

Conception d'un processeur *pipeline*

Architecture avancée, 2018-2019

Cet exercice est largement inspiré de l'exercice A5, page A-82 du Hennessy/Patterson 3ème édition.

Ces questions ont pour but de concevoir une machine dont l'un des opérandes peut être une case mémoire, ce que l'on appelle une machine registre/mémoire. La machine possède 2 formats d'instruction : un format registre/registre et un format registre mémoire. Il n'y a par contre qu'un seul mode d'adressage, le mode indirect registre plus déplacement identique à celui du MIPS que nous avons vu jusqu'à présent.

Les instructions utilisant l'ALU (opérations ADD, SUB, AND, OR ...) s'utilisent comme suit :

```
OPR Rdst, Rsrc1, Rsrc2
OPM Rdst, Rsrc1, CST(Radr)
```

On peut par ailleurs effectuer des accès mémoires sans calculs avec les instructions suivantes :

```
LW Rdst, CST(Radr)
SW Rsrc1, CST(Radr)
```

Les branchements permettent de faire tous les types de comparaison entre deux registres (égalité, supériorité stricte, ...) et sont relatifs au PC.

```
Bcc Rsrc1, Rsrc2, label
```

La machine est pipelinée en 6 étages (dans l'ordre IF, RF, ALU1, MEM, ALU2 et WB) de sorte qu'une instruction est commencée à chaque cycle. L'étage IF procède au chargement de l'instruction via la mémoire d'instruction. L'étage RF (*Registers Fetch*) réalise à la fois le décodage de l'instruction et la récupération des valeurs des registres. L'étage ALU1 est utilisé pour le calcul de l'adresse effective des accès mémoire et des branchements. L'étage MEM permet d'accéder à la mémoire de données. L'étage ALU2 est utilisé pour les opérations de l'ALU et pour les comparaisons utilisées par les branchements.

On fait de plus l'hypothèse que si le même registre est lu *et* écrit au même cycle, alors la lecture retourne la valeur qu'aura l'écriture à la fin du cycle.

Question 0 – Donnez le schéma bloc d'un banc de registres ayant le comportement décrit ci-dessus en supposant que l'on dispose d'un banc de registres « normal », c'est-à-dire qui peut-être écrit et lu dans le même cycle, mais pour lequel si le même registre est la cible d'une écriture et d'une lecture dans le même cycle, on récupérera en lecture la valeur précédent l'écriture. Le banc de registre « normal » ne sera pas détaillé, on se basera uniquement sur ses entrées/sorties pour bâtir la nouvelle version.

Question 1 – Dessinez sommairement le pipeline en positionnant étage par étage les ressources

principales de la machine (mémoires, banc, ALU...). Précisez le nombre maximum d'opérateurs pouvant réaliser une addition dont a besoin cette architecture. Précisez aussi le nombre d'accès et le type (lecture/écriture) de chaque accès nécessaire au banc de registres et aux mémoires.

Pour lister de manière exhaustive les aléas de données, il faut détecter les cas de dépendances lecture après écriture (RaW, Read after Write). Pour le processeur décrit, les instructions accèdent à :

- 1 registre en écriture (Rdst) et 2 en lecture (Rsrc1, Rsrc2) pour les instructions OPR
- 1 registre en écriture (Rdst) et 2 en lecture (Radr, Rsrc1) pour les instructions OPM
- 1 registre en écriture (Rdst) et 1 en lecture (Radr) pour l'instruction LW
- 2 en lecture (Radr, Rsrc1) pour l'instruction SW
- 2 en lecture (Rsrc1, Rsrc2) pour les instructions Bcc

Avec 3 types d'accès en écriture et 9 d'accès en lecture, on peut distinguer 27 cas de dépendances RaW. Pour chacun de ces cas et selon la distance en nombre d'instructions séparant l'instruction impliquée dans l'écriture de celle impliquée dans une lecture, on peut dire rapidement s'il y a un aléa et s'il nécessite une attente.

Question 2 – Chaque situation correspond à une case du tableau fourni. Remplissez-le avec au choix : OK, court-circuit (bypass), bulles (stall) et leur nombre.

dépendance RaW				Distance			
Inst. W	Reg W	Inst. R	Reg. R	1	2	3	4
OPR	Rdst	OPR	Rsrc1				
		OPR	Rsrc2				
		OPM	Rsrc1				
		OPM	Radr				
		LW	Radr				
		SW	Radr				
		SW	Rsrc1				
		Bcc	Rsrc1				
		Bcc	Rsrc2				
OPM	Rdst	OPR	Rsrc1				
		OPR	Rsrc2				
		OPM	Rsrc1				
		OPM	Radr				
		LW	Radr				
		SW	Radr				
		SW	Rsrc1				
		Bcc	Rsrc1				
		Bcc	Rsrc2				
LW	Rdst	OPR	Rsrc1				
		OPR	Rsrc2				
		OPM	Rsrc1				
		OPM	Radr				
		LW	Radr				
		SW	Radr				
		SW	Rsrc1				
		Bcc	Rsrc1				
		Bcc	Rsrc2				

Question 3 – Ajoutez sur votre dessin de pipeline le matériel permettant de résoudre les courts-circuits. On ne demande pas ici la détection des aléas.

Question 4 – Indiquez pour chaque court-circuit l'équation de détection associée, ainsi que celle pour

la détection des cycles d'attente à insérer.

Question 5 – Pour finir, donnez la liste des aléas liés au contrôle (i.e. le nombre d'instructions suivant un branchement, qui peuvent poser un problème).