

Écrivez un programme LMC qui imprime le carré des nombres consécutifs entre 1 et 20.

7.2 Supposons que les instructions suivantes se trouvent aux emplacements :

```
20 LDA 50
21 ADD 51
50 724
51 006
```

- Affichez le contenu de l'IR, du PC, du MAR, du MDR et de A à la fin de l'instruction 20.
- Affichez le contenu de chaque registre à chaque étape du cycle *fetch-execute* lors de l'exécution de l'instruction 21.

7.7

- Quel est l'effet de décaler un nombre non signé dans un registre de deux bits à gauche ? D'un bit à droite ? Supposons que des 0 sont insérés pour remplacer les bits à la fin du registre qui sont devenus vides en raison du décalage.
- Supposons que le numéro soit signé, c'est-à-dire stocké en complément à 2. Quel est maintenant l'effet de décaler le nombre ?
- Supposons que le décalage exclue le bit de signe, de sorte que le bit de signe reste toujours le même. De plus, supposons que lors d'un décalage à droite, le bit de signe est toujours utilisé comme bit d'insertion à la fin du nombre (au lieu de 0). Quel est l'effet de ces changements ?

7.9 En utilisant les opérations de registre, montrez le cycle *fetch-execute* pour une instruction qui produit le complément à 2 du nombre dans A.

Montrez le cycle *fetch-execute* pour une instruction qui efface A (mise à zéro de A).

7.12 La *Little Prince Computer* est une mutation de la LMC. La LPC a une instruction supplémentaire. Cette instruction requiert deux mots consécutifs :

```
0XX
0YY
```

Cette instruction, *move*, déplace les données directement de l'emplacement XX vers l'emplacement YY sans affecter la valeur dans l'accumulateur.

Pour exécuter cette instruction, le petit prince aurait besoin de stocker les données XX temporairement. Il peut le faire en écrivant la valeur dans un morceau de papier et en le gardant jusqu'à ce qu'il récupère la deuxième adresse. L'équivalent dans un vrai CPU pourrait s'appeler le registre d'adresse intermédiaire (IAR).

Écrivez le cycle *fetch-execute* pour l'instruction MOVE de la LPC.