Pipeline P2

Alexandre Delétraz, Michaël Cheneval

1.1

Questions

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 ?
2. Est-ce que ça pose un problème si une instruction dépend du résultat d'une instruction qui est au stage WRITE_BACK ?
3. Quels informations doivent être mémorisées pour chaque instruction ?
4. Quelles informations permettent de savoir si le registre D est utilisé ?

1. Grâce à des registres intermédiaires qui se trouvent dans le decode.

   Dans execute :
   On a, lors de la sélection de l'opérande, un étage de MUX supplémentaire permettant de définir si on utilise une valeur venant d'un forwarding ou non. (0 ⇒ valeur non forwardée, 1 ⇒ valeur de l'execute précédent / registre E, 2 ⇒ valeur du memory access précédent / registre F1, 3 ⇒ 0)

   Changements dans le registre reg_mem_read_data_o. L'entrée des données se fait via un MUX dont la sortie varie en fonction d'une entrée sel_op_mem_forward_s (valeur forwardée depuis le memory access d'avant / registre F1).

2. Non
3. Il faut mémoriser, les adresses des registres de destination, le fait que la banque de registre a été activée en écriture et le flag qui indique si l'instruction est une instruction mémoire.
4. Si tous les signaux "compare" sont à 1

1.2

Questions

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

1. Les “enable” de chaque registre (reg_n_en_i, reg_m_en_i, reg_mem_en_i)

Questions

1. Quelles informations permettent de savoir si le registre D est utilisé ?
2. Une détection d’aléa de donnée va influencer quel(s) enable(s) ? A quel moment ? Pourquoi ?

1. Je ne sais pas.
2. Dans l’ordre: les enable du fetch, du decode, de l’execute, du memory access et de la banque de registres.

1.3

Questions

1. Est-ce que les valeurs dans les registres sont mises à jour correctement et au bon moment ?
2. Pourquoi l’instruction *BL* génère un aléa de contrôle et un aléa de donnée ?
3. Combien de cycles sont nécessaires pour résoudre les aléas de l’instruction *BL* ?
4. Quel est l’IPC pour votre programme ?

1. Oui
2. Car elle fait un saut et elle demande à écrire dans un registre.
3. 3
4. $IPC = \frac{nbrInstr}{nbrClk} = \frac{9}{19} = 0.4736$

Forwarding

2.1

Questions

1. A quoi sert le signal sel_mem_i ?
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 ?
3. Quelles sont les conditions pour que le forwarding puisse avoir lieu ? Quelles sont les conditions pour que le forwarding soit utile ?
4. Quelles sont les conséquences du forwarding sur la gestion des aléas de données ? Quelles sont les conséquences du forwarding sur la gestion des aléas de contrôle ?

1. Il permet de définir quelle opération doit être écrite dans la mémoire.
2. C'est possible mais pas très utile. En effet, le résultat ne serait pas disponible à l'execute de l'instruction suivante. Il faudrait ajouter un registre dans le write back.
3. Il faut que le pipeline contienne des aléas de données. Il est utile lorsque le pipeline subit une forte quantité d'arrêts de pipeline, dûes à des aléas de données.
4. Les aléas de données sont tous éliminés sauf dans le cas d'une instruction suivant directement un LDR et possédant une dépendance sur celui-ci. En revanche, le forwarding n'a aucun effet sur la gestion des aléas de contrôles.

2.2

Questions

1. Pourquoi doit-on faire ça ?
2. Pourquoi doit-on faire ça pour le signal **reg_mem_data_s** ?
3. Que devrait-on faire si on avait un data forwarding venant du WRITE_BACK ?

1. Pour optimiser le circuit. Si le forwarding n'est pas nécessaire, cela ne sert à rien de s'en servir.
2. Pareil
3. Mettre un registre dans le WRITE_BACK

2.3

Questions

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?
2. Quel est l'IPC de votre programme? et le throughput si on considère une clock à 4KHz?
3. Combien de cycles sont nécessaires pour que l'instruction BL soit complétée?
4. Avez-vous d'autres idées d'optimisation de ce processeur?

1. Oui. Oui. Non.
2. $IPC = \frac{nbrInstr}{nbrClk} = \frac{12}{25} = 0.48$
   $Débit = Fréquence = 4KHz$
3. 3 cycles
4. Non.