

## TD-4 : Pipelines scalaires et prédiction de branchement

### Exercice 1

On considère une architecture RISC possédant 4 types d'instructions :

- A = arithmétiques et logiques : registre- registre
- C = chargement : lecture mémoire
- R = rangement : écriture mémoire
- B = branchement. Les branchements conditionnels se font sur code condition

Cette architecture a deux implantations, l'une avec le pipeline P1 dont les phases sont décrites table I et l'autre avec le pipeline P2 qui est décrit table II. Dans les deux cas, les caches instructions et données sont séparés.

LI	Lecture de l'instruction	LI1	Début lecture de l'instruction
LR	Pour toutes les instructions, décodage de l'instruction et lecture des opérandes registre. Pour les types C et R, calcul de l'adresse mémoire Pour le type B, calcul de l'adresse de branchement	LI2	Fin lecture de l'instruction
EX	Pour le type A, opération UAL. Pour le type R, lecture du 3 <sup>ème</sup> opérande. Pour les types C et R, accès mémoire	LR	Décodage de l'instruction et lecture des opérandes registre
RR	Rangement du résultat arithmétique (A) ou de la lecture (C) dans le banc de registres.	EX	Pour le type A, opération UAL. Pour le type R, lecture du 3 <sup>ème</sup> opérande. Pour les types C et R, calcul de l'adresse mémoire. Pour le type B, calcul de l'adresse de branchement
		M1	Pour les types C et R, début accès mémoire
		M2	Pour les types C et R, fin accès mémoire
		RR	Rangement du résultat arithmétique (A) ou de la lecture (C) dans le banc de registres.
<i>Table 1 : PIPELINE 1</i>		<i>Table 2 : PIPELINE 2</i>	

- 1) Définir l'ensemble des dépendances qui peuvent exister entre une instruction qui génère une donnée ou une adresse et une instruction qui utilise cette donnée ou cette adresse.
- 2) Pour toutes les dépendances définies dans la question précédente, on introduit tous les mécanismes d'anticipation nécessaires (bypass) pour supprimer ou réduire les suspensions. Définir dans chaque cas le « bypass » nécessaire et en déduire le nombre de cycles de suspension (avec 0 quand il n'y a pas de suspension)..
- 3) Quelle est la taille du délai de branchement pour P1 et pour P2 ?
- 4) Quelles sont les ressources matérielles nécessaires aux deux pipelines ?

### Exercice 2

On considère une architecture à cache commun instructions et données

- 1) Un pipeline à cinq étages (LI, LR, EX, MEM, RR) est-il utile ?
- 2) Proposer un pipeline d'exécution, qui permette la lecture du registre Rd pendant la phase EX des instructions de rangement, en supposant que le banc de registre n'admet que 2 accès simultanés en lecture.
- 3) Donner la liste des aléas de données de ce pipeline avec anticipation.

### Exercice 3 : prédicteurs de branchement

Soit le programme C suivant :



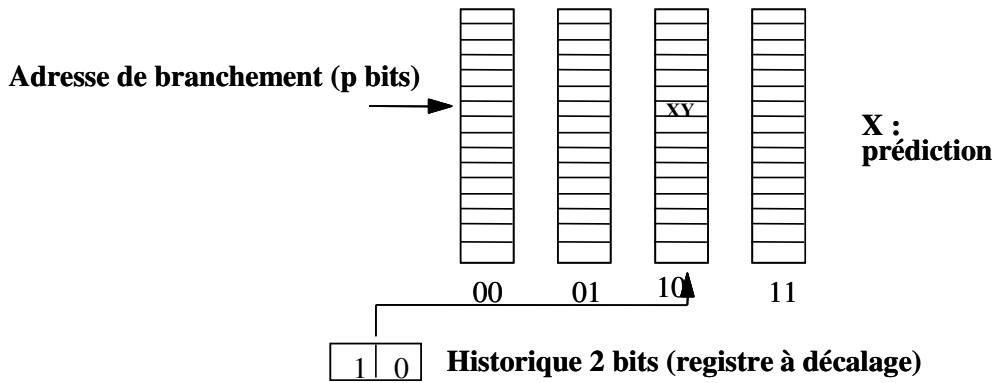


Figure 1 : prédicteur (2,2)

Remplir le tableau ci-dessous correspondant à l'exécution du programme :

R1						
R2						
Branchement						
Corrélation						
Compteur						
Comportement branchement						
Prédiction						

R1						
R2						
Branchement						
Corrélation						
Compteur						
Comportement branchement						
Prédiction						

### Exercice 5 : comparaison de prédicteurs

On utilise trois prédicteurs de branchement différents

- le prédicteur bimodal (figure 1) utilise 14 bits d'adresse pour indexer des compteurs 2 bits
- le prédicteur global (figure 2) utilise un registre global d'historique des 10 derniers branchements pour indexer des compteurs 2 bits
- le prédicteur local (figure 3) utilise 14 bits d'adresse pour indexer  $2^{14}$  historiques locaux des 10 derniers comportements d'un branchement. Chaque comportement indexe des compteurs 2 bits.

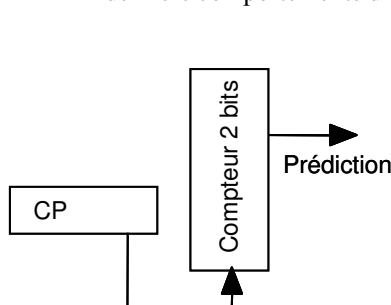


Figure 1 : prédicteur bimodal

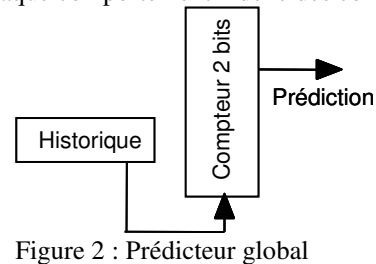


Figure 2 : Prédicteur global

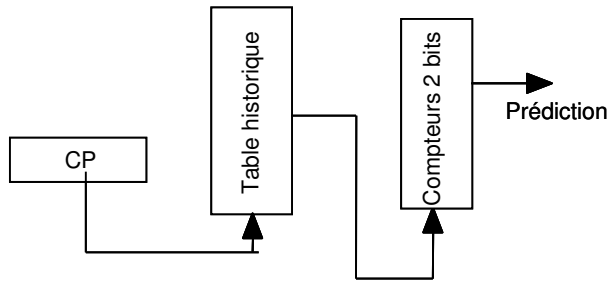


Figure 3 : Prédicteur local

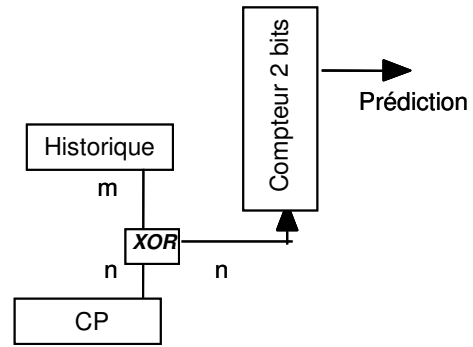


Figure 4 : Prédicteur gshare

- 1) Déterminer le nombre de bits nécessaires pour implanter chaque prédicteur

Soit la fonction

```
void more (int &x, int &y , int a, int &b, int &c){  
    for (int = 0 ; i<100 ; i++) x=x^i;  
        if (a<10) b++; // branchement B, jamais pris;  
    for (int = 0 ; i<100 ; i++) y=y^i;  
        if (a>=10) c++; // branchement C, toujours pris}
```

- 2) Quelle est la précision sur la prédiction des branchements B et C pour les trois prédicteurs
- 3) Pourquoi le prédicteur gshare (figure 4) résout-il le problème du prédicteur global ?