Questions 1.1

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 ?

Il suffit de mémoriser pour les 3 derniers cycles l’adresse du registre de destination. Si pour l’instruction en cours, l’adresse des registres n, m ou mem en lecture est le même que l’un des registres de destination mémorisé, on sait qu’il y aura un aléa de type RAW.

1. Est-ce que ça pose un problème si une instruction dépend du résultat d’une instruction qui est au stage WRITE\_BACK ?

Techniquement, sur Logisim ça fonctionne, le WRITE BACK et la lecture du registre sont faites de manière instantanée, mais en réalité, il faudrait ajouter un coup de clock, à cause du délai de propagation.

1. Quelles informations doivent être mémorisées pour chaque instruction ?
2. L’adresse du registre de destination
3. L’information de si pour cette instruction on veut écrire dans la banque de registre
4. L’information de si pour cette instruction on veut écrire dans la mémoire de données
5. Quelles informations permettent de savoir si le registre D est utilisé ?

Il suffit de savoir si une écriture est demandée dans la banque de registres, donc on doit connaître le signal bank\_wr\_s qui vient du reg\_bank\_cotrol\_unit dans le main\_ctrl\_unit dans le bloc decode.

Questions 1.2

**Commande des signaux dans main\_control\_unit**

1. Quelles informations permettent de savoir si le registre N, M ou mem sont utilisés ?
   1. Pour N : Si le bloc Execute a sélectionné l’opérande 1, on sait que la lecture se fait sur le registre N
   2. De même, si le bus de sélection de l’opérande 2 est à 0, on sait qu’on va lire le registre M
   3. Finalement, si on sait que l’instruction est une instruction de type STR, STRH ou STRB (signal str\_data du bloc memory\_access\_control\_unit dans le bloc decode), on sait que le registre MEM sera lu

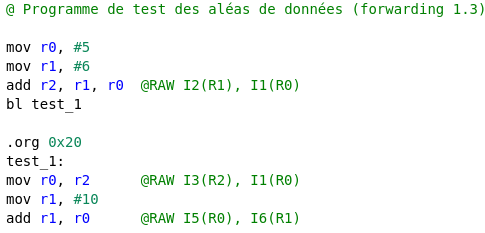
**Commande des signaux dans hazard\_detection**

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

Il suffit de savoir si une écriture est demandée dans la banque de registres, donc on doit connaître le signal bank\_wr\_s qui vient du reg\_bank\_cotrol\_unit dans le main\_ctrl\_unit dans le bloc decode.

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

Lors de la détection d’un aléa de donnée, et si aucun aléa de contrôle n’est détecté, chaque enable des blocs sera désactivé en cascade : DECODE => EXECUTE => M\_ACCESS => WRITE\_BACK (REG\_BANK). La désactivation en cascade permet de terminer de traiter l’instruction en cours

1.3 : programme :

Questions 1.3 (Test aléas de donnée)

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 ?

L’instruction BL est composée de 2 instructions : bl\_msb et bl\_lsb.

bl\_msb calcule une partie de l’adresse du saut et l’écrit dans le LR, puis bl\_lsb lit le LR et calcule d’adresse du saut complète.

Il y a donc un aléa de données, car il faut que bl\_msb ait fini d’écrire avant que bl\_lsb vienne lire dans le LR

Il y a aussi un aléa de contrôle, car il faut attendre que la première partie de l’adresse du saut soit calculée via le bloc execute.

1. Combien de cycles sont nécessaires pour résoudre les aléas de l’instruction BL ?
2. Quel est l’IPC pour votre programme ?

Questions 2.1 (Circuit data\_hazard)

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 ?

Questions 2.2 (Circuit Execute)

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 ?

Questions 2.3 (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 ?
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 ?