

Règles d'exécution des instructions dans le superscalaire Mips

1. Les instructions sont exécutées dans l'ordre du programme : une instruction ne peut exécuter un étage quelconque du pipeline avant une instruction qui la précède.
2. Un superscalaire à deux pipelines possède un buffer d'instructions de 4 places. Les places vides du buffer d'instructions sont remplies avec des Nop* (Il faut faire la différence entre une instruction Nop faisant partie du programme - c'est-à-dire lue à partir de la mémoire - et les Nop* que le processeur insère dans le pipeline - schématisés par les bulles - pour remplir les trous dus aux Stall).
3. Dans le cycle IFC, les instructions sont extraites de la mémoire deux par deux pour un superscalaire à deux pipelines. L'adresse de ce couple d'instructions doit être alignée - c'est à dire multiple de 8. On enregistre les instructions dans le buffer chaque fois qu'au moins deux places sont disponibles - ou vont être disponibles à la fin de ce cycle - dans le buffer. Si le processeur tente de lire des instructions à une adresse non alignée (ce cas peut se produire après un branchement), la mémoire aligne l'adresse (en ignorant les 3 bits de poids faible). Elle envoie au processeur un couple d'instructions lu à l'adresse alignée et le processeur ne conserve que la seconde instruction - c'est-à-dire que seule cette seconde instruction est enregistrée dans le buffer.
4. Le contenu du buffer est invalidé lorsque l'on exécute un branchement pris. D'une manière générale, le contenu du buffer est invalidé lorsque l'exécution du branchement contredit la prédiction. Comme dans notre superscalaire le buffer est alimenté à partir des adresses séquentielles, un branchement qui réussit contredit la prédiction.
5. Le superscalaire est compatible avec le "Mips scalaire" - c'est-à-dire qu'il y a un delayed slot après un branchement. Donc, lorsque l'on exécute une instruction de branchement qui réussit, si le delayed slot reste dans le buffer, il ne faut pas l'invalider.
6. À chaque cycle, dans le cycle DEC, les deux premières instructions du buffer sont décodées. Si les conditions d'exécution de ces instructions sont réunies, elles sont autorisées à passer au cycle **EXE**. On dit qu'elles sont lancées. Pour lancer une instruction on autorise son écriture dans le registre **I_RD** (**I_RD0** pour le pipeline 0 et **I_RD1** pour le pipeline 1).
7. Dans le cycle DEC, la seconde instruction ne peut être lancée (autorisée à passer au cycle **EXE**) que si elle est indépendante de la première instruction.
8. Une instruction de branchement ne peut être lancée que si son delayed slot est présent dans le buffer.
9. Une instruction avance toujours dans le pipeline dans lequel elle a été lancée. Elle ne peut pas changer de pipeline.
10. Une instruction est bloquée dans le pipeline si l'instruction qui se trouve dans l'étage suivant est bloquée (comme dans le Mips scalaire).
11. Une instruction est bloquée dans le pipeline si elle n'a pas les ressources nécessaires à son exécution (dépendance de données comme dans le Mips scalaire).
12. Dans le cycle **MEM**, on ne peut effectuer qu'un seul accès mémoire au maximum (Load ou Store). Si deux instructions d'accès mémoire sont présentes dans le cycle MEM, la première instruction (dans l'ordre du programme) est exécutée, la seconde est bloquée (Stall).
13. Dans le cycle **WBK**, on peut sauvegarder deux résultats à chaque cycle, sauf si les deux instructions présentes dans le cycle **WBK** écrivent toutes les deux dans le même registre. Dans ce cas, la première instruction est exécutée, la seconde est bloquée.