

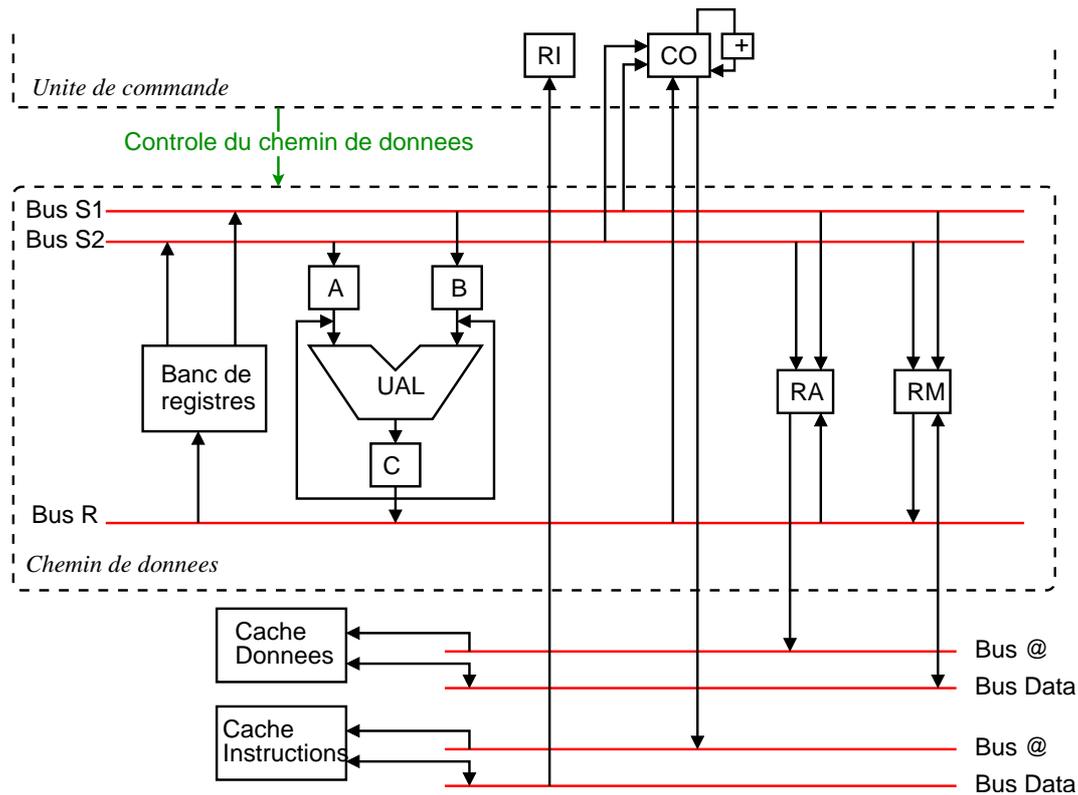
TD 5 – Architecture des Ordinateurs

Etude du pipeline d'un processeur fictif

Licence Informatique 3^{ème} Année – UPPA

1 Description d'un processeur fictif

1.1 Chemin de données



La figure ci-dessus représente le chemin de données pour un processeur fictif. Il se base sur une architecture 3 bus. Les registres RI, RA, RM et CO sont identiques à ceux vus en cours. Le banc de registre contient 16 registres généraux (nommés R0, R1, ... R15). Il supporte l'exécution de 3 opérations simultanément : 1 opération d'écriture et 2 opérations de lecture (un registre écrit ne peut par contre pas être lu simultanément). Les données, adresses et instructions sont toutes stockées sur 32 bits.

L'UAL réalise les opérations logiques et de calcul sur des entiers. Les opérations simples (addition, soustraction, décalage, ...) sont réalisées en un cycle d'horloge. La multiplication, plus complexe, s'exécute en 3 cycles d'horloge. L'UAL exécute les opérations avec les valeurs stockées dans les registres tampons A et B. À la fin du cycle d'horloge (pour les opérations simples), le résultat est disponible dans le registre tampon C.

1.2 Jeu d'instruction du processeur

Le processeur fonctionne en mode d'accès mémoire chargement-rangement. Toutes les opérations arithmétiques et logiques n'accèdent qu'à des registres généraux pour spécifier leurs opérandes.

Voici les principales instructions du jeu d'instruction de ce processeur :

LOAD Ra, Rb/valeur écrit dans le registre Ra le contenu à l'adresse mémoire précisée directement ou contenue dans le registre Rb

STORE Ra, Rb/valeur écrit la valeur du registre Ra à l'adresse mémoire précisée directement ou dans le registre Rb

ADD Rs, Ra, Rb fait l'addition de Ra et Rb et place le résultat dans Rs

MUL Rs, Ra, Rb fait la multiplication de Ra et Rb et place le résultat dans Rs

INC/DEC Ra, x/Rb incrémente/décrémente Ra d'une valeur x ou de la valeur contenue dans Rb

JMP add/Ra saut à une adresse du programme précisée directement ou dans le registre Ra

MOV Ra, Rb/valeur écrit dans Ra le contenu de Rb ou la valeur précisée

JE/JG/JL/JGE/JLE Ra, Rb, add/Rc saut à une adresse – précisée directement ou dans le registre Rb – si Ra est égal/plus grand/plus petit/plus grand ou égal/plus petit ou égal que/à Rb

1.3 Pipeline d'exécution

Le séquençement de l'exécution des opérations comporte 5 phases distinctes, chacune durant un cycle d'horloge, exécutées dans l'ordre suivant :

1. **LE** Lecture de l'instruction via un accès en mémoire à l'adresse contenue dans CO. RI contient l'instruction à exécuter.
2. **DE** Décodage de l'instruction contenue dans RI.
3. **CH** Chargement des registres A,B de l'UAL ou des registres RA et/ou RM pour accès mémoire, à partir des registres généraux.
4. **EX** Exécution de l'instruction dans l'UAL ou accès en mémoire.
5. **ENR** Enregistrement du résultat de l'opération (si résultat) : copie du registre C de l'UAL ou du registre RM dans un registre général ou écriture du registre CO pour une instruction de saut.

Un saut consiste à modifier le contenu du registre CO pour préciser l'adresse de la prochaine branche du programme à suivre. Ce registre est par défaut automatiquement incrémenté de 4 à chaque cycle d'horloge pour pointer sur l'adresse de la prochaine instruction à exécuter.

Pour une opération d'accès en mémoire, le registre RA est écrit avec la valeur de l'adresse mémoire à accéder. Le registre RM contient la valeur lue pour une opération de lecture et à écrire pour une opération d'écriture. On supposera que le temps de lecture/écriture en mémoire prend un seul cycle d'horloge (via la mémoire cache).

2 Questions

1. Traduire en assembleur le programme du TD 4 sur la mémoire cache. La première instruction du programme est placée à l'adresse 1000 en mémoire.
2. Vérifier que les estimations d'accès mémoire faites pendant le TD4 étaient valides et que le taux de succès du cache était correct.
3. Montrer quelques exemples d'aléas structurels, de données et de contrôle.
4. Optimiser le programme (version assembleur) pour limiter ces aléas ainsi que le nombre d'accès en mémoire. Proposer des modifications matérielles simples pour limiter les aléas.