

Questions 3 Pipeline Forwarding Christopher Ristic et Julien Holzer

3.1 Circuit data_hazard

1. A quoi sert le signal `sel_mem_i`?

Il sert à savoir si l'on manipule la mémoire (avec un LDR ou un STR).

2. Est-il possible/utile de faire un data forwarding depuis le stage `WRITE_BACK` ? (l'écriture dans le registre dans la banque de registres). Comment pourrait-il être ajouté au circuit?

Oui cela serait possible mais ce n'est pas utile puisque la valeur aura fini d'être écrite et donc sera disponible après le write back (donc il n'est pas nécessaire de la mettre dans un registre supplémentaire). Il pourrait être implémenté tout comme le forwarding depuis l'exécute ou le memory access.

3. Quelles sont les conditions pour que le forwarding puisse avoir lieu? Quelles sont les conditions pour que le forwarding soit utile?

Le forwarding a lieu lorsque `sel_op_1_forward_s` et `sel_op_2_forward_s` sont à actifs, c'est-à-dire lorsqu'il y a un aléa et qu'une instruction a besoin d'une valeur qui n'est pas encore écrite

Il est utile lorsque qu'une instruction a besoin d'une valeur pas encore écrite (cela permet d'avoir la valeur à l'avance sans avoir besoin d'attendre qu'elle soit écrite)

4. Quelles sont les conséquences du forwarding sur la gestion des aléas de données?

Grace au forwarding il n'y a plus besoin de faire d'arrêt du pipeline (à l'exception du cas où un LDR a une dépendance RAW avec l'instruction immédiatement après auquel cas il faut faire un arrêt d'un cycle).

3.2 Circuit execute

1. Pourquoi doit-on faire ça?

Sinon l'opérand_1_s et 2 prendrait les mauvaises valeurs (il faut donc donner les bonnes valeurs en fonction de si elles doivent provenir de execute ou de memory access ou autre dépendant d'où le forwarding est effectué)

2. Pourquoi doit-on faire ça pour le signal `reg_mem_data_s`?

Pour la même raison que pour les opérandes 1 et 2 mais pour la partie memory access

3. Que devrait-on faire si on avait un data forwarding venant du `WRITE_BACK`?

Durant la phase WB, les résultats d'une opération sont enregistrés en mémoire ou dans un registre. Si une instruction subséquente dans le pipeline (qui pourrait être en phase Decode/ID ou Execute/EX) nécessite cette valeur, le data forwarding permet d'envoyer directement la valeur à cette instruction, évitant ainsi un délai d'attente pour que la valeur soit d'abord écrite puis relue.

3.3 Test: pipeline forwarding

1. Est-ce que votre processeur fonctionne correctement? Est-ce que les timings sont respectés?

Est-ce que les registres contiennent les bonnes valeurs si on regarde étape par étape l'exécution des instructions?

Oui il fonctionne correctement et oui les timings sont respecté et oui les registres contiennent les bonne valeurs.

2. Quel est l'IPC de votre programme? et le throughput si on considère une clock à 4KHz?

$IPC = \text{nbr instruction} / \text{nbr cycle}$

$IPC = 8 / 12 = 0.666$

$\text{Throughput} = IPC * \text{Fréquence} = 0.666 * 4Khz = 2666.666$

3. Avez-vous d'autres idées d'optimisation de ce processeur?

Ajouter des étage de pipelines (augmenter le parallélisme, l'accélération)