’il y a plusieurs requêtes simultanées vers la mémoire, celle qui a la priorité d’accès sur le bus.

La mémoire :

Composée de NWORDS mots de NBITS bits, adressable par mot.

La memoire est interfacée dans le bus et agit sur sollicitation de requêtes d’un des deux caches :

—> READ(ad) : demande de lecture du mot d’adresse ad.

- params : gnt_mem = true

- b_out[CTRL] = false
- b_out[AD] = ad du mot
- b_out[VALID] = true
- returns : b_in[AD] = b_out[AD]
- b_in[DT] = données demandée
- b_in[CTRL] = b_out[CTRL]
- b_in[VALID] = true
- > WRITE(ad, val) : demande d'écriture de la valeur val à l'adresse ad
- params : gnt_mem = true
- b_out[AD] = ad du mot
- b_out[DT] = valeur à écrire
- b_out[CTRL] = true
- b_out[VALID] = true
- returns : b_in[AD] = b_out[AD]
- b_in[DT] = b_out[DT]
- b_in[CTRL] = b_out[CTRL]
- b_in[VALID] = b_out[VALID]
- > Dans tous les autres cas (cycles d'inactivité on va dire)
- returns : b_in[AD] = b_out[AD]
- b_in[DT] = b_out[DT]
- b_in[CTRL] = b_out[CTRL]
- b_in[VALID] = false