

ARO Labo Pipeline Questions

1.1

1. Dans le circuit mult_2, les offsets sont incrémentés de 1 au lieu d'être incrémenté de 2 dans le circuit non-pipeliné, pourquoi?
 - Comme le calcul est fait dans le circuit decode alors on a besoin d'une incrémentation de 1.
2. Dans le circuit fetch_decode_reg, le signal LR_adr_o vient d'un registre et est connecté au bloc decode au lieu du bloc bank_register, pourquoi?
 - On doit stocker la valeur précédente du fetch pour détecter l'aléa.
3. Dans le circuit decode, le signal adr_reg_d_s va dans le circuit decode_execute_reg dans un registre alors que les signaux adr_reg_n_s, adr_reg_m_s et adr_reg_mem_s ne sont pas stockés dans des registres, pourquoi ?
 - Dans le decode on a déjà lu la valeur du registre N, M et MEM donc on en n'a plus besoin. On doit juste garder le registre D jusqu'à la fin pour la lire.
4. Dans le circuit decode_execute_reg, les signaux des bus de contrôle sont connectés aux registres avec une porte MUX contrairement aux autres signaux, pourquoi?
 - Si le decode n'est pas activé, la valeur qui est restée va devenir parasite pour la suite du calcul. On remet à 0 les registres pour nettoyer le pipeline.
5. Si on voulait ajouter le multiplieur 5x3 pipeliné du laboratoire préparatoire, quelles seraient les conséquences sur le pipeline du processeur ? Comment ça pourrait être fait ?
 - La multiplication est une série d'additions successives. Cela signifie que le bloc Execute nécessitera plus de temps pour effectuer cette opération. Il faut s'assurer que l'unité ALU dispose du temps nécessaire pour réaliser ce calcul.

1.2

1. Est-ce que le programme s'exécute correctement? Est-ce que les registres et la mémoire prennent les bonnes valeurs?
 - Le programme est exécuté correctement et la mémoire prend bien les bonnes valeurs.

2. Combien de cycles sont nécessaires pour exécuter ce programme?

- On a besoin de 5 cycles.

1.3

1. Quelles dépendances posent des problèmes d'aléas?

- Les dépendances posent un problème lorsqu'on a un WAR, WAW et RAW.

2. Combien de cycles sont nécessaires pour résoudre un aléa de donnée?

- Il faut 5 cycles pour résoudre un aléa de donnée.

3. Quel est l'IPC sur le programme fourni ? Le throughput si la clock vaut 4KHz? La latence?

- L'IPC = 0.5
- le throughput = 2Hz
- la latence est de 1.25 ms

2.1

1. Comment savoir si une instruction est dépendante d'une instruction qui est pour le moment dans le stage EXECUTE? dans le stage MEMORY_ACCESS? Dans le stage WRITE_BACK?

- Comme on stock dans la banque de registre les valeurs précédentes, on compare ces valeurs avec le registre de l'instruction actuelle. En premier on a l'exécution puis memory access et finalement le write back.

2. Est-ce que ça pose un problème si une instruction dépend du résultat d'une instruction qui est au stage WRITE_BACK?

- Oui c'est un problème.

3. Quelles informations doivent être mémorisées pour chaque instruction?

- On a besoin de adr_reg_n_s, adr_reg_m_s, adr_reg_d_s

4. Quelles informations permettent de savoir si le registre D est utilisé?

- C'est dans adr_reg_d_s

2.2

1. Quelles informations permettent de savoir si le registre N, M ou mem sont utilisés?

- On doit regarder l'instruction et la doc.

1. Quelles informations permettent de savoir si le registre D est utilisé?

- Lorsque l'instruction écrit dans la mémoire ou dans un registre.

2. Une détection d'aléa de donnée va influencer quel(s) enable(s)? A quel moment ? Pourquoi?

- La détection influence l'enable de fetch et decode. Ensuite influence l'execute, memory access et write back après chaque coup d'horloge.